man pages section 3: Library Interfaces and Headers
Contents

Preface ..................................................................................................................................................... 9

Introduction ......................................................................................................................................... 13
Intro(3) ................................................................................................................................................ 14

Library Interfaces and Headers .......................................................................................................... 27
acct.h(3HEAD) .................................................................................................................................. 28
aio.h(3HEAD) ................................................................................................................................. 30
archives.h(3HEAD) .......................................................................................................................... 31
ar.h(3HEAD) .................................................................................................................................... 36
assert.h(3HEAD) ............................................................................................................................. 39
complex.h(3HEAD) .......................................................................................................................... 40
cpio.h(3HEAD) .................................................................................................................................. 42
dirent.h(3HEAD) ............................................................................................................................. 44
errno.h(3HEAD) .................................................................................................................................. 45
fcntl.h(3HEAD) .................................................................................................................................. 46
fenv.h(3HEAD) .................................................................................................................................. 50
float.h(3HEAD) .................................................................................................................................. 53
floatingpoint.h(3HEAD) ....................................................................................................................... 56
fmtmsg.h(3HEAD) .................................................................................................................................. 58
fnmatch.h(3HEAD) .......................................................................................................................... 60
ftw.h(3HEAD) .................................................................................................................................... 61
glob.h(3HEAD) .................................................................................................................................... 62
grp.h(3HEAD) .................................................................................................................................... 64
iconv.h(3HEAD) .................................................................................................................................... 65
if.h(3HEAD) ....................................................................................................................................... 66
inet.h(3HEAD) .................................................................................................................................... 67
<table>
<thead>
<tr>
<th>Header/Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>in.h</td>
<td>68</td>
</tr>
<tr>
<td>inttypes.h</td>
<td>70</td>
</tr>
<tr>
<td>ipc.h</td>
<td>72</td>
</tr>
<tr>
<td>iso646.h</td>
<td>73</td>
</tr>
<tr>
<td>langinfo.h</td>
<td>74</td>
</tr>
<tr>
<td>libadm</td>
<td>78</td>
</tr>
<tr>
<td>libaio</td>
<td>79</td>
</tr>
<tr>
<td>libauto_ef</td>
<td>80</td>
</tr>
<tr>
<td>libbgsdmalloc</td>
<td>81</td>
</tr>
<tr>
<td>libbsm</td>
<td>82</td>
</tr>
<tr>
<td>libc</td>
<td>84</td>
</tr>
<tr>
<td>libc_db</td>
<td>114</td>
</tr>
<tr>
<td>libcfgadm</td>
<td>117</td>
</tr>
<tr>
<td>libcommputil</td>
<td>118</td>
</tr>
<tr>
<td>libcontract</td>
<td>120</td>
</tr>
<tr>
<td>libcpc</td>
<td>122</td>
</tr>
<tr>
<td>libcrypt</td>
<td>124</td>
</tr>
<tr>
<td>libcurses</td>
<td>125</td>
</tr>
<tr>
<td>libcurses3</td>
<td>132</td>
</tr>
<tr>
<td>libdat</td>
<td>134</td>
</tr>
<tr>
<td>libdbm</td>
<td>137</td>
</tr>
<tr>
<td>libdevaid</td>
<td>138</td>
</tr>
<tr>
<td>libdevinfo</td>
<td>139</td>
</tr>
<tr>
<td>libdl</td>
<td>143</td>
</tr>
<tr>
<td>libdpi</td>
<td>144</td>
</tr>
<tr>
<td>libdmi</td>
<td>145</td>
</tr>
<tr>
<td>libdmici</td>
<td>146</td>
</tr>
<tr>
<td>libdmmimi</td>
<td>147</td>
</tr>
<tr>
<td>libdoor</td>
<td>149</td>
</tr>
<tr>
<td>libdtrace</td>
<td>150</td>
</tr>
<tr>
<td>libelf</td>
<td>151</td>
</tr>
<tr>
<td>libelfi</td>
<td>152</td>
</tr>
<tr>
<td>libexecct</td>
<td>154</td>
</tr>
<tr>
<td>libfmevent</td>
<td>156</td>
</tr>
<tr>
<td>libform</td>
<td>158</td>
</tr>
<tr>
<td>libgen</td>
<td>160</td>
</tr>
<tr>
<td>Library</td>
<td>Page</td>
</tr>
<tr>
<td>-----------------</td>
<td>------</td>
</tr>
<tr>
<td>libgen.h</td>
<td>162</td>
</tr>
<tr>
<td>libgss</td>
<td>163</td>
</tr>
<tr>
<td>libhbaapi</td>
<td>165</td>
</tr>
<tr>
<td>libidnkit</td>
<td>169</td>
</tr>
<tr>
<td>libintl</td>
<td>172</td>
</tr>
<tr>
<td>libkrb5</td>
<td>173</td>
</tr>
<tr>
<td>libkstat</td>
<td>180</td>
</tr>
<tr>
<td>libkvm</td>
<td>181</td>
</tr>
<tr>
<td>libl</td>
<td>182</td>
</tr>
<tr>
<td>liblayout</td>
<td>183</td>
</tr>
<tr>
<td>liblgrp</td>
<td>184</td>
</tr>
<tr>
<td>libm</td>
<td>185</td>
</tr>
<tr>
<td>libmail</td>
<td>202</td>
</tr>
<tr>
<td>libmalloc</td>
<td>203</td>
</tr>
<tr>
<td>libmapmalloc</td>
<td>204</td>
</tr>
<tr>
<td>libmd</td>
<td>205</td>
</tr>
<tr>
<td>libmd5</td>
<td>206</td>
</tr>
<tr>
<td>libmenu</td>
<td>207</td>
</tr>
<tr>
<td>libmlib</td>
<td>209</td>
</tr>
<tr>
<td>libmlib_mt</td>
<td>280</td>
</tr>
<tr>
<td>libmp</td>
<td>282</td>
</tr>
<tr>
<td>libMPAPI</td>
<td>283</td>
</tr>
<tr>
<td>libmtmalloc</td>
<td>288</td>
</tr>
<tr>
<td>libmvect</td>
<td>290</td>
</tr>
<tr>
<td>libnls</td>
<td>292</td>
</tr>
<tr>
<td>libnsi</td>
<td>293</td>
</tr>
<tr>
<td>libnvpair</td>
<td>301</td>
</tr>
<tr>
<td>libpam</td>
<td>304</td>
</tr>
<tr>
<td>libpanel</td>
<td>306</td>
</tr>
<tr>
<td>libpapi</td>
<td>307</td>
</tr>
<tr>
<td>libpctx</td>
<td>310</td>
</tr>
<tr>
<td>libpicl</td>
<td>311</td>
</tr>
<tr>
<td>libpicltree</td>
<td>312</td>
</tr>
<tr>
<td>libPKcs11</td>
<td>314</td>
</tr>
<tr>
<td>libplot</td>
<td>318</td>
</tr>
<tr>
<td>libpool</td>
<td>320</td>
</tr>
</tbody>
</table>
Contents

libproject(3LIB) .......................................................................................................................... 328
libpthread(3LIB) .......................................................................................................................... 329
libresolv(3LIB) ............................................................................................................................ 332
librpcsoc(3LIBUCB) ................................................................................................................... 334
librpcsvc(3LIB) .......................................................................................................................... 335
librsm(3LIB) ............................................................................................................................... 336
librt(3LIB) ................................................................................................................................... 338
librtld_db(3LIB) .......................................................................................................................... 341
libsasl(3LIB) ............................................................................................................................... 342
libscf(3LIB) .................................................................................................................................. 344
libscctp(3LIB) ............................................................................................................................ 348
libsec(3LIB) .................................................................................................................................. 349
libsecdb(3LIB) ............................................................................................................................. 350
libsendfile(3LIB) ........................................................................................................................ 352
libslp(3LIB) .................................................................................................................................. 353
libslp(3LIB) .................................................................................................................................. 358
libsmartcard(3LIB) ....................................................................................................................... 359
libsocket(3LIB) ............................................................................................................................ 361
libssagent(3LIB) .......................................................................................................................... 363
libssasmp(3LIB) .......................................................................................................................... 364
libsys(3LIB) .................................................................................................................................. 365
libsysevent(3LIB) ......................................................................................................................... 371
libtecla(3LIB) ................................................................................................................................ 372
libtermcap(3LIBUCB) ................................................................................................................... 375
libthread(3LIB) ............................................................................................................................. 376
libtnfctl(3LIB) ............................................................................................................................. 378
libtsalarm(3LIB) ........................................................................................................................... 380
libtsnet(3LIB) ................................................................................................................................ 381
libtsol(3LIB) .................................................................................................................................. 382
libucb(3LIBUCB) ........................................................................................................................... 384
libumem(3LIB) ................................................................................................................................ 386
libusb(3LIB) ................................................................................................................................... 387
libuuuid(3LIB) ................................................................................................................................ 389
libv12n(3LIB) .................................................................................................................................. 390
libvolmgt(3LIB) ............................................................................................................................ 391
libw(3LIB) ...................................................................................................................................... 392
<table>
<thead>
<tr>
<th>Library</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>libwsreg(3LIB)</td>
<td></td>
<td>394</td>
</tr>
<tr>
<td>libxnet(3LIB)</td>
<td></td>
<td>396</td>
</tr>
<tr>
<td>libXtsol(3LIB)</td>
<td></td>
<td>399</td>
</tr>
<tr>
<td>liby(3LIB)</td>
<td></td>
<td>400</td>
</tr>
<tr>
<td>limits.h(3HEAD)</td>
<td></td>
<td>401</td>
</tr>
<tr>
<td>locale.h(3HEAD)</td>
<td></td>
<td>411</td>
</tr>
<tr>
<td>math.h(3HEAD)</td>
<td></td>
<td>413</td>
</tr>
<tr>
<td>mman.h(3HEAD)</td>
<td></td>
<td>416</td>
</tr>
<tr>
<td>monetary.h(3HEAD)</td>
<td></td>
<td>418</td>
</tr>
<tr>
<td>mqueue.h(3HEAD)</td>
<td></td>
<td>419</td>
</tr>
<tr>
<td>msg.h(3HEAD)</td>
<td></td>
<td>420</td>
</tr>
<tr>
<td>ndbm.h(3HEAD)</td>
<td></td>
<td>421</td>
</tr>
<tr>
<td>netdb.h(3HEAD)</td>
<td></td>
<td>422</td>
</tr>
<tr>
<td>nl_types.h(3HEAD)</td>
<td></td>
<td>424</td>
</tr>
<tr>
<td>poll.h(3HEAD)</td>
<td></td>
<td>425</td>
</tr>
<tr>
<td>pthread.h(3HEAD)</td>
<td></td>
<td>427</td>
</tr>
<tr>
<td>pwd.h(3HEAD)</td>
<td></td>
<td>429</td>
</tr>
<tr>
<td>regex.h(3HEAD)</td>
<td></td>
<td>430</td>
</tr>
<tr>
<td>resource.h(3HEAD)</td>
<td></td>
<td>432</td>
</tr>
<tr>
<td>sched.h(3HEAD)</td>
<td></td>
<td>434</td>
</tr>
<tr>
<td>search.h(3HEAD)</td>
<td></td>
<td>435</td>
</tr>
<tr>
<td>select.h(3HEAD)</td>
<td></td>
<td>436</td>
</tr>
<tr>
<td>semaphore.h(3HEAD)</td>
<td></td>
<td>437</td>
</tr>
<tr>
<td>sem.h(3HEAD)</td>
<td></td>
<td>438</td>
</tr>
<tr>
<td>setjmp.h(3HEAD)</td>
<td></td>
<td>440</td>
</tr>
<tr>
<td>shm.h(3HEAD)</td>
<td></td>
<td>441</td>
</tr>
<tr>
<td>siginfo.h(3HEAD)</td>
<td></td>
<td>442</td>
</tr>
<tr>
<td>signal.h(3HEAD)</td>
<td></td>
<td>446</td>
</tr>
<tr>
<td>socket.h(3HEAD)</td>
<td></td>
<td>453</td>
</tr>
<tr>
<td>spawn.h(3HEAD)</td>
<td></td>
<td>459</td>
</tr>
<tr>
<td>stat.h(3HEAD)</td>
<td></td>
<td>460</td>
</tr>
<tr>
<td>statvfs.h(3HEAD)</td>
<td></td>
<td>462</td>
</tr>
<tr>
<td>stdbool.h(3HEAD)</td>
<td></td>
<td>463</td>
</tr>
<tr>
<td>stddef.h(3HEAD)</td>
<td></td>
<td>464</td>
</tr>
<tr>
<td>stdint.h(3HEAD)</td>
<td></td>
<td>465</td>
</tr>
<tr>
<td>stdio.h(3HEAD)</td>
<td></td>
<td>472</td>
</tr>
<tr>
<td>Header File</td>
<td>Page</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>stdlib.h</td>
<td>474</td>
<td></td>
</tr>
<tr>
<td>string.h</td>
<td>476</td>
<td></td>
</tr>
<tr>
<td>strings.h</td>
<td>477</td>
<td></td>
</tr>
<tr>
<td>stropts.h</td>
<td>478</td>
<td></td>
</tr>
<tr>
<td>syslog.h</td>
<td>483</td>
<td></td>
</tr>
<tr>
<td>tar.h</td>
<td>485</td>
<td></td>
</tr>
<tr>
<td>tcp.h</td>
<td>488</td>
<td></td>
</tr>
<tr>
<td>termios.h</td>
<td>489</td>
<td></td>
</tr>
<tr>
<td>tgmath.h</td>
<td>494</td>
<td></td>
</tr>
<tr>
<td>timeb.h</td>
<td>498</td>
<td></td>
</tr>
<tr>
<td>time.h</td>
<td>499</td>
<td></td>
</tr>
<tr>
<td>times.h</td>
<td>501</td>
<td></td>
</tr>
<tr>
<td>types32.h</td>
<td>502</td>
<td></td>
</tr>
<tr>
<td>types.h</td>
<td>503</td>
<td></td>
</tr>
<tr>
<td>ucontext.h</td>
<td>507</td>
<td></td>
</tr>
<tr>
<td>uio.h</td>
<td>508</td>
<td></td>
</tr>
<tr>
<td>ulimit.h</td>
<td>509</td>
<td></td>
</tr>
<tr>
<td>un.h</td>
<td>510</td>
<td></td>
</tr>
<tr>
<td>unistd.h</td>
<td>511</td>
<td></td>
</tr>
<tr>
<td>utime.h</td>
<td>522</td>
<td></td>
</tr>
<tr>
<td>utmpx.h</td>
<td>523</td>
<td></td>
</tr>
<tr>
<td>utsname.h</td>
<td>525</td>
<td></td>
</tr>
<tr>
<td>values.h</td>
<td>526</td>
<td></td>
</tr>
<tr>
<td>wait.h</td>
<td>527</td>
<td></td>
</tr>
<tr>
<td>wchar.h</td>
<td>529</td>
<td></td>
</tr>
<tr>
<td>wctype.h</td>
<td>531</td>
<td></td>
</tr>
<tr>
<td>wordexp.h</td>
<td>532</td>
<td></td>
</tr>
</tbody>
</table>
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 ioctl(2) system call is called ioctl and generates its own
heading. `ioctl` calls for a specific device are listed alphabetically (on the man page for that specific device). `ioctl` calls are used for a particular class of devices all of which have an `io` ending, such as `mtio(7I)`.

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 `errno` indicating why they failed. This section lists alphabetically all error codes a function can generate and describes the conditions that cause each error. When more than one condition can cause the same error, each condition is described in a separate paragraph under the error code.

USAGE
This section 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
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXAMPLES</td>
<td>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.</td>
</tr>
<tr>
<td>ENVIRONMENT VARIABLES</td>
<td>This section lists any environment variables that the command or function affects, followed by a brief description of the effect.</td>
</tr>
<tr>
<td>EXIT STATUS</td>
<td>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.</td>
</tr>
<tr>
<td>FILES</td>
<td>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.</td>
</tr>
<tr>
<td>ATTRIBUTES</td>
<td>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.</td>
</tr>
<tr>
<td>SEE ALSO</td>
<td>This section lists references to other man pages, in-house documentation, and outside publications.</td>
</tr>
<tr>
<td>DIAGNOSTICS</td>
<td>This section lists diagnostic messages with a brief explanation of the condition causing the error.</td>
</tr>
<tr>
<td>WARNINGS</td>
<td>This section lists warnings about special conditions which could seriously affect your working conditions. This is not a list of diagnostics.</td>
</tr>
<tr>
<td>NOTES</td>
<td>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.</td>
</tr>
<tr>
<td>BUGS</td>
<td>This section describes known bugs and, wherever possible, suggests workarounds.</td>
</tr>
</tbody>
</table>
REFERENCE

Introduction
**Name**  Intro – introduction to functions and libraries

**Description**  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 `#include` 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 six volumes as described below. A seventh 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.

**Library Interfaces and Headers**  This volume describes the contents of each shared library and each header used by functions, macros, and external variables described in the remaining six volumes.

- **(3LIB)**  The libraries described in this section are implemented as shared objects. Descriptions of shared objects can include a definition of the global symbols that define the shared objects' public interface, for example `SUNW_1.1`. Other interfaces can exist within the shared object, for example `SUNW_private_1.1`. The public interface provides a stable, committed set of symbols for application development. The private interfaces are for internal use only, and could change at any time.

- **(3LIBUCB)**  The SunOS/BSD Compatibility libraries described in this section are implemented as a shared object. See (3LIB) above.

- **(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.

**Basic Library Functions**  The functions described in this volume are the core C library functions that are basic to application development.

- **(3C)**  These functions, together with those of Section 2, constitute the standard C library, libc, which is automatically linked by the C compilation system. The standard C library is implemented as a shared object, libc.so. See `libc(3LIB)` 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).
The libpthread and libthread libraries are filter libraries on libc that are used for building multithreaded applications: libpthread implements the POSIX (see standards(5)) threads interface, whereas libthread implements the Solaris threads interface. See MULTITHREADED APPLICATIONS, below.

(3C_DB) These functions constitute the threads debugging library, libc_db. This library is implemented as a shared object, libc_db.so, but is not automatically linked by the C compilation system. Specify -lc_db on the cc command line to link with this library. See libc_db(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). 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, 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/ucb/include. See libucb(3LIBUCB).

Networking Library Functions

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

(3COMMPUTIL) These functions constitute the communication protocol parser utilities library, libcommputil. This library is implemented as a shared object, libcommputil.so, but it is not automatically linked by the C compilation system. Specify -lcommputil on the cc command line to link with this library. See libcommputil(3LIB).

(3DLPI) These functions constitute the data link provider interface library, libdlpi. This library is implemented as a shared object, libdlpi.so, but it is not automatically linked by the C compilation system. Specify -ldlpi on the cc command line to link with this library. See libdlpi(3LIB).

(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, \texttt{libgss.so}, but it is not automatically linked by the C compilation system. Specify \texttt{-lgss} on the \texttt{cc} command line to link with this library. See \texttt{libgss(3LIB)}.

(3LDAP) These functions constitute the lightweight directory access protocol library, \texttt{libldap}. This library is implemented as a shared object, \texttt{libldap.so}, but is not automatically linked by the C compilation system. Specify \texttt{-lldap} on the \texttt{cc} command line to link with this library. See \texttt{ldap(3LDAP)}.

(3NSL) These functions constitute the network service library, \texttt{libnsl}. This library is implemented as a shared object, \texttt{libnsl.so}, but is not automatically linked by the C compilation system. Specify \texttt{-lnsl} on the \texttt{cc} command line to link with this library. See \texttt{libnsl(3LIB)}.

Many base networking functions are also available in the X/Open networking interfaces library, \texttt{libxnet}. See section (3XNET) below for more information on the \texttt{libxnet} interfaces.

(3RESOLV) These functions constitute the resolver library, \texttt{libresolv}. This library is implemented as a shared object, \texttt{libresolv.so}, but is not automatically linked by the C compilation system. Specify \texttt{-lresolv} on the \texttt{cc} command line to link with this library. See \texttt{libresolv(3LIB)}.

(3RPC) These functions constitute the remote procedure call libraries, \texttt{librpcsvc} and \texttt{librpcsoc}. The latter is provided for compatibility only; new applications should not link to it. Both libraries are implemented as shared objects, \texttt{librpcsvc.so} and \texttt{librpcsoc.so}, respectively. Neither library is automatically linked by the C compilation system. Specify \texttt{-lrpcsvc} or \texttt{-lrpcsoc} on the \texttt{cc} command line to link with these libraries. See \texttt{librpcsvc(3LIB)} and \texttt{librpcsoc(3LIBUCB)}.

(3SASL) These functions constitute the simple authentication and security layer library, \texttt{libsasl}. This library is implemented as a shared object, \texttt{libsasl.so}, but it is not automatically linked by the C compilation system. Specify \texttt{-lsasl} on the \texttt{cc} command line to link with this library. See \texttt{libsasl(3LIB)}.

(3SIP) These functions constitute the session initiation protocol library, \texttt{libsip}. This library is implemented as a shared object, \texttt{libsip.so}, but it is not automatically linked by the C compilation system. Specify \texttt{-lsip} on the \texttt{cc} command line to link with this library. See \texttt{libsip(3LIB)}.

(3SLP) These functions constitute the service location protocol library, \texttt{libslp}. This library is implemented as a shared object, \texttt{libslp.so}, but it is not automatically linked by the C compilation system. Specify \texttt{-lslp} on the \texttt{cc} command line to link with this library. See \texttt{libslp(3LIB)}.
These functions constitute the sockets library, `libsocket`. This library is implemented as a shared object, `libsocket.so`, but is not automatically linked by the C compilation system. Specify `-lsocket` on the `cc` command line to link with this library. See `libsocket(3LIB)`.

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 XPG4v2 (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 interfaces, the XTI interfaces (available with `libxnet`) are recommended over the TLI interfaces (available with `libnsl`).

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

These functions constitute the following libraries:

- `libcurses` : This library is implemented as a shared object, `libcurses.so`, 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` : This library is implemented as a shared object, `libform.so`, 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` : This library is implemented as a shared object, `libmenu.so`, 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` : This library is implemented as a shared
These functions constitute the graphics library, \texttt{libplot}. This library is implemented as a shared object, \texttt{libplot.so}, but is not automatically linked by the C compilation system. Specify \texttt{-lplot} on the \texttt{cc} command line to link with this library. See \texttt{libplot(3LIB)}.

These functions constitute the X/Open curses library, located in 
\texttt{/usr/xpg4/lib/libcurses.so}. 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. See \texttt{libcurses(3XCURSES)}.

The functions described in this volume constitute the real-time libraries.

These functions constitute the asynchronous I/O library, \texttt{libaio}. This library is implemented as a shared object, \texttt{libaio.so}, but is not automatically linked by the C compilation system. Specify \texttt{-laio} on the \texttt{cc} command line to link with this library. See \texttt{libaio(3LIB)}.

These functions constitute the POSIX.4 real-time library, \texttt{librt}. It is implemented as a shared object, \texttt{librt.so}, but is not automatically linked by the C compilation system. Specify \texttt{-lrt} on the \texttt{cc} command line to link with this library. Note that the former name for this library, \texttt{libposix4}, is maintained for backward compatibility but should be avoided. See \texttt{librt(3LIB)}.

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

These functions constitute the basic security library, \texttt{libbsm}. This library is implemented as a shared object, \texttt{libbsm.so}, but is not automatically linked by the C compilation system. Specify \texttt{-lbsm} on the \texttt{cc} command line to link with this library. See \texttt{libbsm(3LIB)}.

These functions constitute the configuration administration library, \texttt{libcfgadm}. This library is implemented as a shared object, \texttt{libcfgadm.so}, but is not automatically linked by the C compilation system. Specify \texttt{-lcfgadm} on the \texttt{cc} command line to link with this library. See \texttt{libcfgadm(3LIB)}. 

object, \texttt{libpanel.so}, but is not automatically linked by the C compilation system. Specify \texttt{-lpanel} on the \texttt{cc} command line to link with this library. See \texttt{libpanel(3LIB)}.
These functions constitute the contract management library, libcontract. This library is implemented as a shared object, libcontract.so, but is not automatically linked by the C compilation system. Specify -lcontract on the cc command line to link with this library. See libcontract(3LIB).

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).

These functions constitute the direct access transport library, libdat. This library is implemented as a shared object, libdat.so, but is not automatically linked by the C compilation system. Specify -ldat on the cc command line to link with this library. See libdat(3LIB).

These functions constitute the device ID library, libdevid. This library is implemented as a shared object, libdevid.so, but is not automatically linked by the C compilation system. Specify -ldevid on the cc command line to link with this library. See libdevid(3LIB).

These functions constitute the device information library, libdevinfo. This library is implemented as a shared object, libdevinfo.so, but is not automatically linked by the C compilation system. Specify -ldevinfo on the cc command line to link with this library. See libdevinfo(3LIB).

These functions constitute the DMI libraries, libdmi, libdmici, and libdmimi. These libraries are implemented as shared objects, libdmi.so, libdmici.so, and libdmimi.so, respectively, but are not automatically linked by the C compilation system. Specify -ldmi, -ldmici, or -ldmimi on the cc command line to link with these libraries. See libdmi(3LIB), libdmici(3LIB), and libdmimi(3LIB).

These functions constitute the ELF access library, libelf, (Extensible Linking Format). This library provides the interface for the creation and analyses of "elf" files; executables, objects, and shared objects. libelf is implemented as a shared object, libelf.so, but is not automatically linked by the C compilation system. Specify -lelf on the cc command line to link with this library. See libelf(3LIB).

These functions constitute the extended accounting access library, libexacct, and the project database access library, libproject. These libraries are implemented as shared objects, libexacct.so and libproject.so, respectively, but are not automatically linked by the C
compilation system. Specify -lexacct or -lproject on the cc command line to link with these libraries. See `libexacct(3LIB)` and `libproject(3LIB)`.

(3FM) These functions constitute the fault management events library. This library is implemented as a shared object, `libfmevent.so`, but is not automatically linked by the C compilation system. Specify -lfmevent on the cc command line to link with this library. See `libfmevent(3LIB)`.

(3GEN) These functions constitute the string pattern-matching and pathname manipulation library, `libgen`. This library is implemented as a shared object, `libgen.so`, but is not automatically linked by the C compilation system. Specify -lgen on the cc command line to link with this library. See `libgen(3LIB)`.

(3HBAAPI) These functions constitute the common fibre channel HBA information library, `libhbbaapi`. This library is implemented as a shared object, `libhbbaapi.so`, but is not automatically linked by the C compilation system. Specify -lhbaapi on the cc command line to link with this library. See `libhbbaapi(3LIB)`.

(3KSTAT) These functions constitute the kernel statistics library, which is implemented as a shared object, `libkstat.so`, but is not automatically linked by the C compilation system. Specify -lkstat on the cc command line to link with this library. See `libkstat(3LIB)`.

(3KVM) These functions allow access to the kernel's virtual memory library, which is implemented as a shared object, `libkvm.so`, but is not automatically linked by the C compilation system. Specify -lkvm on the cc command line to link with this library. See `libkvm(3LIB)`.

(3LAYOUT) These functions constitute the layout service library, which is implemented as a shared object, `liblayout.so`, but is not automatically linked by the C compilation system. Specify -llayout on the cc command line to link with this library. See `liblayout(3LIB)`.

(3LGRP) These functions constitute the locality group library, which is implemented as a shared object, `liblgrp.so`, but is not automatically linked by the C compilation system. Specify -llgrp on the cc command line to link with this library. See `liblgrp(3LIB)`.

(3M) These functions constitute the mathematical library, `libm`. This library is implemented as a shared object, `libm.so`, but is not automatically linked by the C compilation system. Specify -lm on the cc command line to link with this library. See `libm(3LIB)`.

(3MAIL) These functions constitute the user mailbox management library, `libmail`. This library is implemented as a shared object, `libmail.so`, but
is not automatically linked by the C compilation system. Specify `-lmail` on the `cc` command line to link with this library. See `libmail(3LIB)`.

(3MP) These functions constitute the integer mathematical library, `libmp`. This library is implemented as a shared object, `libmp.so`, but is not automatically linked by the C compilation system. Specify `-lmp` on the `cc` command line to link with this library. See `libmp(3LIB)`.

(3MPAPI) These functions constitute the Common Multipath Management library, `libMPAPI`. This library is implemented as a shared object, `libMPAPI.so`, but is not automatically linked by the C compilation system. Specify `-lMPAPI` on the `cc` command line to link with this library. See `libMPAPI(3LIB)`.

(3MVEC) These functions constitute the vector mathematical library, `libmvec`. This library is implemented as a shared object, `libmvec.so`, but is not automatically linked by the C compilation system. Specify `-lmvec` on the `cc` command line to link with this library. See `libmvec(3LIB)`.

(3NVPAIR) These functions constitute the name–value pair library, `libnvpair`. This library is implemented as a shared object, `libnvpair.so`, but is not automatically linked by the C compilation system. Specify `-lnvpair` on the `cc` command line to link with this library. See `libnvpair(3LIB)`.

(3PAM) These functions constitute the pluggable authentication module library, `libpam`. This library is implemented as a shared object, `libpam.so`, but is not automatically linked by the C compilation system. Specify `-lpam` on the `cc` command line to link with this library. See `libpam(3LIB)`.

(3PAPI) These functions constitute the Free Standards Group Open Printing API (PAPI) library, `libpapi`. This library is implemented as a shared object, `libpapi.so`, but is not automatically linked by the C compilation system. Specify `-lpapi` on the `cc` command line to link with this library. See `libpapi(3LIB)`.

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

(3PICLTREE) These functions constitute the PICL plug-in library, `libpicltree`. This library is implemented as a shared object, `libpicltree.so`, but is not automatically linked by the C compilation system. Specify `-lpicltree` on the `cc` command line to link with this library. See `libpicltree(3LIB)` and `libpicltree(3PICLTREE)`.

(3POOL) These functions constitute the pool configuration manipulation library, `libpool`. This library is implemented as a shared object, `libpool.so`, but
is not automatically linked by the C compilation system. Specify `-lpool` on the `cc` command line to link with this library. See `libpool(3LIB)`.

(3PROJECT) These functions constitute the project database access library, `libproject`. This library is implemented as a shared object, `libproject.so`, but is not automatically linked by the C compilation system. Specify `-lproject` on the `cc` command line to link with this library. See `libproject(3LIB)`.

(3RSM) These functions constitute the remote shared memory library, `librsm`. This library is implemented as a shared object, `librsm.so`, but is not automatically linked by the C compilation system. Specify `-lrsm` on the `cc` command line to link with this library. See `librsm(3LIB)`.

(3SCF) These functions constitute the object-caching memory allocation library, `libscf`. This library is implemented as a shared object, `libscf.so`, but is not automatically linked by the C compilation system. Specify `-lscf` on the `cc` command line to link with this library. See `libscf(3LIB)`.

(3SEC) These functions constitute the file access control library, `libsec`. This library is implemented as a shared object, `libsec.so`, but is not automatically linked by the C compilation system. Specify `-lsec` on the `cc` command line to link with this library. See `libsec(3LIB)`.

(3SECDB) These functions constitute the security attributes database library, `libsecdb`. This library is implemented as a shared object, `libsecdb.so`, but is not automatically linked by the C compilation system. Specify `-lsecdb` on the `cc` command line to link with this library. See `libsecdb(3LIB)`.

(3SMARTCARD) These functions constitute the smartcard library, `libsmartcard`. This library is implemented as a shared object, `libsmartcard.so`, but is not automatically linked by the C compilation system. Specify `-lsmartcard` on the `cc` command line to link with this library. See `libsmartcard(3LIB)`.

(3SNMP) These functions constitute the SNMP libraries, `libssagent` and `libssasnmp`. These libraries are implemented as shared objects, `libssagent.so` and `libssasnmp.so`, respectively, but are not automatically linked by the C compilation system. Specify `-lssagent` or `-lssasnmp` on the `cc` command line to link with these libraries. See `libssagent(3LIB)` and `libssasnmp(3LIB)`.

(3SYSEVENT) These functions constitute the system event library, `libsysevent`. This library is implemented as a shared object, `libsysevent.so`, but is not automatically linked by the C compilation system. Specify `-l sysevent` on the `cc` command line to link with this library. See `libsysevent(3LIB)`.
<table>
<thead>
<tr>
<th>Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>(3TECLA)</strong></td>
<td>These functions constitute the interactive command-line input library, libtecla. This library is implemented as a shared object, libtecla.so, but is not automatically linked by the C compilation system. Specify -ltecla on the cc command line to link with this library. See libtecla(3LIB).</td>
</tr>
<tr>
<td><strong>(3TNF)</strong></td>
<td>These functions constitute the TNF libraries, libtnf, libtnfctl, and libtnfprobe. These libraries are implemented as shared objects, libtnf.so, libtnfctl.so, and libtnfprobe.so, respectively, but are not automatically linked by the C compilation system. Specify -lt nf, -lt nfctl, or -ltnfprobe on the cc command line to link with these libraries. See libtnfctl(3TNF) and libtnfctl(3LIB).</td>
</tr>
<tr>
<td><strong>(3TSOL)</strong></td>
<td>These functions constitute the Trusted Extensions library, libtsol, and the Trusted Extensions network library, libtsnet. These libraries are implemented as shared objects, libtsol.so and libtsnet.so, but are not automatically linked by the C compilation system. Specify -lt tsol or -ltsnet on the cc command line to link with these libraries. See libtsol(3LIB) and libtsnet(3LIB).</td>
</tr>
<tr>
<td><strong>(3UUUID)</strong></td>
<td>These functions constitute the universally unique identifier library, libuuid. This library is implemented as a shared object, libuuid.so, but is not automatically linked by the C compilation system. Specify -luuid on the cc command line to link with this library. See libuuid(3LIB).</td>
</tr>
<tr>
<td><strong>(3VOLMGT)</strong></td>
<td>These functions constitute the volume management library, libvolmgt. This library is implemented as a shared object, libvolmgt.so, but is not automatically linked by the C compilation system. Specify -lvolmgt on the cc command line to link with this library. See libvolmgt(3LIB).</td>
</tr>
<tr>
<td><strong>(3WSREG)</strong></td>
<td>These functions constitute the product install registry library, libwsreg. This library is implemented as a shared object, libwsreg.so, but is not automatically linked by the C compilation system. Specify -lwsreg on the cc command line to link with this library. See libwsreg(3LIB).</td>
</tr>
<tr>
<td><strong>(3XTSOL)</strong></td>
<td>These functions constitute the Trusted Extensions to the X windows library, libXtsol. This library is implemented as a shared object, libXtsol.so, but is not automatically linked by the C compilation system. Specify -lX11 and then -lXtsol on the cc command line to link with this library. See libXtsol(3LIB).</td>
</tr>
<tr>
<td><strong>(3MLIB)</strong></td>
<td>These functions constitute the mediaLib library, libmlib. This library is implemented as a shared object, libmlib.so, but is not automatically linked by the C compilation system. Specify -lmlib on the cc command line to link with this library. See libmlib(3LIB).</td>
</tr>
</tbody>
</table>
A character is any bit pattern able to fit into a byte on the machine. In some international languages, however, a “character” might 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.

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.

The `libpthread(3LIB)` and `libthread(3LIB)` libraries are implemented as filters on `libc(3LIB)`.

When compiling a multithreaded application, the `-mt` option must be specified on the command line.

There is no need for a multithreaded application to link with `-lthread`. An application must link with `-lpthread` only when POSIX semantics for `fork(2)` are desired. When an application is linked with `-lpthread`, a call to `fork()` assumes the behavior `fork1(2)` rather than the default behavior that forks all threads.

When compiling a POSIX-conforming application, either the `_POSIX_C_SOURCE` or `_POSIX_PTHREAD_SEMANTICS` option must be specified on the command line. For POSIX.1c-conforming applications, define the `_POSIX_C_SOURCE` flag to be `>= 199506L`:

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

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

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

For Solaris threads behavior, compile as follows:

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

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

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.
The environment variable `LD_BIND_NOW` must be set 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.

**Files**

<table>
<thead>
<tr>
<th>INCDIR</th>
<th>usually /usr/include</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIBDIR</td>
<td>usually either /lib or /usr/lib (32-bit) or either /lib/64 or /usr/lib/64 (64-bit)</td>
</tr>
<tr>
<td>LIBDIR/*.so</td>
<td>shared libraries</td>
</tr>
</tbody>
</table>

**See Also**

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

*Linker and Libraries Guide*

*Performance 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, ±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, ±HUGE VAL is returned instead of ±HUGE. (HUGE VAL and HUGE are defined in `<math.h>` to be infinity and the largest-magnitude single-precision number, respectively.)

**Notes**

None of the functions, external variables, or macros should be redefined in the user’s programs. Any other name can be redefined without affecting the behavior of other library functions, but such redefinition might 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 can also be used and will report discrepancies even if the headers are not included with `#include` statements. Definitions for Sections 2 and 3C are checked automatically. Other definitions can be included by using the `-l` option to `lint`. (For example, `-l m` 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 can 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\}.
REFERENCE

Library Interfaces and Headers
#include <sys/types.h>
#include <sys/acct.h>

Files produced as a result of calling `acct(2)` have records in the form defined by `<sys/acct.h>`, whose contents are:

```c
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 */

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:

\[
\text{(data size) + (text size) / (number of in-core processes using text)}
\]

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.
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.

```c
/*
 * 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), acctmrg(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.
#include <aio.h>

The `aio.h` header defines the `aiocb` structure which includes the following members:

- `int aio_fildes`: file descriptor
- `off_t aio_offset`: file offset
- `volatile void* aio_buf`: location of buffer
- `size_t aio_nbytes`: length of transfer
- `int aio_reqprio`: request priority offset
- `struct sigevent aio_sigevent`: signal number and value
- `int aio_lio_opcode`: operation to be performed

This header also includes the following constants:

- `AIO_ALLDONE`: A return value indicating that none of the requested operations could be canceled since they are already complete.
- `AIO_CANCELED`: A return value indicating that all requested operations have been canceled.
- `AIO_NOTCANCELED`: A return value indicating that some of the requested operations could not be canceled since they are in progress.
- `LIO_NOP`: A `lio_listio(3RT)` element operation option indicating that no transfer is requested.
- `LIO_NOWAIT`: A `lio_listio()` synchronization operation indicating that the calling thread is to continue execution while the `lio_listio()` operation is being performed, and no notification is given when the operation is complete.
- `LIO_READ`: A `lio_listio()` element operation option requesting a read.
- `LIO_WAIT`: A `lio_listio()` synchronization operation indicating that the calling thread is to suspend until the `lio_listio()` operation is complete.
- `LIO_WRITE`: A `lio_listio()` element operation option requesting a write.

See Also: `lseek(2), read(2), write(2), fsync(3C), libaio(3LIB), lio_listio(3RT)`
/* Magic numbers */
#define CMN_ASC 0x070701 /* Cpio Magic Number for -c header */
#define CMN_BIN 070707 /* Cpio Magic Number for Binary header */
#define CMN_BBS 0143561 /* Cpio Magic Number for Byte-Swap header */
#define CMN_CRC 0x070702 /* Cpio Magic Number for CRC header */
#define CMS_ASC "070701" /* Cpio Magic String for -c header */
#define CMS_CHR "070707" /* Cpio Magic String for odc header */
#define CMS_CRC "070702" /* Cpio Magic String for CRC header */
#define CMS_LEN 6 /* Cpio Magic String length */
/* Various header and field lengths */
#define CHRSZ 76 /* -H odc size minus filename field */
#define ASCSZ 110 /* -c and CRC hdr size minus filename field */
#define TARSZ 512 /* TAR hdr size */
#define HNAMLEN 256 /* maximum filename length for binary and odc headers */
#define EXPNLEN 1024 /* maximum filename length for -c and CRC headers */
#define HTIMLEN 2 /* length of modification time field */
#define HSIZLEN 2 /* length of file size field */
/* cpio binary header definition */
struct hdr_cpio {
    short h_magic, /* magic number field */
    h_dev; /* file system of file */
    ushort_t h_ino, /* inode of file */
    h_mode, /* modes of file */
    h_uid, /* uid of file */
    h_gid; /* gid of file */
    short h_nlink, /* number of links to file */
    h_rdev, /* maj/min numbers for special files */
    h_mtime[HTIMLEN], /* modification time of file */
    h_namesize, /* length of filename */
    h_filesize[HSIZLEN]; /* size of file */
    char h_name[HNAMLEN]; /* filename */
};
/* cpio -H odc header format */
struct c_hdr {
    char c_magic[CMS_LEN],
    c_dev[6],
    c_ino[6],
    c_mode[6],
    c_uid[6],
    c_gid[6],
    c_nlink[6],
    c_rdev[6],
    c_mtime[11],
    c_namesz[6],
c_filesz[11],
c_name[HNAMLEN];
}

/* -c and CRC header format */
struct Exp_cpio_hdr {
    char E_magic[CMS_LEN],
    E_ino[8],
    E_mode[8],
    E_uid[8],
    E_gid[8],
    E_nlink[8],
    E_mtime[8],
    E_filesize[8],
    E_maj[8],
    E_min[8],
    E_rmaj[8],
    E_rmin[8],
    E_namesize[8],
    E chksum[8],
    E_name[EXPNLEN];
};

/* Tar header structure and format */
#define TBLOCK 512 /* length of tar header and data blocks */
#define TNAMLEN 100 /* maximum length for tar file names */
#define TMODLEN 8 /* length of mode field */
#define TUIDLEN 8 /* length of uid field */
#define TGIDLEN 8 /* length of gid field */
#define TSIZLEN 12 /* length of size field */
#define TTIMLEN 12 /* length of modification time field */
#define TCRCLEN 8 /* length of header checksum field */

/* tar header definition */
union tblock {
    char dummy[TBLOCK];
    struct {
        char t_name[TNAMLEN]; /* name of file */
        char t_mode[TMODLEN]; /* mode of file */
        char t_uid[TUIDLEN]; /* uid of file */
        char t_gid[TGIDLEN]; /* gid of file */
        char t_size[TSIZLEN]; /* size of file in bytes */
        char t_mtime[TTIMLEN]; /* modification time of file */
        char t_chksum[TCRCLEN]; /* checksum of header */
        char t_typeflag; /* flag to indicate type of file */
        char t_linkname[TNAMLEN]; /* file this file is linked with */
        char t_magic[6]; /* magic string always "ustar" */
        char t_version[2]; /* version strings always ’00’ */
        char t_uname[32]; /* owner of file in ASCII */
        char t_gname[32]; /* group of file in ASCII */
    };
};
# Define archive formats for extended attributes.

* Extended attributes are stored in two pieces.
  * 1. An attribute header which has information about
      * what file the attribute is for and what the attribute
        * is named.
  * 2. The attribute record itself. Stored as a normal file type
      * of entry.
  * Both the header and attribute record have special modes/typeflags
      * associated with them.
  *
  * The names of the header in the archive look like:
    * /dev/null/attr.hdr
      *
  * The name of the attribute looks like:
    * /dev/null/attr.
      *

* This is done so that an archiver that doesn't understand these formats
  * can just dispose of the attribute records unless the user chooses to
    * rename them via cpio -r or pax -i
The format is composed of a fixed size header followed by a variable sized xattr_buf. If the attribute is a hard link to another attribute, then another xattr_buf section is included for the link.

The xattr_buf is used to define the necessary "pathing" steps to get to the extended attribute. This is necessary to support a fully recursive attribute model where an attribute may itself have an attribute.

The basic layout looks like this.

```
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>xattr_hdr</td>
</tr>
<tr>
<td>--------------------------------</td>
</tr>
<tr>
<td>--------------------------------</td>
</tr>
<tr>
<td>xattr_buf</td>
</tr>
<tr>
<td>--------------------------------</td>
</tr>
<tr>
<td>(optional link info)</td>
</tr>
<tr>
<td>--------------------------------</td>
</tr>
<tr>
<td>attribute itself</td>
</tr>
<tr>
<td>stored as normal tar</td>
</tr>
<tr>
<td>or cpio data with special mode</td>
</tr>
<tr>
<td>or typeflag</td>
</tr>
<tr>
<td>--------------------------------</td>
</tr>
</tbody>
</table>
```

#define XATTR_ARCH_VERS "1.0"

/*
* extended attribute fixed header
*
* h_version format version.
* h_size size of header + variable sized data sections.
* h_component_len Length of entire pathing section.
* h_link_component_len Length of link component section. Again same
struct xattr_hdr {
    char h_version[7];
    char h_size[10];
    char h_component_len[10]; /* total length of path component */
    char h_link_component_len[10];
};

struct xattr_buf {
    char h_namesz[7]; /* length of h_names */
    char h_typeflag; /* actual typeflag of file being archived */
    char h_names[1]; /* filepathNULattrpathNUL... */
};

#define _XATTR_HDRTYPE 'E'

#define _XATTR_CPIO_MODE 0xB000
The archive command `ar` is used to combine several files into one. Archives are used mainly as libraries to be searched by the link editor `ld`.

Each archive begins with the archive magic string.

```c
#define ARMAG "!<arch>
/* magic string */
#define SARMAG 8 /* length of magic string */
```

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:

```c
#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 */
};
```

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 `ar_mode` which is in octal). Thus, if the archive contains printable files, the archive itself is printable.

If the file member name fits, the `ar_name` 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, `ar_name` contains a slash (/) followed by a decimal representation of the name's offset in the archive string table described below.

The `ar_date` 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 `ar` is used.

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.

Notice there is no provision for empty areas in an archive file.

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

The archive symbol table has a zero length name (that is, \texttt{ar\_name[0]} is ‘/’), \texttt{ar\_name[1]}==’ ‘
, 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.

\begin{center}
\begin{tabular}{c c c c}
\hline
0 & 1 & 2 & 3 \\
\hline
0x01020304 & 01 & 02 & 03 & 04 \\
\hline
\end{tabular}
\end{center}

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: \texttt{ar\_size} – 4 bytes * ("the number of symbols" + 1).

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

<table>
<thead>
<tr>
<th>Example Symbol Table</th>
<th>Offset</th>
<th>+0</th>
<th>+1</th>
<th>+2</th>
<th>+3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>-</td>
<td></td>
<td></td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>114</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>122</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>426</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>434</td>
<td>-</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td>n</td>
<td>a</td>
<td>m</td>
<td>e</td>
</tr>
<tr>
<td>24</td>
<td></td>
<td>\textbackslash{}o</td>
<td>b</td>
<td>j</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td></td>
<td>e</td>
<td>c</td>
<td>t</td>
<td>\textbackslash{}0</td>
</tr>
<tr>
<td>32</td>
<td></td>
<td>f</td>
<td>u</td>
<td>n</td>
<td>c</td>
</tr>
<tr>
<td>36</td>
<td></td>
<td>t</td>
<td>i</td>
<td>o</td>
<td>n</td>
</tr>
<tr>
<td>40</td>
<td></td>
<td>\textbackslash{}0</td>
<td>n</td>
<td>a</td>
<td>m</td>
</tr>
<tr>
<td>44</td>
<td></td>
<td>e</td>
<td>2</td>
<td>\textbackslash{}0</td>
<td></td>
</tr>
</tbody>
</table>

\begin{center}
\texttt{ar.h(3HEAD)}
\end{center}
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.

<table>
<thead>
<tr>
<th>Offset</th>
<th>+0</th>
<th>+1</th>
<th>+2</th>
<th>+3</th>
<th>+4</th>
<th>+5</th>
<th>+6</th>
<th>+7</th>
<th>+8</th>
<th>+9</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>f</td>
<td>i</td>
<td>l</td>
<td>e</td>
<td>n</td>
<td>a</td>
<td>m</td>
<td>e</td>
<td>_</td>
<td>_</td>
</tr>
<tr>
<td>10</td>
<td>s</td>
<td>a</td>
<td>m</td>
<td>p</td>
<td>l</td>
<td>e</td>
<td>/</td>
<td>\n</td>
<td>l</td>
<td>o</td>
</tr>
<tr>
<td>20</td>
<td>n</td>
<td>g</td>
<td>e</td>
<td>r</td>
<td>f</td>
<td>i</td>
<td>l</td>
<td>e</td>
<td>n</td>
<td>a</td>
</tr>
<tr>
<td>30</td>
<td>m</td>
<td>e</td>
<td>x</td>
<td>a</td>
<td>m</td>
<td>p</td>
<td>l</td>
<td>e</td>
<td>/</td>
<td>\n</td>
</tr>
</tbody>
</table>

Member Name | ar_name
-------------|-------------
short-name | short-name/ | Not in string table
file_name_sample | /0 | Offset 0 in string table
longerfilenamexample | /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.
### assert.h

#### Name
assert.h, assert – verify program assertion

#### Synopsis
```
#include <assert.h>
```

#### Description
The `<assert.h>` header defines the `assert()` macro. It refers to the macro `NDEBUG` which is not defined in the header. If `NDEBUG` is defined as a macro name before the inclusion of this header, the `assert()` macro is defined simply as:
```
#define assert(ignore)((void) 0)
```
Otherwise, the macro behaves as described in `assert(3C)`.

The `assert()` macro is redefined according to the current state of `NDEBUG` each time `<assert.h>` is included.

The `assert()` macro is implemented as a macro, not as a function. If the macro definition is suppressed in order to access an actual function, the behavior is undefined.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

#### See Also
`assert(3C), attributes(5), standards(5)`
The `<complex.h>` header defines the following macros:

- `complex` expands to `_Complex`.
- `_Complex_I` expands to a constant expression of type `const float _Complex`, with the value of the imaginary unit (that is, a number i such that \( i^2 = -1 \)).
- `imaginary` expands to `_Imaginary`.
- `_Imaginary_I` expands to a constant expression of type `const float _Imaginary` with the value of the imaginary unit.
- `I` expands to either `_Imaginary_I` or `_Complex_I`. If `_Imaginary_I` is not defined, I expands to `_Complex_I`.

An application can undefine and then, if appropriate, redefine the `complex`, `imaginary`, and `I` macros.

**Usage**

Values are interpreted as radians, not degrees.

**Attributes**

See [attributes(5)](manpagessection3:LibraryInterfacesandHeaders) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**

cabs(3M), cacos(3M), cacosh(3M), carg(3M), casin(3M), casinh(3M), ccatan(3M), catanh(3M), cccos(3M), ccosh(3M), cexp(3M), cimag(3M), clog(3M), conj(3M), cpow(3M), cproj(3M), creal(3M), csin(3M), csinh(3M), csqrt(3M), ctan(3M), ctanh(3M), attributes(5), standards(5)

**Notes**

The choice of `I` instead of `i` for the imaginary unit concedes to the widespread use of the identifier `i` for other purposes. The application can use a different identifier, say `j`, for the imaginary unit by following the inclusion of the `<complex.h>` header with:

```
#undef I
#define j _Imaginary_I
```

An `I` suffix to designate imaginary constants is not required, as multiplication by `I` provides a sufficiently convenient and more generally useful notation for imaginary terms. The corresponding real type for the imaginary unit is `float`, so that use of `I` for algorithmic or notational convenience does not result in widening types.

On systems with imaginary types, the application has the ability to control whether use of the macro `I` introduces an imaginary type, by explicitly defining `I` to be `_Imaginary_I` or `_Complex_I`.

---

**Synopsis**

```c
#include <complex.h>
```
Disallowing imaginary types is useful for some applications intended to run on implementations without support for such types.

The macro `Imaginary_I` provides a test for whether imaginary types are supported. The `cis()` function `(\cos(x) + I*\sin(x))` was considered but rejected because its implementation is easy and straightforward, even though some implementations could compute sine and cosine more efficiently in tandem.
Values needed by the `c_mode` field of the `cpio` archive format are described as follows:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_IRUSR</td>
<td>Read by owner</td>
</tr>
<tr>
<td>C_IWUSR</td>
<td>Write by owner</td>
</tr>
<tr>
<td>C_IXUSR</td>
<td>Execute by owner</td>
</tr>
<tr>
<td>C_IRGRP</td>
<td>Read by group</td>
</tr>
<tr>
<td>C_IWGRP</td>
<td>Write by group</td>
</tr>
<tr>
<td>C_IXGRP</td>
<td>Execute by group</td>
</tr>
<tr>
<td>C_IROTH</td>
<td>Read by others</td>
</tr>
<tr>
<td>C_IWOTH</td>
<td>Write by others</td>
</tr>
<tr>
<td>C_IXOTH</td>
<td>Execute by others</td>
</tr>
<tr>
<td>C_ISUID</td>
<td>Set user ID</td>
</tr>
<tr>
<td>C_ISGID</td>
<td>Set group ID</td>
</tr>
<tr>
<td>C_ISVTX</td>
<td>On directories, restricted deletion flag</td>
</tr>
<tr>
<td>C_ISDIR</td>
<td>Directory</td>
</tr>
<tr>
<td>C_ISFIFO</td>
<td>FIFO</td>
</tr>
<tr>
<td>C_ISREG</td>
<td>Regular file</td>
</tr>
<tr>
<td>C_ISBLK</td>
<td>Block special</td>
</tr>
<tr>
<td>C_ISCHR</td>
<td>Character special</td>
</tr>
<tr>
<td>C_ISCTG</td>
<td>Reserved</td>
</tr>
<tr>
<td>C_ISLNK</td>
<td>Symbolic link</td>
</tr>
<tr>
<td>C_ISSOCK</td>
<td>Socket</td>
</tr>
</tbody>
</table>

The header defines the symbolic constant:

```
MAGIC  "070707"
```

Attributes See attributes(5) for descriptions of the following attributes:
## Interface Stability

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**  
*pax(1), attributes(5), standards(5)*
Name  dirent.h, dirent – format of directory entries

Synopsis  #include <dirent.h>

Description  The internal format of directories is unspecified. The <dirent.h> header defines the following type:

\[\begin{array}{l}
\text{DIR} & \text{A type representing a directory stream.} \\
\end{array}\]

The header also defines the structure dirent, which includes the following members:

\[\begin{array}{ll}
\text{ino_t d_ino} & /* file serial number */ \\
\text{char d_name[]} & /* name of entry */ \\
\end{array}\]

The type ino_t is defined as described in <sys/types.h>. See types(3HEAD).

The character array d_name is of unspecified size, but the number of bytes preceding the terminating null byte must not exceed \{NAME_MAX\}.

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

\[\begin{array}{|c|c|}
\hline
\text{ATTRIBUTE TYPE} & \text{ATTRIBUTE VALUE} \\
\hline
\text{Interface Stability} & \text{Standard} \\
\hline
\end{array}\]

See Also  closedir(3C), opendir(3C), readdir(3C), rewinddir(3C), seekdir(3C), telldir(3C), types.h(3HEAD), attributes(5), standards(5)
#include <errno.h>

The <errno.h> header provides a declaration for errno and gives positive values for the symbolic constants listed on the Intro(2) manual page.

Values for errno are required to be distinct positive values rather than non-zero values.

See also Intro(2), attributes(5), standards(5)
The `<fcntl.h>` header defines the following requests and arguments for use by the functions `fcntl(2)`, `open(2)`, and `openat(2)`.

Values for `cmd` used by `fcntl()` (the following values are unique):

- `F_DUPFD` - Duplicate file descriptor.
- `F_DUP2FD` - Similar to `F_DUPFD`, but always returns `arg`.
- `F_GETFD` - Get file descriptor flags.
- `F_SETFD` - Set file descriptor flags.
- `F_GETFL` - Get file status flags.
- `F_SETFL` - Set file status flags.
- `F_GETOWN` - Get processor process group ID to receive `SIGURG` signals.
- `F_SETOWN` - Set processor process group ID to receive `SIGURG` signals.
- `F_FREESP` - Free storage space associated with a section of the ordinary file `fildes`.
- `F_GETLK` - Get record locking information.
- `F_GETLK64` - Equivalent to `F_GETLK`, but takes a `struct flock64` argument rather than a `struct flock` argument.
- `F_SETLK` - Set record locking information.
- `F_SETLK64` - Equivalent to `F_SETLK`, but takes a `struct flock64` argument rather than a `struct flock` argument.
- `F_SETLKW` - Set record locking information; wait if blocked.
- `F_SETLKW64` - Equivalent to `F_SETLKW`, but takes a `struct flock64` argument rather than a `struct flock` argument.
- `F_SHARE` - Set share reservation.
- `F_UNSHARE` - Remove share reservation.

File descriptor flags used for `fcntl()`:

- `FD_CLOEXEC` - Close the file descriptor upon execution of an `exec` function (see `exec(2)`).

Values for `l_type` used for record locking with `fcntl()` (the following values are unique):

- `F_RDLCK` - Shared or read lock.
F_UNLOCK       Unlock.
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_NOCCTTY      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 openat():
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 */
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 */`

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**

creatin(2), execin(2), fcntlin(2), openin(2), fdatasyncin(3RT), fsyncin(3C), fsattrin(5), attributes(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 fdatasyncin(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 fsyncin(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.
The `<fenv.h>` header defines the following data types through `typedef`:

- `fenv_t`: Represents the entire floating-point environment. The floating-point environment refers collectively to any floating-point status flags and control modes supported by the implementation.

- `fexcept_t`: Represents the floating-point status flags collectively, including any status the implementation associates with the flags. A floating-point status flag is a system variable whose value is set (but never cleared) when a floating-point exception is raised, which occurs as a side effect of exceptional floating-point arithmetic to provide auxiliary information. A floating-point control mode is a system variable whose value can be set by the user to affect the subsequent behavior of floating-point arithmetic.

The `<fenv.h>` header defines the following constants if and only if the implementation supports the floating-point exception by means of the floating-point functions `feclearexcept()`, `fegetexceptflag()`, `fearaiseexcept()`, `fesetexceptflag()`, and `fetestexcept()`. Each expands to an integer constant expression with values such that bitwise-inclusive ORs of all combinations of the constants result in distinct values.

- `FE_DIVBYZERO`
- `FE_INEXACT`
- `FE_INVALID`
- `FE_OVERFLOW`
- `FE_UNDERFLOW`

The `<fenv.h>` header defines the following constant, which is simply the bitwise-inclusive OR of all floating-point exception constants defined above:

- `FE_ALL_EXCEPT`

The `<fenv.h>` header defines the following constants. Each expands to an integer constant expression whose values are distinct non-negative values.

- `FE_DOWNWARD`
- `FE_TONEAREST`
- `FE_TOWARDZERO`
- `FE_UPWARD`

The `<fenv.h>` header defines the following constant, which represents the default floating-point environment (that is, the one installed at program startup) and has type pointer to const-qualified `fenv_t`. It can be used as an argument to the functions within the `<fenv.h>` header that manage the floating-point environment.

- `FE_DFL_ENV`
The FENV_ACCESS pragma provides a means to inform the implementation when an application might access the floating-point environment to test floating-point status flags or run under non-default floating-point control modes. The pragma occurs either outside external declarations or preceding all explicit declarations and statements inside a compound statement. When outside external declarations, the pragma takes effect from its occurrence until another FENV_ACCESS pragma is encountered, or until the end of the translation unit. When inside a compound statement, the pragma takes effect from its occurrence until another FENV_ACCESS pragma is encountered (including within a nested compound statement), or until the end of the compound statement; at the end of a compound statement the state for the pragma is restored to its condition just before the compound statement. If this pragma is used in any other context, the behavior is undefined.

If part of an application tests floating-point status flags, sets floating-point control modes, or runs under non-default mode settings, but was translated with the state for the FENV_ACCESS pragma off, the behavior is undefined. The default state (on or off) for the pragma is implementation-defined. (When execution passes from a part of the application translated with FENV_ACCESS off to a part translated with FENV_ACCESS on, the state of the floating-point status flags is unspecified and the floating-point control modes have their default settings.)

**Usage**

This header is designed to support the floating-point exception status flags and directed-rounding control modes required by the IEC 60559: 1989 standard, and other similar floating-point state information. Also, it is designed to facilitate code portability among all systems. Certain application programming conventions support the intended model of use for the floating-point environment:

- A function call does not alter its caller’s floating-point control modes, clear its caller’s floating-point status flags, or depend on the state of its caller’s floating-point status flags unless the function is so documented.
- A function call is assumed to require default floating-point control modes, unless its documentation promises otherwise.
- A function call is assumed to have the potential for raising floating-point exceptions, unless its documentation promises otherwise.

With these conventions, an application can safely assume default floating-point control modes (or be unaware of them). The responsibilities associated with accessing the floating-point environment fall on the application that does so explicitly.

Even though the rounding direction macros might expand to constants corresponding to the values of FLT_ROUNDS, they are not required to do so. For example:

```c
#include <fenv.h>
void f(double x)
{
    #pragma STDC FENV_ACCESS ON
    void g(double);
}
void h(double);
/* ... */
g(x + 1);
h(x + 1);
/* ... */
}

If the function \texttt{g}() might depend on status flags set as a side effect of the first \texttt{x+1}, or if the second \texttt{x+1} might depend on control modes set as a side effect of the call to function \texttt{g}(), then the application must contain an appropriately placed invocation as follows:

\begin{verbatim}
#pragma STDC FENV_ACCESS ON
\end{verbatim}

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**  feclearexcept\((3M)\), fegetenv\((3M)\), fegetexceptflag\((3M)\), fegetround\((3M)\), feholdexcept\((3M)\), feraiseexcept\((3M)\), fesetenv\((3M)\), fesetexceptflag\((3M)\), fesetround\((3M)\), fetestexcept\((3M)\), feupdateenv\((3M)\), attributes\((5)\), standards\((5)\)
The characteristics of floating types are defined in terms of a model that describes a representation of floating-point numbers and values that provide information about an implementation’s floating-point arithmetic.

The following parameters are used to define the model for each floating-point type:

- **s** sign (±1)
- **b** base or radix of exponent representation (an integer >1)
- **e** exponent (an integer between a minimum \( e_{\text{min}} \) and a maximum \( e_{\text{max}} \))
- **p** precision (the number of base-\( b \) digits in the significand)
- \( f_k \) non-negative integers less than \( b \) (the significand digits)

In addition to normalized floating-point numbers (\( f_i > 0 \) if \( x \neq 0 \)), floating types might be able to contain other kinds of floating-point numbers, such as subnormal floating-point numbers (\( x \neq 0, e = e_{\text{min}}, f_i = 0 \)) and unnormalized floating-point numbers (\( x \neq 0, e = e_{\text{min}}, f_i = 0 \)), and values that are not floating-point numbers, such as infinities and NaNs. A **quiet NaN** propagates through almost every arithmetic operation without raising a floating-point exception; a **signaling NaN** generally raises a floating-point exception when occurring as an arithmetic operand.

The accuracy of the library functions in `math.h` and `complex.h` that return floating-point results is defined on the `libm` manual page.

All integer values in the `<float.h>` header, except FLT_ROUNDS, are constant expressions suitable for use in #if preprocessing directives; all floating values are constant expressions. All except DECIMAL_DIG, FLT_EVAL_METHOD, FLT_RADIX, and FLT_ROUNDS have separate names for all three floating-point types. The floating-point model representation is provided for all values except FLT_EVAL_METHOD and FLT_ROUNDS.

The rounding mode for floating-point addition is characterized by the value of FLT_ROUNDS:

- \(-1\) Indeterminable.
- \(0\) Toward zero.
- \(1\) To nearest.
- \(2\) Toward positive infinity.
- \(3\) Toward negative infinity.

The values of operations with floating operands and values subject to the usual arithmetic conversions and of floating constants are evaluated to a format whose range and precision
might be greater than required by the type. The use of evaluation formats is characterized by
the architecture-dependent value of FLT_EVAL_METHOD:

- 1  Indeterminable.
0  Evaluate all operations and constants just to the range and precision of the type.
1  Evaluate operations and constants of type float and double to the range and precision
   of the double type; evaluate long double operations and constants to the range and
   precision of the long double type.
2  Evaluate all operations and constants to the range and precision of the long double
type.

The values given in the following list are defined as constants.

- Radix of exponent representation, \( b \).
  
  FLT_RADIX

- Number of base-FLT_RADIX digits in the floating-point significand, \( p \).
  
  FLT_MANT_DIG
  DBL_MANT_DIG
  LDBL_MANT_DIG

- Number of decimal digits, \( n \), such that any floating-point number in the widest supported
  floating type with \( p_{\text{max}} \) radix \( b \) digits can be rounded to a floating-point number with \( n \)
  decimal digits and back again without change to the value.
  
  DECIMAL_DIG

- Number of decimal digits, \( q \), such that any floating-point number with \( q \) decimal digits can
  be rounded into a floating-point number with \( p \) radix \( b \) digits and back again without
  change to the \( q \) decimal digits.
  
  FLT_DIG
  DBL_DIG
  LDBL_DIG

- Minimum negative integer such that FLT_RADIX raised to that power minus 1 is a
  normalized floating-point number, \( e_{\text{min}} \).
  
  FLT_MIN_EXP
  DBL_MIN_EXP
  LDBL_MIN_EXP

- Minimum negative integer such that 10 raised to that power is in the range of normalized
  floating-point numbers.
  
  FLT_MIN_10_EXP
  DBL_MIN_10_EXP
  LDBL_MIN_10_EXP
- Maximum integer such that `FLT_RADIX` raised to that power minus 1 is a representable finite floating-point number, \(e_{\text{max}}\).
  
  - `FLT_MAX_EXP`
  - `DBL_MAX_EXP`
  - `LDBL_MAX_EXP`

- Maximum integer such that 10 raised to that power is in the range of representable finite floating-point numbers.
  
  - `FLT_MAX_10_EXP`
  - `DBL_MAX_10_EXP`
  - `LDBL_MAX_10_EXP`

The values given in the following list are defined as constant expressions with values that are greater than or equal to those shown:

- Maximum representable finite floating-point number.
  
  - `FLT_MAX`
  - `DBL_MAX`
  - `LDBL_MAX`

The values given in the following list are defined as constant expressions with implementation-defined (positive) values that are less than or equal to those shown:

- The difference between 1 and the least value greater than 1 that is representable in the given floating-point type, \(b^{1-p}\).
  
  - `FLT_EPSILON`
  - `DBL_EPSILON`
  - `LDBL_EPSILON`

- Minimum normalized positive floating-point number, \(b^{e_{\text{min}}-1}\).
  
  - `FLT_MIN`
  - `DBL_MIN`
  - `LDBL_MIN`

**Attributes**
see attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**
complex.h(3HEAD), math.h(3HEA), attributes(5), standards(5)
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 libc. The included header file <sys/ieeefp.h> defines certain types of interest to the kernel.

### IEEE Rounding Modes
- **fp_direction_type**: The type of the IEEE rounding direction mode. Note: the order of enumeration varies according to hardware.
- **fp_precision_type**: 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:
  - **sigfpe_code_type**: The type of a SIGFPE code.
  - **sigfpe_handler_type**: The type of a user-definable SIGFPE exception handler called to handle a particular SIGFPE code.
- **SIGFPE_DEFAULT**: 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 abort(3C).
- **SIGFPE_IGNORE**: A macro indicating an alternate SIGFPE exception handling, namely to ignore and continue execution.
- **SIGFPE_ABORT**: A macro indicating an alternate SIGFPE exception handling, namely to abort with a core dump.

### IEEE Exception Handling
- **N_IEEE_EXCEPTION**: The number of distinct IEEE floating-point exceptions.
- **fp_exception_type**: The type of the N_IEEE_EXCEPTION exceptions. Each exception is given a bit number.
- **fp_exception_field_type**: The type intended to hold at least N_IEEE_EXCEPTION bits corresponding to the IEEE exceptions numbered by fp_exception_type. Thus fp_inexact corresponds to the least significant bit and fp_invalid to the fifth least significant bit. Note: some operations may set more than one exception.

### IEEE Formats and Classification
- **single; extended; quadruple**
- **fp_class_type**: An enumeration of the various classes of IEEE values and symbols.

### IEEE Base Conversion
The functions described under floating_to_decimal(3C) and decimal_to_floating(3C) satisfy not only the IEEE Standard, but also the stricter requirements of correct rounding for all arguments.
DECIMAL_STRING_LENGTH  The length of a decimal_string.
decimal_string  The digit buffer in a decimal_record.
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/ieeefp.h

See Also  abort(3C), decimal_to_floating(3C), econvert(3C), floating_to_decimal(3C),
sigfpe(3C), string_to_decimal(3C), strtod(3C)
The `fmtmsg.h` header defines the following macros, which expand to constant integer expressions:

- **MM_HARD**: Source of the condition is hardware.
- **MM_SOFT**: Source of the condition is software.
- **MM_FIRM**: Source of the condition is firmware.
- **MM_APPL**: Condition detected by application.
- **MM_UTIL**: Condition detected by utility.
- **MM_OPSYS**: Condition detected by operating system.
- **MM_RECOVER**: Recoverable error.
- **MM_NRECOV**: Non-recoverable error.
- **MM_HALT**: Error causing application to halt.
- **MM_ERROR**: Application has encountered a non-fatal fault.
- **MM_WARNING**: Application has detected unusual non-error condition.
- **MM_INFO**: Informative message.
- **MM_NOSEV**: No severity level provided for the message.
- **MM_PRINT**: Display message on standard error.
- **MM_CONSOLE**: Display message on system console.

The table below indicates the null values and identifiers for `fmtmsg(3C)` arguments. The `fmtmsg.h` header defines the macros in the Identifier column, which expand to constant expressions that expand to expressions of the type indicated in the Type column:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Null-Value</th>
<th>Identifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>label</td>
<td>char*</td>
<td>(char*) NULL</td>
<td>MM_NULLLBL</td>
</tr>
<tr>
<td>severity</td>
<td>int</td>
<td>0</td>
<td>MM_NULLSEV</td>
</tr>
<tr>
<td>class</td>
<td>long</td>
<td>0L</td>
<td>MM_NULLMC</td>
</tr>
<tr>
<td>text</td>
<td>char*</td>
<td>(char*) NULL</td>
<td>MM_NULLTXT</td>
</tr>
<tr>
<td>action</td>
<td>char*</td>
<td>(char*) NULL</td>
<td>MM_NULLACT</td>
</tr>
<tr>
<td>tag</td>
<td>char*</td>
<td>(char*) NULL</td>
<td>MM_NULLTAG</td>
</tr>
</tbody>
</table>
The `<fmtmsg.h>` header also defines the following macros for use as return values for `fmtmsg()`:

- **MM_OK**: The function succeeded.
- **MM_NOTOK**: The function failed completely.
- **MM_NOMSG**: The function was unable to generate a message on standard error, but otherwise succeeded.
- **MM_NOCON**: The function was unable to generate a console message, but otherwise succeeded.

### Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

### See Also

`fmtmsg(3C), attributes(5), standards(5)`
Name fnmatch.h, fnmatch – filename-matching types

Synopsis #include <fnmatch.h>

Description The <fnmatch.h> header defines the following constants:

- FNM_NOMATCH The string does not match the specified pattern.
- FNM_PATHNAME Slash in string only matches slash in pattern.
- FNM_PERIOD Leading period in string must be exactly matched by period in pattern.
- FNM_NOESCAPE Disable backslash escaping.
- FNM_NOSYS Reserved.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also fnmatch(3C), attributes(5), standards(5)
**Name**
ftw.h, ftw – file tree traversal

**Synopsis**
#include <ftw.h>

**Description**
The `<ftw.h>` header defines the FTW structure that includes the following members:

```c
int base
int level
```

The `<ftw.h>` header defines macros for use as values of the third argument to the application-supplied function that is passed as the second argument to `ftw()` and `nftw()` (see `ftw(3C)`):

- **FTW_F** file
- **FTW_D** directory
- **FTW_DNR** directory without read permission
- **FTW_DP** directory with subdirectories visited
- **FTW_NS** unknown type; `stat()` failed
- **FTW_SL** symbolic link
- **FTW_SLN** symbolic link that names a nonexistent file

The `<ftw.h>` header defines macros for use as values of the fourth argument to `nftw()`:

- **FTW_PHYS** Physical walk, does not follow symbolic links. Otherwise, `nftw()` follows links but does not walk down any path that crosses itself.
- **FTW_MOUNT** The walk does not cross a mount point.
- **FTW_DEPTH** All subdirectories are visited before the directory itself.
- **FTW_CHDIR** The walk changes to each directory before reading it.

The `<ftw.h>` header defines the `stat` structure and the symbolic names for `st_mode` and the file type test macros as described in `<sys/stat.h>`. Inclusion of the `<ftw.h>` header might also make visible all symbols from `<sys/stat.h>`.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**
`ftw(3C), stat.h(3HEAD), attributes(5), standards(5)`
glob.h (3HEAD)

**Name**
glob.h, glob – pathname pattern-matching types

**Synopsis**
#include <glob.h>

**Description**
The <glob.h> header defines the structures and symbolic constants used by the glob(3).

The structure type `glob_t` contains the following members:

- `size_t gl_pathc /* count of paths matched by pattern */`
- `char **gl_pathv /* pointer to a list of matched pathnames */`
- `size_t gl_offs /* lots to reserve at the beginning of gl_pathv */`

The following constants are provided as values for the `flags` argument:

- **GLOB_APPEND** Append generated pathnames to those previously obtained.
- **GLOB_DOOFFS** Specify how many null pointers to add to the beginning of `gl_pathv`.
- **GLOB_ERR** Cause `glob()` to return on error.
- **GLOB_MARK** Each pathname that is a directory that matches pattern has a slash appended.
- **GLOB_NOCHECK** If pattern does not match any pathname, then return a list consisting of only pattern.
- **GLOB_NOESCAPE** Disable backslash escaping.
- **GLOB_NOSORT** Do not sort the pathnames returned.

The following constants are defined as error return values:

- **GLOB_ABORTED** The scan was stopped because GLOB_ERR was set or (*errfunc)() returned non-zero.
- **GLOB_NOMATCH** The pattern does not match any existing pathname, and GLOB_NOCHECK was not set in flags.
- **GLOB_NOSPACE** An attempt to allocate memory failed.
- **GLOB_NOSYS** Reserved.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>
See Also  glob(3C), attributes(5), standards(5)
The `<grp.h>` header declares the structure `group`, which includes the following members:

```c
char *gr_name /* name of the group */
gid_t gr_gid /* numerical group ID */
char **gr_mem /* pointer to a null-terminated array of
    character pointers to member names */
```

The `gid_t` type is defined as described in `<sys/types.h>` (see `types(3HEAD)`).

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**

`getgrnam(3C), types.h(3HEAD), attributes(5), standards(5)`
iconv.h(3HEAD)

Name  iconv.h, iconv – codeset conversion facility

Synopsis  #include <iconv.h>

Description  The <iconv.h> header defines the following type:

iconv_t  Identifies the conversion from one codeset to another.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also  iconv(3C), iconv_close(3C), iconv_open(3C), attributes(5), standards(5)
Name  if.h, if – sockets local interfaces

Synopsis  
```
#include <net/if.h>
```

Description  The `<net/if.h>` header defines the `if_nameindex` structure, which includes the following members:
```
unsigned if_index /* numeric index of the interface */
char *if_name /* null-terminated name of the interface */
```

The `<net/if.h>` header defines the following macro for the length of a buffer containing an interface name (including the terminating null character):
```
IF_NAMESIZE  interface name length
```

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also  `if_nametoindex(3XNET), attributes(5), standards(5)`
The `<arpa/inet.h>` header defines the type `in_port_t`, the type `in_addr_t`, and the `in_addr` structure, as described in `in.h(3HEAD)`. Inclusion of the `<arpa/inet.h>` header may also make visible all symbols from `in.h(3HEAD)`.

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

```
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);
```

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 may be declared as functions, or defined as macros, or both:

```
uint32_t htonl(uint32_t);
uint16_t htons(uint16_t);
uint32_t ntohl(uint32_t);
uint16_t ntohs(uint16_t);
```

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 may be declared as functions, or defined as macros, or both:

```
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);
```

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

```
ATTRIBUTES TYPE ATTRIBUTE VALUE
Interface Stability Standard
```

See Also `Intro(3)`, `htonl(3SOCKET)`, `htonl(3XNET)`, `inet_addr(3SOCKET)`, `inet_addr(3XNET)`, `in.h(3HEAD)`, attributes(5), standards(5)
in.h(3HEAD)

Name  in.h, in – Internet Protocol family

Synopsis  #include <netinet/in.h>

Description  The <netinet/in.h> header defines the following types through typedef:

    in_port_t  An unsigned integral type of exactly 16 bits.
    in_addr_t  An unsigned integral type of exactly 32 bits. The <netinet/in.h> header defines the in_addr structure that includes the following member:

    in_addr_t  s_addr

The <netinet/in.h> header defines the in_addr structure that includes the following member:

    in_addr_t  s_addr

The <netinet/in.h> header defines the type sa_family_t as described in socket.h(3HEAD).

The <netinet/in.h> header defines the following macros for use as values of the level argument of getsockopt() and setsockopt():

    IPPROTO_IP     Dummy for IP
    IPPROTO_ICMP   Control message protocol
    IPPROTO_TCP    TCP
    IPPROTO_UDP    User datagram protocol

The <netinet/in.h> header defines the following macros for use as destination addresses for connect(), sendmsg(), and sendto():

    INADDR_ANY     Local host address
    INADDR_BROADCAST Broadcast address

The <netinet/in.h> header defines the sockaddr_in structure that is used to store addresses for the Internet protocol family. Values of this type must be cast to struct sockaddr for use with the socket interfaces.

Default  For applications that do not require standard-conforming behavior (those that use the socket interfaces described in section (3SOCKET) 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
    char         sin_zero[8]
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]`

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**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.h(3HEAD)*, *attributes(5)*, *standards(5)*
The `<inttypes.h>` header includes the `<stdint.h>` header. The `<inttypes.h>` header includes a definition of the following type:

`imaxdiv_t` structure type that is the type of the value returned by the `imaxdiv()` function.

The following macros are defined. Each expands to a character string literal containing a conversion specifier, possibly modified by a length modifier, suitable for use within the format argument of a formatted input/output function when converting the corresponding integer type. These macros have the general form of `PRI` (character string literals for the `fprintf()` and `fwprintf()` family of functions) or `SCN` (character string literals for the `fscanf()` and `fscanf()` family of functions), followed by the conversion specifier, followed by a name corresponding to a similar type name in `<stdint.h>`. In these names, `N` represents the width of the type as described in `<stdint.h>`. For example, `PRIdFAST32` can be used in a format string to print the value of an integer of type `int_fast32_t`.

The `fprintf()` macros for signed integers are:

- `PRI%d`  `PRI%dN`  `PRI%dLEASTN`  `PRI%dFASTN`  `PRI%dMAX`  `PRI%dPTR`
- `PRI%i`  `PRI%iN`  `PRI%iLEASTN`  `PRI%iFASTN`  `PRI%iMAX`  `PRI%iPTR`

The `fprintf()` macros for unsigned integers are:

- `PRIu`  `PRIuN`  `PRIuLEASTN`  `PRIuFASTN`  `PRIuMAX`  `PRIuPTR`
- `PRIx`  `PRIxN`  `PRIxLEASTN`  `PRIxFASTN`  `PRIxMAX`  `PRIxPTR`
- `PRIX`  `PRIXN`  `PRIXLEASTN`  `PRIXFASTN`  `PRIXMAX`  `PRIXPTR`

The `fscanf()` macros for signed integers are:

- `SCNd`  `SCNdN`  `SCNdLEASTN`  `SCNdFASTN`  `SCNdMAX`  `SCNdPTR`
- `SCNi`  `SCNiN`  `SCNiLEASTN`  `SCNiFASTN`  `SCNiMAX`  `SCNiPTR`

The `fscanf()` macros for unsigned integers are:

- `SCNu`  `SCNuN`  `SCNuLEASTN`  `SCNuFASTN`  `SCNuMAX`  `SCNuPTR`
- `SCNx`  `SCNxN`  `SCNxLEASTN`  `SCNxFASTN`  `SCNxMAX`  `SCNxPTR`

For each type that the implementation provides in `<stdint.h>`, the corresponding `fprintf()` and `fwprintf()` macros must be defined. The corresponding `fscanf()` and `fscanf()` macros must be defined as well, unless the implementation does not have a suitable modifier for the type.
The purpose of `<inttypes.h>` is to provide a set of integer types whose definitions are consistent across machines and independent of operating systems and other implementation idiosyncrasies. It defines, with a typedef, integer types of various sizes. Implementations are free to typedef them as ISO C standard integer types or extensions that they support. Consistent use of this header greatly increases the portability of applications across platforms.

**Examples**

**EXAMPLE 1  Use of Macro**

The following code uses one of the macros available through `<inttypes.h>`.

```c
#include <inttypes.h>
#include <wchar.h>

int main(void)
{
    uintmax_t i = UINTMAX_MAX; // This type always exists.
    wprintf("The largest integer value is %020" PRIxMAX, "\n", i);
    return 0;
}
```

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**

`imaxdiv(3C), attributes(5), standards(5)`
The `<sys/ipc.h>` header is used by three mechanisms for interprocess communication (IPC): messages, semaphores, and shared memory. All use a common structure type, `ipc_perm`, to pass information used in determining permission to perform an IPC operation.

The `ipc_perm` structure contains the following members:

- `uid_t uid /* owner's user ID */`
- `gid_t gid /* owner's group ID */`
- `uid_t cuid /* creator's user ID */`
- `gid_t cgid /* creator's group ID */`
- `mode_t mode /* read/write permission`

The `uid_t`, `gid_t`, `mode_t`, and `key_t` types are defined as described in `<sys/types.h>`. See `types.h(3HEAD)`.

Definitions are provided for the constants listed below.

Mode bits:

- `IPC_CREAT` Create entry if key does not exist.
- `IPC_EXCL` Fail if key exists.
- `IPC_NOWAIT` Error if request must wait.

Keys:

- `IPC_PRIVATE` Private key.

Control commands:

- `IPC_RMID` Remove identifier.
- `IPC_SET` Set options.
- `IPC_STAT` Get options.

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**

`ftok(3C)`, `types.h(3HEAD)`, `attributes(5)`, `standards(5)`
#include <iso646.h>

The `<iso646.h>` header defines the following macros (on the left) that expand to the corresponding tokens (on the right):

- `and` `&&`
- `and_eq` `&=`
- `bitand` `&`
- `bitor` `|`
- `compl` `~`
- `not` `!`
- `not_eq` `!=`
- `or` `||`
- `or_eq` `|=`
- `xor` `^`
- `xor_eq` `^=`

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also** attributes(5), standards(5)
**Name**  
langinfo.h, langinfo – language information constants

**Synopsis**  
#include <langinfo.h>

**Description**  
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>`.

The following constants are defined. The entries under Category indicate in which `setlocale(3C)` category each item is defined.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Category</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>CODESET</td>
<td>LC_CTYPE</td>
<td>codeset name</td>
</tr>
<tr>
<td>D_FMT</td>
<td>LC_TIME</td>
<td>string for formatting date and time</td>
</tr>
<tr>
<td>D_FMT</td>
<td>LC_TIME</td>
<td>date format string</td>
</tr>
<tr>
<td>T_FMT</td>
<td>LC_TIME</td>
<td>time format string</td>
</tr>
<tr>
<td>T_FMT AMPM</td>
<td>LC_TIME</td>
<td>a.m. or p.m. time format string</td>
</tr>
<tr>
<td>AM_STR</td>
<td>LC_TIME</td>
<td>ante-meridiem affix</td>
</tr>
<tr>
<td>PM_STR</td>
<td>LC_TIME</td>
<td>post-meridiem affix</td>
</tr>
<tr>
<td>DAY_1</td>
<td>LC_TIME</td>
<td>name of the first day of the week (for example, Sunday)</td>
</tr>
<tr>
<td>DAY_2</td>
<td>LC_TIME</td>
<td>name of the second day of the week (for example, Monday)</td>
</tr>
<tr>
<td>DAY_3</td>
<td>LC_TIME</td>
<td>name of the third day of the week (for example, Tuesday)</td>
</tr>
<tr>
<td>DAY_4</td>
<td>LC_TIME</td>
<td>name of the fourth day of the week (for example, Wednesday)</td>
</tr>
<tr>
<td>DAY_5</td>
<td>LC_TIME</td>
<td>name of the fifth day of the week (for example, Thursday)</td>
</tr>
<tr>
<td>DAY_6</td>
<td>LC_TIME</td>
<td>name of the sixth day of the week (for example, Friday)</td>
</tr>
<tr>
<td>DAY_7</td>
<td>LC_TIME</td>
<td>name of the seventh day of the week (for example, Saturday)</td>
</tr>
<tr>
<td>ABDAY_1</td>
<td>LC_TIME</td>
<td>abbreviated name of the first day of the week</td>
</tr>
<tr>
<td>ABDAY_2</td>
<td>LC_TIME</td>
<td>abbreviated name of the second day of the week</td>
</tr>
<tr>
<td>ABDAY_3</td>
<td>LC_TIME</td>
<td>abbreviated name of the third day of the week</td>
</tr>
<tr>
<td>Constant</td>
<td>Category</td>
<td>Meaning</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>ABDAY_4</td>
<td>LC_TIME</td>
<td>abbreviated name of the fourth day of the week</td>
</tr>
<tr>
<td>ABDAY_5</td>
<td>LC_TIME</td>
<td>abbreviated name of the fifth day of the week</td>
</tr>
<tr>
<td>ABDAY_6</td>
<td>LC_TIME</td>
<td>abbreviated name of the seventh day of the week</td>
</tr>
<tr>
<td>ABDAY_7</td>
<td>LC_TIME</td>
<td>abbreviated name of the seventh day of the week</td>
</tr>
<tr>
<td>MON_1</td>
<td>LC_TIME</td>
<td>name of the first month of the year</td>
</tr>
<tr>
<td>MON_2</td>
<td>LC_TIME</td>
<td>name of the second month</td>
</tr>
<tr>
<td>MON_3</td>
<td>LC_TIME</td>
<td>name of the third month</td>
</tr>
<tr>
<td>MON_4</td>
<td>LC_TIME</td>
<td>name of the fourth month</td>
</tr>
<tr>
<td>MON_5</td>
<td>LC_TIME</td>
<td>name of the fifth month</td>
</tr>
<tr>
<td>MON_6</td>
<td>LC_TIME</td>
<td>name of the sixth month</td>
</tr>
<tr>
<td>MON_7</td>
<td>LC_TIME</td>
<td>name of the seventh month</td>
</tr>
<tr>
<td>MON_8</td>
<td>LC_TIME</td>
<td>name of the eighth month</td>
</tr>
<tr>
<td>MON_9</td>
<td>LC_TIME</td>
<td>name of the ninth month</td>
</tr>
<tr>
<td>MON_10</td>
<td>LC_TIME</td>
<td>name of the tenth month</td>
</tr>
<tr>
<td>MON_11</td>
<td>LC_TIME</td>
<td>name of the eleventh month</td>
</tr>
<tr>
<td>MON_12</td>
<td>LC_TIME</td>
<td>name of the twelfth month</td>
</tr>
<tr>
<td>ABMON_1</td>
<td>LC_TIME</td>
<td>abbreviated name of the first month</td>
</tr>
<tr>
<td>ABMON_2</td>
<td>LC_TIME</td>
<td>abbreviated name of the second month</td>
</tr>
<tr>
<td>ABMON_3</td>
<td>LC_TIME</td>
<td>abbreviated name of the third month</td>
</tr>
<tr>
<td>ABMON_4</td>
<td>LC_TIME</td>
<td>abbreviated name of the fourth month</td>
</tr>
<tr>
<td>ABMON_5</td>
<td>LC_TIME</td>
<td>abbreviated name of the fifth month</td>
</tr>
<tr>
<td>ABMON_6</td>
<td>LC_TIME</td>
<td>abbreviated name of the sixth month</td>
</tr>
<tr>
<td>ABMON_7</td>
<td>LC_TIME</td>
<td>abbreviated name of the seventh month</td>
</tr>
<tr>
<td>ABMON_8</td>
<td>LC_TIME</td>
<td>abbreviated name of the eighth month</td>
</tr>
<tr>
<td>ABMON_9</td>
<td>LC_TIME</td>
<td>abbreviated name of the ninth month</td>
</tr>
<tr>
<td>ABMON_10</td>
<td>LC_TIME</td>
<td>abbreviated name of the tenth month</td>
</tr>
<tr>
<td>ABMON_11</td>
<td>LC_TIME</td>
<td>abbreviated name of the eleventh month</td>
</tr>
<tr>
<td>ABMON_12</td>
<td>LC_TIME</td>
<td>abbreviated name of the twelfth month</td>
</tr>
</tbody>
</table>
### Constant Category Meaning

<table>
<thead>
<tr>
<th>Constant</th>
<th>Category</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERA</td>
<td>LC_TIME</td>
<td>era description segments</td>
</tr>
<tr>
<td>ERA_D_FMT</td>
<td>LC_TIME</td>
<td>era date format string</td>
</tr>
<tr>
<td>ERA_D_T_FMT</td>
<td>LC_TIME</td>
<td>era date and time format string</td>
</tr>
<tr>
<td>ERA_T_FMT</td>
<td>LC_TIME</td>
<td>era time format string</td>
</tr>
<tr>
<td>ALT_DIGITS</td>
<td>LC_TIME</td>
<td>alternative symbols for digits</td>
</tr>
<tr>
<td>RADIXCHAR</td>
<td>LC_NUMERIC</td>
<td>radix character</td>
</tr>
<tr>
<td>THOUSEP</td>
<td>LC_NUMERIC</td>
<td>separator for thousands</td>
</tr>
<tr>
<td>YESEXPR</td>
<td>LC_MESSAGES</td>
<td>affirmative response expression</td>
</tr>
<tr>
<td>NOEXPR</td>
<td>LC_MESSAGES</td>
<td>negative response expression</td>
</tr>
<tr>
<td>YESSTR</td>
<td>LC_MESSAGES</td>
<td>affirmative response for yes/no queries</td>
</tr>
<tr>
<td>NOSTR</td>
<td>LC_MESSAGES</td>
<td>negative response for yes/no queries</td>
</tr>
<tr>
<td>CRNCYSTR</td>
<td>LC_MONETARY</td>
<td>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</td>
</tr>
</tbody>
</table>

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:

```c
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 strftime(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.

### Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>
See Also  mkmsgs(1), localeconv(3C), nl_langinfo(3C), nl_types.h(3HEAD), setlocale(3C), strftime(3C), attributes(5), standards(5)
libadm — general administrative library

**Synopsis**
cc [ flag... ] file... -ladm [ library... ]

**Description**
Functions in this library provide device management, VTOC handling, regular expressions, and packaging routines.

**Interfaces**
The shared object `libadm.so.1` provides the public interfaces defined below. See `Intro(3)` for additional information on shared object interfaces.

```c
circf loc1 loc2 locs nbra pkgdir
read_extvtoc read_vtoc
write_extvtoc
write_vtoc
```

**Files**
`/lib/libadm.so.1` shared object
`/lib/64/libadm.so.1` 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

**See Also**
`pvs(1), Intro(3), read_vtoc(3EXT), attributes(5), regexp(5)`
libaio – asynchronous I/O library

**Synopsis**

```
cc [ flag... ] file... -laiot [ library... ]
```

**Description**

 Functions in this library perform asynchronous I/O operations.

**Interfaces**

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

- `aiocancel`
- `aioread`
- `aiowait`
- `aiowrite`
- `assfail`
- `close`
- `fork`
- `sigaction`

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

- `aioread64`
- `aiowrite64`

**Files**

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

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

**See Also**

`pvs(1), Intro(2), Intro(3), aiocancel(3AIO), aioread(3AIO), aiowait(3AIO), aiowrite(3AIO), aio.h(3HEAD), attributes(5)"
libauto_ef(3LIB)

**Name**
libauto_ef – auto encoding finder library

**Synopsis**
```bash
cc [ flag... ] file... -lauto_ef [ library... ]
#include <auto_ef.h>
```

**Description**
Functions in this library provide automatic encoding identification.

**Interface Level**
The shared object `libauto_ef.so.1` provides the public interfaces defined below. See *Intro(3)* for additional information on shared object interfaces.

- `auto_ef_file`
- `auto_ef_free`
- `auto_ef_get_encoding`
- `auto_ef_get_score`
- `auto_ef_str`

**Files**
- `/usr/lib/libauto_ef.so.1` shared object
- `/usr/lib/64/libauto_ef.so.1` 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWautoef (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWautoefx (64-bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Stable</td>
</tr>
<tr>
<td>MT-Level</td>
<td>MT-Safe</td>
</tr>
</tbody>
</table>

**See Also**
`auto_ef(1), auto_ef(3EXT), attributes(5)`

*International Language Environments Guide*
libbsdmalloc(3LIB)

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.so.1 shared object
/usr/lib/64/libbsdmalloc.so.1 64–bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32–bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64–bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

See Also pvs(1), Intro(3), bsdmalloc(3MALLOC), attributes(5)
Name    libbsm – basic security library

Synopsis  cc [ flag... ] file. -lbsm [ library... ]

Description Functions in this library provide basic security, library object reuse, and auditing.

Interfaces The shared object libbsm.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

au_close au_open
au_preselect au_to_arg
au_to_arg32 au_to_arg64
au_to_attr au_to_cmd
au_to_data au_to_groups
au_to_in_addr au_to_ipc
au_to_iport au_to_me
au_to_newgroups au_to_opaque
au_to_path au_to_process
au_to_process_ex au_to_return
au_to_return32 au_to_return64
au_to_socket au_to_subject
au_to_subject_ex au_to_text
au_user_mask au_write
audit auditon
auditsvc endac
endauclass endauevent
endausuer getacdir
getacflg getacmin
getacna getauclass
getauclassent_r getauclassnam
getauclassnam_r getaudit
getaudit_addr getauditflagsbin
getauditflagschar getauevent
getauevent_r          getauevnam
getauevnam_r          getauevnonam
getauevnum            getauevnum_r
getauid               getauuserent
getauuserent_r        getaauusernam
getaauusernam_r       getfauditflags
setac                 setauclass
setauclassfile        setauaudit
setaudit_addr         setauevent
setaueventfile        setauuid
setauuser             setauuserfile
testac

**Files**
/lib/libbsm.so.1     shared object
/lib/64/libbsm.so.1   64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcs (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcsx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>See individual man page for each function.</td>
</tr>
</tbody>
</table>

**See Also** pvs(1), Intro(3), attributes(5)
Name  libc – C library

Description  Functions in this library provide various facilities defined by System V, ANSI C, POSIX, and so on. See standards(5). 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, libthread and libpthread.

Interfaces  The shared object libc.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.
libc(3LIB)

Library Interfaces and Headers 85
<table>
<thead>
<tr>
<th>Function</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>_getcwd</td>
<td>_getdate</td>
</tr>
<tr>
<td>_getdate_err</td>
<td>_getdate_err_addr</td>
</tr>
<tr>
<td>_getegid</td>
<td>_geteuid</td>
</tr>
<tr>
<td>_getexecname</td>
<td>_getgid</td>
</tr>
<tr>
<td>_getgrgid</td>
<td>_getgrnam</td>
</tr>
<tr>
<td>_getgroups</td>
<td>_getitimer</td>
</tr>
<tr>
<td>_getlogin</td>
<td>_getmsg</td>
</tr>
<tr>
<td>_getopt</td>
<td>_getpass</td>
</tr>
<tr>
<td>_getpgid</td>
<td>_getpgrp</td>
</tr>
<tr>
<td>_getpid</td>
<td>_getpmsg</td>
</tr>
<tr>
<td>_getppid</td>
<td>_getprojid</td>
</tr>
<tr>
<td>_getpwnam</td>
<td>_getpwuid</td>
</tr>
<tr>
<td>_getrlimit</td>
<td>_getsid</td>
</tr>
<tr>
<td>_getsubopt</td>
<td>_gettaskid</td>
</tr>
<tr>
<td>_gettimeofday</td>
<td>_gettxt</td>
</tr>
<tr>
<td>_getuid</td>
<td>_getw</td>
</tr>
<tr>
<td>_grantpt</td>
<td>_hcreate</td>
</tr>
<tr>
<td>_hdestroy</td>
<td>_hsearch</td>
</tr>
<tr>
<td>_initgroups</td>
<td>_insque</td>
</tr>
<tr>
<td>_iob</td>
<td>_ioctl</td>
</tr>
<tr>
<td>_isascii</td>
<td>_isastream</td>
</tr>
<tr>
<td>_isatty</td>
<td>_isnan</td>
</tr>
<tr>
<td>_isnan</td>
<td>_kill</td>
</tr>
<tr>
<td>_lchown</td>
<td>_lfind</td>
</tr>
<tr>
<td>_link</td>
<td>_lockf</td>
</tr>
<tr>
<td>_longjmp</td>
<td>_lsearch</td>
</tr>
<tr>
<td>_lseek</td>
<td>_lstat</td>
</tr>
<tr>
<td>_lwp_cond_broadcast</td>
<td>_lwp_cond_reltimedwait</td>
</tr>
<tr>
<td>_lwp_cond_signal</td>
<td>_lwp_cond_timedwait</td>
</tr>
</tbody>
</table>
libc(3LIB)

profil ptrace
ptsname putacct
putenv putmsg
putpmsg putw
read readdir
readlink readdir
rename
resolvepath readdir
rmdir rmdir
_rw_write_held rwlock_destroy
_sbrk scalb
_seekdir sema_destroy
_sema_held semctl
_semget semids
_semop semtimedop
_setcontext setgdi
_setgroups setitimer
_setjmp setkey
_setpgid setpgid
_setrlimit setsid
_settaskid setuid
_shmat shmctl
_shmdt shmget
_shmids sibuf
_sigaction sigaddset
_sigaltstack sigdelset
_sigemptyset sigfillset
_sighold sigignore
_siginame 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_clldlist
_sys_fpelist     _sys_illlist
_sys_seglst      _sys_siginfolistp
_sys_siglist     _sys_siglistn
_sys_siglnstp    _sys_traplist
_syscall         _sysconf
_sysinfo         _syslog
_tcdrain         _tcflow
_tcfush          _tcgetattr
_tcgetpgrp       _tcgetsid
_tcsendbreak     _tcsetattr
_tcsetpgrp       _tdelete
_tell            _telldir
_tempnam         _tfind
_time            _times
_timezone        _toascii
_tolower         _toupper
_tsearch         _ttyname
_twalk           _tzname
_tzset
_umask
_umount
_umount2
_unlink
_utime
_waitid
_waitpid
_wraacct
_writev
abort
access
acl
addsever
alarm
altzone
asctime
asctime_r
atexit
atoi
atoll
atomic_add_16
atomic_add_16_nv
atomic_add_32
atomic_add_32_nv
atomic_add_4
atomic_add_4_nv
atomic_add_long
atomic_add_long_nv
atomic_and_32
atomic_or_32
attropen
basename
bcmp
bcopy
bindtextdomain
bind_textdomain_codeset
brk
bsd_signal
bsearch
btowc
bzero calloc
catclose catgets
catopen cfgetispeed
cfgetospeed cfsetispeed
cfgetospeed cftime
chdir chmod
chown chroot
clearerr clock
close closedir
closefrom closelog
cond_broadcast cond_destroy
cond_init cond_reltimedwait
cond_signal cond_timedwait
cond_wait confstr
creat crypt
crypt_genhash_impl crypt_gensalt
crypt_gensalt_impl csetcol
csetlen ctermid
ctermid_r ctime
cftime_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 dcgettext
<table>
<thead>
<tr>
<th>Function</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>difftime</td>
<td>directio</td>
</tr>
<tr>
<td>dirname</td>
<td>div</td>
</tr>
<tr>
<td>dladdr</td>
<td>dladdr1</td>
</tr>
<tr>
<td>dlclose</td>
<td>dldump</td>
</tr>
<tr>
<td>dlerror</td>
<td>dlinfo</td>
</tr>
<tr>
<td>dlmopen</td>
<td>dlopen</td>
</tr>
<tr>
<td>dlSYM</td>
<td>dngettext</td>
</tr>
<tr>
<td>double_to_decimal</td>
<td>drand48</td>
</tr>
<tr>
<td>dup</td>
<td>dup2</td>
</tr>
<tr>
<td>econvert</td>
<td>ecvt</td>
</tr>
<tr>
<td>encrypt</td>
<td>endgrent</td>
</tr>
<tr>
<td>endnetgrent</td>
<td>endpwent</td>
</tr>
<tr>
<td>endspent</td>
<td>endusershell</td>
</tr>
<tr>
<td>endutent</td>
<td>endutxent</td>
</tr>
<tr>
<td>environ</td>
<td>erand48</td>
</tr>
<tr>
<td>errno</td>
<td>euccol</td>
</tr>
<tr>
<td>euclen</td>
<td>eucscol</td>
</tr>
<tr>
<td>exec</td>
<td>execle</td>
</tr>
<tr>
<td>execcl</td>
<td>execv</td>
</tr>
<tr>
<td>execvp</td>
<td>execvp</td>
</tr>
<tr>
<td>exit</td>
<td>extended_to_decimal</td>
</tr>
<tr>
<td>facl</td>
<td>fattach</td>
</tr>
<tr>
<td>fchdir</td>
<td>fchmod</td>
</tr>
<tr>
<td>fchown</td>
<td>fchownat</td>
</tr>
<tr>
<td>fchroot</td>
<td>fclose</td>
</tr>
<tr>
<td>fcntl</td>
<td>fconvert</td>
</tr>
<tr>
<td>fcvt</td>
<td>fdetach</td>
</tr>
<tr>
<td>fdopen</td>
<td>fdopendir</td>
</tr>
<tr>
<td>fdwalk</td>
<td>feof</td>
</tr>
<tr>
<td>Function</td>
<td>Function</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>ferror</td>
<td>fflush</td>
</tr>
<tr>
<td>ffs</td>
<td>fgetc</td>
</tr>
<tr>
<td>fgetgrent</td>
<td>fgetgrent_r</td>
</tr>
<tr>
<td>fgetpos</td>
<td>fgetpwent</td>
</tr>
<tr>
<td>fgetpwent_r</td>
<td>fgets</td>
</tr>
<tr>
<td>fgetspent</td>
<td>fgetspent_r</td>
</tr>
<tr>
<td>fgetwc</td>
<td>fgetws</td>
</tr>
<tr>
<td>file_to_decimal</td>
<td>fileno</td>
</tr>
<tr>
<td>finite</td>
<td>flockfile</td>
</tr>
<tr>
<td>fmtmsg</td>
<td>fnmatch</td>
</tr>
<tr>
<td>fopen</td>
<td>fork</td>
</tr>
<tr>
<td>fork1</td>
<td>forkall</td>
</tr>
<tr>
<td>fpathconf</td>
<td>fpclass</td>
</tr>
<tr>
<td>fggettext</td>
<td>fpgetround</td>
</tr>
<tr>
<td>fggettextsticky</td>
<td>fprintf</td>
</tr>
<tr>
<td>fpsetmask</td>
<td>fpsetround</td>
</tr>
<tr>
<td>fpsetsticky</td>
<td>fputc</td>
</tr>
<tr>
<td>fputs</td>
<td>fputwc</td>
</tr>
<tr>
<td>fputws</td>
<td>fread</td>
</tr>
<tr>
<td>free</td>
<td>freopen</td>
</tr>
<tr>
<td>frexp</td>
<td>fscanf</td>
</tr>
<tr>
<td>fseek</td>
<td>fseeko</td>
</tr>
<tr>
<td>fsetpos</td>
<td>fstat</td>
</tr>
<tr>
<td>fstatat</td>
<td>fstatfs</td>
</tr>
<tr>
<td>fstatvfs</td>
<td>fsync</td>
</tr>
<tr>
<td>ftell</td>
<td>ftello</td>
</tr>
<tr>
<td>ftime</td>
<td>ftok</td>
</tr>
<tr>
<td>ftruncate</td>
<td>ftrylockfile</td>
</tr>
<tr>
<td>futimens</td>
<td></td>
</tr>
</tbody>
</table>
ftw func_to_decimal
funlockfile futimesat
fwide fwprintf
fwrite fwscanf
gconvert gcvt
getacct getc
getc_unlocked getchar
getchar_unlocked getcontext
cgetpuid getcwd
cdate getdate
getdate getdate_err
getdents getdtablesize
getegid getenv
geteuid getexecname
ggettextmminentent getgid
getgrent getgrent_r
getgrgid getgrgid_r
getgrnam getgrnam_r
getgroups gethomelogroup
gethostid gethostname
gethrtime gethrvtime
getisax getitim
getloadavg getlogin
getlogin_r getmntany
getmntent getmsg
getnetgrent getnetgrent_r
getopt getopt_clip
getopt_long getopt_long_only
getpagesize getpagesizes
getpass getpassphrase
getpeerucred  getpflags
getpgid     getpgrp
getpid      getpmsg
getppid     getppriv
getpriority getprojid
getpw       getpwent
getpwent_r  getpwent
getpwnam_r  getpwnam
getpwuid    getrcntl
getrcntl_r  getrusage
getrlimit   getsid
gets        getsid
getspent    getspent_r
getspnam    getspnam_r
getsubopt   gettaskid
gettext     gettimeofday
gettext    getutid
getusershell getustack
getutent    getutid
getutent    getutmp
getutmpx    getutxent
getutxid    getutxline
getutxline  getvfsent
getvfsany   getvfspec
getvfsfile  getwc
getw        getwd
getwchar     getws
getzoneid   getzoneidbyname
getzonenamebyid  glob
globfree   gmtime
<table>
<thead>
<tr>
<th>Function</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>gmtime_r</td>
<td>grantpt</td>
</tr>
<tr>
<td>gsignal</td>
<td>hasmntopt</td>
</tr>
<tr>
<td>hcreate</td>
<td>hdestroy</td>
</tr>
<tr>
<td>hsearch</td>
<td>iconv</td>
</tr>
<tr>
<td>iconv_close</td>
<td>iconv_open</td>
</tr>
<tr>
<td>imaxabs</td>
<td>imaxdiv</td>
</tr>
<tr>
<td>index</td>
<td>initgroups</td>
</tr>
<tr>
<td>initstate</td>
<td>innetgr</td>
</tr>
<tr>
<td>insque</td>
<td>ioctl</td>
</tr>
<tr>
<td>isaexec</td>
<td>isalnum</td>
</tr>
<tr>
<td>isalpha</td>
<td>isasci</td>
</tr>
<tr>
<td>isastream</td>
<td>isatty</td>
</tr>
<tr>
<td>isblank</td>
<td>iscntrl</td>
</tr>
<tr>
<td>isdigit</td>
<td>isenglish</td>
</tr>
<tr>
<td>isgraph</td>
<td>isideogram</td>
</tr>
<tr>
<td>islower</td>
<td>isnan</td>
</tr>
<tr>
<td>isnand</td>
<td>isnanf</td>
</tr>
<tr>
<td>isnumber</td>
<td>isphonogram</td>
</tr>
<tr>
<td>isprint</td>
<td>ispunct</td>
</tr>
<tr>
<td>issetugid</td>
<td>isspace</td>
</tr>
<tr>
<td>isspecial</td>
<td>isupper</td>
</tr>
<tr>
<td>iswalnum</td>
<td>iswalpha</td>
</tr>
<tr>
<td>iswblank</td>
<td>iswcntrl</td>
</tr>
<tr>
<td>iswctype</td>
<td>iswdigit</td>
</tr>
<tr>
<td>iswgraph</td>
<td>iswlower</td>
</tr>
<tr>
<td>iswprint</td>
<td>iswpunct</td>
</tr>
<tr>
<td>iswspace</td>
<td>iswupper</td>
</tr>
<tr>
<td>iswxdigit</td>
<td>isxdigit</td>
</tr>
<tr>
<td>jrand48</td>
<td>kill</td>
</tr>
</tbody>
</table>
killpg  l64a
labs   ladd
lchown lckpwd
lcong48 ldexp
ldivide lexp10
lfind  lfmt
link   llabs
ldiv   llog10
llseek lltostr
localeconv localtime
localtime_r lockf
logb   lone
longjmp lrand48
lsearch lseek
lshiftl lstat
lsub   lten
lzero  madvise
makecontext makeutx
malloc mblen
mbrlen mbtowc
mbinit mbsrtowcs
mbstowcs mbtowc
memalign memccpy
memchr memcmp
memcntl memcpy
meminfo memmove
memset mincore
mkdir mkfifo
mknod mkstemp
<table>
<thead>
<tr>
<th>libc(3LIB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>mktemp</td>
</tr>
<tr>
<td>mlock</td>
</tr>
<tr>
<td>mmap</td>
</tr>
<tr>
<td>modf</td>
</tr>
<tr>
<td>modutx</td>
</tr>
<tr>
<td>mount</td>
</tr>
<tr>
<td>mrand48</td>
</tr>
<tr>
<td>msgget</td>
</tr>
<tr>
<td>msgrcv</td>
</tr>
<tr>
<td>msgsnd</td>
</tr>
<tr>
<td>munlock</td>
</tr>
<tr>
<td>munmap</td>
</tr>
<tr>
<td>mutex_init</td>
</tr>
<tr>
<td>mutex_trylock</td>
</tr>
<tr>
<td>nextafter</td>
</tr>
<tr>
<td>nftw</td>
</tr>
<tr>
<td>nice</td>
</tr>
<tr>
<td>nrand48</td>
</tr>
<tr>
<td>nss_delete</td>
</tr>
<tr>
<td>nss_getent</td>
</tr>
<tr>
<td>nss_setent</td>
</tr>
<tr>
<td>ntp_gettime</td>
</tr>
<tr>
<td>openat</td>
</tr>
<tr>
<td>openlog</td>
</tr>
<tr>
<td>opterr</td>
</tr>
<tr>
<td>optopt</td>
</tr>
<tr>
<td>pathconf</td>
</tr>
<tr>
<td>pclose</td>
</tr>
<tr>
<td>perror</td>
</tr>
</tbody>
</table>
pipe
poll
port_alert
port_create
port_get
port_send
posix_openpt
posix_spawn
posix_spawn_file_actions_addclose
posix_spawn_file_actions_addopen
posix_spawn_file_actions_init
posix_spawnattr_getflags
posix_spawnattr_getgrent
posix_spawnattr_getgrentdefault
posix_spawnattr_init
posix_spawnattr_setgrent
posix_spawnattr_setschedparam
posix_spawnattr_setschedpolicy
posix spawns
pread
printf
priocntl
priocntlsel
priv allocset
priv delset
priv fillset
priv getbyname
priv getsetbyname
priv gettext
priv intersect
priv isemptyset
priv isfullset
priv_isubset priv_set
priv_set_to_str priv_str_to_set
priv_union processor_bind
processor_info profi
pselect 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_getstack
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_setstack
pthread_attr_setstackaddr
pthread_attr_setstacksize
pthread_barrier_destroy
pthread_barrier_init
pthread_barrier_wait
pthread_barrierattr_destroy
pthread_barrierattr_getpshared
pthread_barrierattr_init
pthread_barrierattr_setpshared
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_getclock
<table>
<thead>
<tr>
<th>Function</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>pthread_condattr_getpshared</td>
<td>pthread_condattr_init</td>
</tr>
<tr>
<td>pthread_condattr_setclock</td>
<td>pthread_condattr_setpshared</td>
</tr>
<tr>
<td>pthread_create</td>
<td>pthread_detach</td>
</tr>
<tr>
<td>pthread_equal</td>
<td>pthread_exit</td>
</tr>
<tr>
<td>pthread_getconcurrency</td>
<td>pthread_getschedparam</td>
</tr>
<tr>
<td>pthread_getspecific</td>
<td>pthread_join</td>
</tr>
<tr>
<td>pthread_key_create</td>
<td>pthread_key_delete</td>
</tr>
<tr>
<td>pthread_kill</td>
<td>pthread_mutex_consistent_np</td>
</tr>
<tr>
<td>pthread_mutex_destroy</td>
<td>pthread_mutex_getprioceiling</td>
</tr>
<tr>
<td>pthread_mutex_init</td>
<td>pthread_mutex_lock</td>
</tr>
<tr>
<td>pthread_mutex_reltimedlock_np</td>
<td>pthread_mutex_setprioceiling</td>
</tr>
<tr>
<td>pthread_mutex_timedlock</td>
<td>pthread_mutex_trylock</td>
</tr>
<tr>
<td>pthread_mutex_unlock</td>
<td>pthread_mutexattr_destroy</td>
</tr>
<tr>
<td>pthread_mutexattr_getprioceiling</td>
<td>pthread_mutexattr_getprotocol</td>
</tr>
<tr>
<td>pthread_mutexattr_getpshared</td>
<td>pthread_mutexattr_getrobust_np</td>
</tr>
<tr>
<td>pthread_mutexattr_gettype</td>
<td>pthread_mutexattr_init</td>
</tr>
<tr>
<td>pthread_mutexattr_setprioceiling</td>
<td>pthread_mutexattr_setprotocol</td>
</tr>
<tr>
<td>pthread_mutexattr_setpshared</td>
<td>pthread_mutexattr_setrobust_np</td>
</tr>
<tr>
<td>pthread_mutexattr_settype</td>
<td>pthread_once</td>
</tr>
<tr>
<td>pthread_rwlock_destroy</td>
<td>pthread_rwlock_init</td>
</tr>
<tr>
<td>pthread_rwlock_rdlock</td>
<td>pthread_rwlock_reltimedrdlock_np</td>
</tr>
<tr>
<td>pthread_rwlock_rwlock_np</td>
<td>pthread_rwlock_timedrdlock</td>
</tr>
<tr>
<td>pthread_rwlock_timedrdlock</td>
<td>pthread_rwlock_tryrdlock</td>
</tr>
<tr>
<td>pthread_rwlock_trywrlock</td>
<td>pthread_rwlock_unlock</td>
</tr>
<tr>
<td>pthread_rwlock_wrlcok</td>
<td>pthread_rwlockattr_destroy</td>
</tr>
<tr>
<td>pthread_rwlockattr_getpshared</td>
<td>pthread_rwlockattr_init</td>
</tr>
<tr>
<td>pthread_rwlockattr_setpshared</td>
<td>pthread_rwlockattr_setpshared</td>
</tr>
<tr>
<td>pthread_setcancelstate</td>
<td>pthread_setcanceltype</td>
</tr>
<tr>
<td>pthread_setconcurrency</td>
<td>pthread_setspecific</td>
</tr>
</tbody>
</table>

**libc(3LIB)**
pthread_sigmask  pthread_setschedparam
pthread_setschedprio  pthread_spin_destroy
pthread_spin_init  pthread_spin_lock
pthread_spin_trylock  pthread_spin_unlock
pthread_testcancel  ptsname
putacct  putc
putc_unlocked  putchar
putchar_unlocked  putenv
putmsg  putpmsg
putw  puts
putspent  pututline
putuxline  putw
putwc  putwchar
putws  pwrite
qconvert  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_recipient_pid
rctlblk_set_value  rctlblk_size
re_comp  re_exec
<table>
<thead>
<tr>
<th>Function</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>read</td>
<td>readdir</td>
</tr>
<tr>
<td>readdir_r</td>
<td>readlink</td>
</tr>
<tr>
<td>readv</td>
<td>realloc</td>
</tr>
<tr>
<td>realpath</td>
<td>reboot</td>
</tr>
<tr>
<td>regcmp</td>
<td>regcomp</td>
</tr>
<tr>
<td>regerror</td>
<td>regex</td>
</tr>
<tr>
<td>regexec</td>
<td>regfree</td>
</tr>
<tr>
<td>remove</td>
<td>remque</td>
</tr>
<tr>
<td>rename</td>
<td>renameat</td>
</tr>
<tr>
<td>resetmnttab</td>
<td>resolvepath</td>
</tr>
<tr>
<td>rewind</td>
<td>rewinddir</td>
</tr>
<tr>
<td>rindex</td>
<td>rmdir</td>
</tr>
<tr>
<td>rw_rdlock</td>
<td>rw_read_held</td>
</tr>
<tr>
<td>rw_tryrdlock</td>
<td>rw_trywrlock</td>
</tr>
<tr>
<td>rw_unlock</td>
<td>rw_write_held</td>
</tr>
<tr>
<td>rw_wrlock</td>
<td>rwlock_destroy</td>
</tr>
<tr>
<td>rwlock_init</td>
<td>sbrk</td>
</tr>
<tr>
<td>scalb</td>
<td>scandir</td>
</tr>
<tr>
<td>scanf</td>
<td>schedctl_exit</td>
</tr>
<tr>
<td>schedctl_init</td>
<td>schedctl_lookup</td>
</tr>
<tr>
<td>schedctl_start</td>
<td>schedctl_stop</td>
</tr>
<tr>
<td>seconvert</td>
<td>seed48</td>
</tr>
<tr>
<td>seekdir</td>
<td>select</td>
</tr>
<tr>
<td>sema_destroy</td>
<td>sema_held</td>
</tr>
<tr>
<td>sema_init</td>
<td>sema_post</td>
</tr>
<tr>
<td>sema_trywait</td>
<td>sema_wait</td>
</tr>
<tr>
<td>semctl</td>
<td>semget</td>
</tr>
<tr>
<td>semids</td>
<td>semop</td>
</tr>
<tr>
<td>semtimedop</td>
<td>setbuf</td>
</tr>
</tbody>
</table>
siglongjmp  signal
sigpause  sigpending
sigprocmask  sigreelse
sigsend  sigsendset
sigset  sigsetjmp
sigstack  sigsuspend
sigwait  single_to_decimal
sleep  snprintf
sprintf  srand
srand48  srandom
sscanf  ssignal
stack_getbounds  stack_inbounds
stack_setbounds  stackViolation
stat  statfs
statvfs  stime
str2sig  strcasecmp
strcat  strchr
strcoll  strcmp
strcpn  strcspn
strdup  strerror
strerror_r  strfmon
strftime  string_to_decimal
strlcat  strlcpy
strlcpy  strlcmp
strlcpy  strncasecmp
strncat  strncmp
strncpy  strpbrk
strptime  strrrchr
strstr  strspn
strndir  strtod
<table>
<thead>
<tr>
<th>Function</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>strtof</td>
<td>strtoimax</td>
</tr>
<tr>
<td>strtok</td>
<td>strtok_r</td>
</tr>
<tr>
<td>strtol</td>
<td>strtold</td>
</tr>
<tr>
<td>strtoll</td>
<td>strtoul</td>
</tr>
<tr>
<td>strtoull</td>
<td>strtooumax</td>
</tr>
<tr>
<td>strtowcs</td>
<td>strxfrm</td>
</tr>
<tr>
<td>swab</td>
<td>swapcontext</td>
</tr>
<tr>
<td>swapctl</td>
<td>swprintf</td>
</tr>
<tr>
<td>swscanf</td>
<td>symlink</td>
</tr>
<tr>
<td>sync</td>
<td>sync_instruction_memory</td>
</tr>
<tr>
<td>sysconf</td>
<td>sysfs</td>
</tr>
<tr>
<td>sysinfo</td>
<td>syslog</td>
</tr>
<tr>
<td>system</td>
<td>tcdrain</td>
</tr>
<tr>
<td>tcflow</td>
<td>tcflush</td>
</tr>
<tr>
<td>tcgetattr</td>
<td>tcgetpgrp</td>
</tr>
<tr>
<td>tcgetattr</td>
<td>tcgetpgrp</td>
</tr>
<tr>
<td>tcgetsid</td>
<td>tcsetpgrp</td>
</tr>
<tr>
<td>tcdelete</td>
<td>tell</td>
</tr>
<tr>
<td>telldir</td>
<td>tempnam</td>
</tr>
<tr>
<td>textdomain</td>
<td>tfind</td>
</tr>
<tr>
<td>thr_continue</td>
<td>thr_create</td>
</tr>
<tr>
<td>thr_exit</td>
<td>thr_getconcurrency</td>
</tr>
<tr>
<td>thr_getprio</td>
<td>thr_getspecific</td>
</tr>
<tr>
<td>thr_join</td>
<td>thr_keycreate</td>
</tr>
<tr>
<td>thr_kill</td>
<td>thr_main</td>
</tr>
<tr>
<td>thr_min_stack</td>
<td>thr_self</td>
</tr>
<tr>
<td>thr_setconcurrency</td>
<td>thr_setconcurrency</td>
</tr>
<tr>
<td>thr_setspecific</td>
<td>thr_setprio</td>
</tr>
<tr>
<td>thr_stksegment</td>
<td>thr_suspend</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
</tr>
<tr>
<td>thr_yield</td>
<td>time</td>
</tr>
<tr>
<td>times</td>
<td>timezone</td>
</tr>
<tr>
<td>tmpfile</td>
<td>tmpnam</td>
</tr>
<tr>
<td>tmpnam_r</td>
<td>toascii</td>
</tr>
<tr>
<td>tolower</td>
<td>toupper</td>
</tr>
<tr>
<td>towctrans</td>
<td>tolower</td>
</tr>
<tr>
<td>towupper</td>
<td>truncate</td>
</tr>
<tr>
<td>tsearch</td>
<td>ttiname</td>
</tr>
<tr>
<td>ttiname_r</td>
<td>ttyslot</td>
</tr>
<tr>
<td>twalk</td>
<td>tzname</td>
</tr>
<tr>
<td>tzset</td>
<td>uadmin</td>
</tr>
<tr>
<td>ualarm</td>
<td>ucred_free</td>
</tr>
<tr>
<td>ucred_get</td>
<td>ucred_getegid</td>
</tr>
<tr>
<td>ucred_geteuid</td>
<td>ucred_getgroups</td>
</tr>
<tr>
<td>ucred_getpflags</td>
<td>ucred_getpid</td>
</tr>
<tr>
<td>ucred_getprivset</td>
<td>ucred_getprojid</td>
</tr>
<tr>
<td>ucred_getgid</td>
<td>ucred_getruid</td>
</tr>
<tr>
<td>ucred_getsgid</td>
<td>ucred_getsetsid</td>
</tr>
<tr>
<td>ucred_getzoneid</td>
<td>ucred_size</td>
</tr>
<tr>
<td>ulckpwdf</td>
<td>ulimit</td>
</tr>
<tr>
<td>ulltostr</td>
<td>umask</td>
</tr>
<tr>
<td>umount</td>
<td>umount2</td>
</tr>
<tr>
<td>uname</td>
<td>ungetc</td>
</tr>
<tr>
<td>ungetwc</td>
<td>unlink</td>
</tr>
<tr>
<td>unlinkat</td>
<td>unlockpt</td>
</tr>
<tr>
<td>unordered</td>
<td>unsetenv</td>
</tr>
<tr>
<td>updwtmp</td>
<td>updwtmpx</td>
</tr>
<tr>
<td>usleep</td>
<td>ustat</td>
</tr>
<tr>
<td>utime</td>
<td>utimes</td>
</tr>
<tr>
<td>Function 1</td>
<td>Function 2</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>utmpname</td>
<td>utmpxname</td>
</tr>
<tr>
<td>valloc</td>
<td>vfork</td>
</tr>
<tr>
<td>vfprintf</td>
<td>vfscanf</td>
</tr>
<tr>
<td>vfwpprintf</td>
<td>vfswscanf</td>
</tr>
<tr>
<td>vhangup</td>
<td>vlfmt</td>
</tr>
<tr>
<td>vpfmt</td>
<td>vprintf</td>
</tr>
<tr>
<td>vscanf</td>
<td>vsnprintf</td>
</tr>
<tr>
<td>vsprintf</td>
<td>vsscanf</td>
</tr>
<tr>
<td>vwprintf</td>
<td>vwscanf</td>
</tr>
<tr>
<td>vwscanff</td>
<td>vwscanff</td>
</tr>
<tr>
<td>vwprintf</td>
<td>vwscanf</td>
</tr>
<tr>
<td>tantal</td>
<td>wait</td>
</tr>
<tr>
<td>wait3</td>
<td>wait4</td>
</tr>
<tr>
<td>waitid</td>
<td>waitpid</td>
</tr>
<tr>
<td>walkcontext</td>
<td>watoll</td>
</tr>
<tr>
<td>wcrtomb</td>
<td>wcscat</td>
</tr>
<tr>
<td>wcschr</td>
<td>wcscmp</td>
</tr>
<tr>
<td>wcschr</td>
<td>wcscmp</td>
</tr>
<tr>
<td>wcscoll</td>
<td>wcscpy</td>
</tr>
<tr>
<td>wcscspn</td>
<td>wcsftime</td>
</tr>
<tr>
<td>wcslen</td>
<td>wcsncat</td>
</tr>
<tr>
<td>wcscncmp</td>
<td>wcscncpy</td>
</tr>
<tr>
<td>wcspbrk</td>
<td>wcsrcrchr</td>
</tr>
<tr>
<td>wcstombs</td>
<td>wcsspn</td>
</tr>
<tr>
<td>wcstr</td>
<td>wcstod</td>
</tr>
<tr>
<td>wcstof</td>
<td>wcstoimax</td>
</tr>
<tr>
<td>wcstok</td>
<td>wcstol</td>
</tr>
<tr>
<td>wcstold</td>
<td>wcstoll</td>
</tr>
<tr>
<td>wcstombs</td>
<td>wcstoul</td>
</tr>
<tr>
<td>wcstoull</td>
<td>wcstoumax</td>
</tr>
<tr>
<td>wcswidth</td>
<td>wcswcs</td>
</tr>
<tr>
<td>Function</td>
<td>Function</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>wcsxfrm</td>
<td>wctob</td>
</tr>
<tr>
<td>wctomb</td>
<td>wctrans</td>
</tr>
<tr>
<td>wctype</td>
<td>wcwidth</td>
</tr>
<tr>
<td>wmemchr</td>
<td>wmencmp</td>
</tr>
<tr>
<td>wmemcpy</td>
<td>wmemmove</td>
</tr>
<tr>
<td>wmemset</td>
<td>wordexp</td>
</tr>
<tr>
<td>wordfree</td>
<td>wprintf</td>
</tr>
<tr>
<td>wracct</td>
<td>write</td>
</tr>
<tr>
<td>writev</td>
<td>wscanf</td>
</tr>
<tr>
<td>wscasecmp</td>
<td>wscat</td>
</tr>
<tr>
<td>wschr</td>
<td>wscmp</td>
</tr>
<tr>
<td>wscoll</td>
<td>wscoll</td>
</tr>
<tr>
<td>wscpy</td>
<td>wscspn</td>
</tr>
<tr>
<td>wsdup</td>
<td>wslen</td>
</tr>
<tr>
<td>wsnrcasecmp</td>
<td>wncat</td>
</tr>
<tr>
<td>wncmp</td>
<td>wncpy</td>
</tr>
<tr>
<td>wspbrk</td>
<td>wprintf</td>
</tr>
<tr>
<td>wsrchr</td>
<td>wsscanf</td>
</tr>
<tr>
<td>wsspn</td>
<td>wstod</td>
</tr>
<tr>
<td>wstok</td>
<td>wstol</td>
</tr>
<tr>
<td>wstoll</td>
<td>wstosstr</td>
</tr>
<tr>
<td>wxxfm</td>
<td>yield</td>
</tr>
</tbody>
</table>

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
<table>
<thead>
<tr>
<th>Function</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>_ftruncate64</td>
<td>_ftw64</td>
</tr>
<tr>
<td>_getdents64</td>
<td>_getrlimit64</td>
</tr>
<tr>
<td>_lastbuf</td>
<td>_lockf64</td>
</tr>
<tr>
<td>_lseek64</td>
<td>_lstat64</td>
</tr>
<tr>
<td>_mkstemp64</td>
<td>_mmap64</td>
</tr>
<tr>
<td>_nftw64</td>
<td>_open64</td>
</tr>
<tr>
<td>_pread64</td>
<td>_pwrite64</td>
</tr>
<tr>
<td>_readdir64</td>
<td>_readdir64_r</td>
</tr>
<tr>
<td>_s_fcntl</td>
<td>_setrlimit64</td>
</tr>
<tr>
<td>_stat64</td>
<td>_statvfs64</td>
</tr>
<tr>
<td>_sys_nsig</td>
<td>_tell64</td>
</tr>
<tr>
<td>_truncate64</td>
<td>_xftw64</td>
</tr>
<tr>
<td>creat64</td>
<td>fgetpos64</td>
</tr>
<tr>
<td>fopen64</td>
<td>freopen64</td>
</tr>
<tr>
<td>fseeko64</td>
<td>fsopen64</td>
</tr>
<tr>
<td>fstat64</td>
<td>fsetpos64</td>
</tr>
<tr>
<td>fstatvfs64</td>
<td>fstatvfs64</td>
</tr>
<tr>
<td>ftello64</td>
<td>ftruncate64</td>
</tr>
<tr>
<td>ftw64</td>
<td>getdents64</td>
</tr>
<tr>
<td>getrlimit64</td>
<td>lockf64</td>
</tr>
<tr>
<td>lseek64</td>
<td>lstat64</td>
</tr>
<tr>
<td>mkstemp64</td>
<td>mmap64</td>
</tr>
<tr>
<td>nftw64</td>
<td>open64</td>
</tr>
<tr>
<td>pread64</td>
<td>ptrace</td>
</tr>
<tr>
<td>pwrite64</td>
<td>readdir64</td>
</tr>
<tr>
<td>readdir64_r</td>
<td>s_fcntl</td>
</tr>
<tr>
<td>s_ioctl</td>
<td>select_large_fdset</td>
</tr>
<tr>
<td>setrlimit64</td>
<td>stat64</td>
</tr>
<tr>
<td>statvfs64</td>
<td>sys_errlist</td>
</tr>
<tr>
<td>sys_nerr</td>
<td>tell64</td>
</tr>
</tbody>
</table>
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   _Q_flt
._Q_fne  _Q_itoq
._Q lltoq _Q_mul
._Q_neg   _Q_qtd
._Q_qtoi  _Q_qtod
._Q_qtos  _Q_qtoq
._Q qtoull _Q_sqrt
._Q stoq  _Q_sub
._Q ulltoq _Q_utoq
._dtoll  __dtob
._dttoll __f32toll
.__ftou __ftoull
.__umul64
```

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

```
__fpstart  __fp_hw
```

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

- _Qp_set
- _Qp_cmpe
- _Qp_dtoq
- _Qp_fge
- _Qp_fle
- _Qp_fne
- _Qp_mul
- _Qp_qtod
- _Qp_qtoq
- _Qp_qtouq
- _Qp_sqrt
- _Qp_sub
- _Qp_uptoq
- __align_cpy_1
- __align_cpy_16
- __align_cpy_2
- __align_cpy_4
- __align_cpy_8
- __dtoul
- __foul
- __sparc_utrap_install

**Files**
- /lib/libc.so.1 shared object
- /lib/64/libc.so.1 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td>ATTRIBUTE TYPE</td>
<td>ATTRIBUTE VALUE</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also  `pvs(1), Intro(2), Intro(3), attributes(5), lf64(5), standards(5)`
The \texttt{libc\_db} library provides support for monitoring and manipulating threads-related aspects of a multithreaded program. There are at least two processes involved, the controlling process and one or more target processes. The controlling process is the \texttt{libc\_db} client, which links with \texttt{libc\_db} and uses \texttt{libc\_db} to inspect or modify threads-related aspects of one or more target processes. The target processes must be multithreaded processes that use \texttt{libc}. The controlling process might or might not be multithreaded itself.

The most commonly anticipated use for \texttt{libc\_db} is that the controlling process will be a debugger for a multithreaded program, hence the "db" in \texttt{libc\_db}.

The \texttt{libc\_db} library is dependent on the internal implementation details of \texttt{libc}. It is a "friend" of \texttt{libc} in the C++ sense, which is precisely the "value added" by \texttt{libc\_db}. It encapsulates the knowledge of \texttt{libc} internals that a debugger needs to manipulate the threads-related state of a target process.

To be able to inspect and manipulate target processes, \texttt{libc\_db} makes use of certain process control primitives that must be provided by the process using \texttt{libc\_db}. The imported interfaces are defined in \texttt{proc\_service(3PROC)}. In other words, the controlling process is linked with \texttt{libc\_db} and calls routines in \texttt{libc\_db}. In turn, \texttt{libc\_db} calls certain routines that it expects the controlling process to provide. These process control primitives allow \texttt{libc\_db} to:

- Look up symbols in a target process.
- Stop and continue individual lightweight processes (LWPs) within a target process.
- Stop and continue an entire target process.
- Read and write memory and registers in a target process.

Initially, a controlling process obtains a handle for a target process. Through that handle it can then obtain handles for the component objects of the target process, its threads, its synchronization objects, and its thread-specific-data keys.

When \texttt{libc\_db} needs to return sets of handles to the controlling process, for example, when returning handles for all the threads in a target process, it uses an iterator function. An iterator function calls back a client-specified function once for each handle to be returned, passing one handle back on each call to the callback function. The calling function also passes another parameter to the iterator function, which the iterator function passes on to the callback function. This makes it easy to build a linked list of thread handles for a particular target process. The additional parameter is the head of the linked list, and the callback function simply inserts the current handle into the linked list.
Callback functions are expected to return an integer. Iteration terminates early if a callback function returns a non-zero value. Otherwise, iteration terminates when there are no more handles to pass back.

**Interfaces** The shared object `libc_db.so.1` provides the public interfaces defined below. See `Intro(3)` for additional information on shared object interfaces.

```plaintext
td_init   td_log
td_sync_get_info   td_sync_get_stats
td_sync_setstate   td_sync_waiters
td_ta_clear_event   td_ta_delete
td_ta_enable_stats   td_ta_event_addr
td_ta_event_getmsg   td_ta_get_nthreads
td_ta_get_ph   td_ta_get_stats
td_ta_map_addr2sync   td_ta_map_id2thr
td_ta_map_lwp2thr   td_ta_new
td_ta_reset_stats   td_ta_set_event
td_ta_setconcurrency   td_ta_sync_iter
td_ta_sync_tracking_enable   td_ta_thr_iter
td_ta_tsd_iter   td_thr_clear_event
td_thr_dbresume   td_thr_dbsuspend
td_thr_event_enable   td_thr_event_getmsg
td_thr_get_info   td_thr_getfpregs
td_thr_getregs   td_thr_getxregs
td_thr_getxregsizes   td_thr_lockowner
td_thr_set_event   td_thr_setfpregs
td_thr_setgrends   td_thr_setprio
td_thr_setsigpending   td_thr_setxregs
td_thr_sigsetmask   td_thr_sleepinfo
td_thr_tlsbase   td_thr_tsd
td_thr_validate
```
Files
/lib/libc_db.so.1 shared object
/lib/64/libc_db.so.1 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also
Intro(3), td_ta_new(3C_DB), attributes(5), threads(5)
Name  libcfgadm – configuration administration library

Synopsis  cc { flag... } file... -lcfgadm -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.

    config_ap_id_cmp  config_change_state
    config_help  config_list
    config_list_ext  config_private_func
    config_stat  config_strerror
    config_test  config_unload_libs

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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32–bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64–bit)</td>
</tr>
<tr>
<td>MT–Level</td>
<td>Mt–Safe</td>
</tr>
</tbody>
</table>

See Also  pvs(1), cfgadm(1M), Intro(3), config_admin(3CFGADM), attributes(5)
libcommputil – communication protocol parser utilities library

Synopsis

cc [ flag...] file... -lcommputil [ library... ]
#include <sdp.h>

Description

The communication protocol parser utilities library is a placeholder for public interfaces that facilitate parsing of various communication protocols. Functions in this library parse the SDP (Session Description Protocol) description, check for syntax conformance, and generate SDP descriptions.

SDP (Session Description Protocol), described in RFC 4566, describes multimedia sessions for the purposes of session announcement, session invitation, and other forms of multimedia session initiation. SDP is used to convey session information in Session Initiation Protocol (SIP), Streaming Media (Real Time Streaming Protocol, RTSP), email, and World Wide Web and Multicast Session Announcement.

Interfaces

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

```
sdp_add_attribute          sdp_add_bandwidth
sdp_add_connection         sdp_add_email
sdp_add_information        sdp_add_key
sdp_add_media              sdp_add_name
sdp_add_origin             sdp_add_phone
sdp_add_repeat             sdp_add_time
sdp_add_uri                sdp_add_zone
sdp_clone_session          sdp_delete_attribute
sdp_delete_field           sdp_delete_media
sdp_delete_media_field     sdp_find_attribute
sdp_find_media             sdp_find_media_rtpmap
sdp_free_session           sdp_new_session
sdp_parse                  sdp_session_to_str
```

Files

/lib/libcommputil.so.1 shared object.
/lib/64/libcommputil.so.1 64-bit shared object.
Attributes  See attributes(5) for description of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcs (32–bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64–bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Committed</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also  Intro(3), attributes(5)
## Name
libcontract – contract management library

## Synopsis
cc [ flag... ] 'getconf LFS_CFLAGS' file... -lcontract [ library... ]
#include <libcontract.h>

## Description
Functions in this library provide various interfaces to interact with the contract(4) file system. The header provides structure and function declarations for all library interfaces.

## Interfaces
The shared object libcontract.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Function Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>ct_ctl_abandon</td>
<td>ct_ctl_ack</td>
</tr>
<tr>
<td>ct_ctl_adopt</td>
<td>ct_ctl_newct</td>
</tr>
<tr>
<td>ct_ctl_qack</td>
<td>ct_event_free</td>
</tr>
<tr>
<td>ct_event_get_ctid</td>
<td>ct_event_get_evid</td>
</tr>
<tr>
<td>ct_event_get_flags</td>
<td>ct_event_get_nevid</td>
</tr>
<tr>
<td>ct_event_get_newct</td>
<td>ct_event_get_type</td>
</tr>
<tr>
<td>ct_event_read</td>
<td>ct_event_read_critical</td>
</tr>
<tr>
<td>ct_event_reliable</td>
<td>ct_event_reset</td>
</tr>
<tr>
<td>ct_pr_event_get_exitstatus</td>
<td>ct_pr_event_get_gcorefile</td>
</tr>
<tr>
<td>ct_pr_event_get_pcorefile</td>
<td>ct_pr_event_get_pid</td>
</tr>
<tr>
<td>ct_pr_event_get_ppid</td>
<td>ct_pr_event_get_sender</td>
</tr>
<tr>
<td>ct_pr_event_get_senderct</td>
<td>ct_pr_event_get_signal</td>
</tr>
<tr>
<td>ct_pr_event_get_zcorefile</td>
<td>ct_pr_status_get_contracts</td>
</tr>
<tr>
<td>ct_pr_status_get_fatal</td>
<td>ct_pr_status_get_members</td>
</tr>
<tr>
<td>ct_pr_status_get_param</td>
<td>ct_pr_tmpl_get_fatal</td>
</tr>
<tr>
<td>ct_pr_tmpl_get_param</td>
<td>ct_pr_tmpl_get_transfer</td>
</tr>
<tr>
<td>ct_pr_tmpl_set_fatal</td>
<td>ct_pr_tmpl_set_param</td>
</tr>
<tr>
<td>ct_pr_tmpl_set_transfer</td>
<td>ct_status_free</td>
</tr>
<tr>
<td>ct_status_get_cookie</td>
<td>ct_status_get_critical</td>
</tr>
<tr>
<td>ct_status_get_holder</td>
<td>ct_status_get_id</td>
</tr>
<tr>
<td>ct_status_get_informative</td>
<td>ct_status_get_nevents</td>
</tr>
<tr>
<td>ct_status_get_nevid</td>
<td>ct_status_get_ntime</td>
</tr>
</tbody>
</table>
ct_status_get_qtime  ct_status_get_state
c_t_status_get_type  ct_status_get_zoneid
ct_status_read  ct_tmpl_activate
ct_tmpl_clear  ct_tmpl_create
ct_tmpl_get_cookie  ct_tmpl_get_critical
ct_tmpl_get_informative  ct_tmpl_set_cookie
ct_tmpl_set_critical  ct_tmpl_set_informative

Files
/usr/lib/libcontract.so.1 shared object
/usr/lib/64/libcontract.so.1 64–bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Evolving</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also pvs(1), Intro(3), contract(4), attributes(5), lfcompile(5)
**Name** libcpc – CPU performance counter library

**Synopsis**

```
cc [ flag... ] file... -lcpc [ library... ]
```

**Description**

Functions in this library provide access to CPU performance counters on platforms that contain the appropriate hardware.

**Interfaces**

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

```c

cpc_access    cpc_bind_cpu

cpc_bind_curlwp    cpc_bind_event

cpc_bind_pctx    cpc_buf_add

cpc_buf_copy    cpc_buf_create

cpc_buf_destroy    cpc_buf_get

cpc_buf_hrtme    cpc_buf_set

cpc_buf_sub    cpc_buf_tick

cpc_buf_zero    cpc_caps

cpc_cciname    cpc_close

cpc_cpuref    cpc_count_sys_events

cpc_count_usr_events    cpc_disable

cpc_enable    cpc_event_accum

cpc_event_diff    cpc_eventtostr

cpc_getcciname    cpc_getcpuref

cpc_getcpuver    cpc_getnpic

cpc_getusage    cpc_npic

cpc_open    cpc_pctx_bind_event

cpc_pctx_invalidate    cpc_pctx_rele

cpc_pctx_take_sample    cpc_rele

cpc_request_preset    cpc_set_add_request

cpc_set_create    cpc_set_destroy

cpc_set_restart    cpc_set_sample

cpc_seterrfn    cpc_seterrhndlr
```
cpc_shared_bind_event  cpc_shared_close
  cpc_shared_open     cpc_shared_rele
  cpc_shared_take_sample cpc_strtoevent
  cpc_take_sample     cpc_unbind
  cpc_version         cpc_walkAttrs
  cpc_walk_events_all cpc_walk_events_pic
  cpc_walk_names      cpc_walk_requests

Files
/usr/lib/libcpc.so.1  shared object
/usr/lib/64/libcpc.so.1  64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcpcu</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also  cputrack(1), cpustat(1M), Intro(3), cpc(3CPC), attributes(5)
libcrypt(3LIB)

**Name**  libcrypt – encryption/decryption library

**Synopsis**  
```
cc [ flag... ] file... -lcrypt [ library... ]
```

**Description**  
Functions in this library provide encoding and decoding handling routines.

**Interfaces**  
The shared object `libcrypt.so.1` provides the public interfaces defined below. See `Intro(3)` for additional information on shared object interfaces.

```
crypt encrypt setkey
```

**Files**  
```
/usr/lib/libcrypt.so.1      shared object
/usr/lib/64/libcrypt.so.1   64-bit shared object
```

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

**See Also**  
`crypt(1), Intro(3), encrypt(3C), setkey(3C), attributes(5) `
Library Interfaces and Headers

**Name**  libcurses, libtermcap, libtermlib – screen handling and optimization library

**Synopsis**  
```
cc [ flag... ] file... -lcurses [ library... ]
```

**Description**  Functions in the libcurses library provide a terminal-independent method of updating character screens with reasonable optimization. The libtermcap and libtermlib libraries are identical to libcurses and are maintained for backward compatibility.

See `libcurses(3XCURSES)` for information about the curses library that conforms to X/Open Curses, Issue 4, Version 2.

**Interfaces**  The shared objects libcurses.so.1, libtermcap.so.1, and libtermlib.so.1 provide the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

```
_getsyx
_getyx
_meta
_ring
_setecho
_setnionl
_setqiflush
_addch
_addchstr
_addnstr
_addnstr
_addwch
_addwchstr
_addwchstr
_addwstr
_attroff
_attron
_attrset
_baudrate
_beep
_bkgd
_bkgdset
_border
_box
_can_change_color
_cbreak
_clear
_clearok
_clrtobot
_clrtoeol
_color_content
_copywin
_crmode
_curs_get
_curs_set
_def_prog_mode
_def_shell_mode
_delay_output
```
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>delch</td>
<td>delete ln</td>
</tr>
<tr>
<td>delkeymap</td>
<td>delscreen</td>
</tr>
<tr>
<td>delwin</td>
<td>derwin</td>
</tr>
<tr>
<td>doupdate</td>
<td>dupwin</td>
</tr>
<tr>
<td>echo</td>
<td>echar</td>
</tr>
<tr>
<td>echowchar</td>
<td>endwin</td>
</tr>
<tr>
<td>erase</td>
<td>erasechar</td>
</tr>
<tr>
<td>filter</td>
<td>flash</td>
</tr>
<tr>
<td>flushinp</td>
<td>getbmap</td>
</tr>
<tr>
<td>getch</td>
<td>getmouse</td>
</tr>
<tr>
<td>getnwstr</td>
<td>getstr</td>
</tr>
<tr>
<td>getwch</td>
<td>getwin</td>
</tr>
<tr>
<td>getwstr</td>
<td>halfdelay</td>
</tr>
<tr>
<td>has_colors</td>
<td>has_ic</td>
</tr>
<tr>
<td>has_il</td>
<td>idcok</td>
</tr>
<tr>
<td>idlok</td>
<td>immedok</td>
</tr>
<tr>
<td>inch</td>
<td>inchstr</td>
</tr>
<tr>
<td>inchstr</td>
<td>init_color</td>
</tr>
<tr>
<td>init_pair</td>
<td>initscr</td>
</tr>
<tr>
<td>innstr</td>
<td>innwstr</td>
</tr>
<tr>
<td>insch</td>
<td>insdelln</td>
</tr>
<tr>
<td>insertln</td>
<td>insnstr</td>
</tr>
<tr>
<td>insnwstr</td>
<td>insstr</td>
</tr>
<tr>
<td>instr</td>
<td>inswch</td>
</tr>
<tr>
<td>inswstr</td>
<td>intrflush</td>
</tr>
<tr>
<td>inwch</td>
<td>inwchnstr</td>
</tr>
<tr>
<td>inwchstr</td>
<td>inwstr</td>
</tr>
<tr>
<td>is_linetouched</td>
<td>is_wintouched</td>
</tr>
<tr>
<td>isendwin</td>
<td>keyname</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>keypad</td>
<td>killchar</td>
</tr>
<tr>
<td>leaveok</td>
<td>longname</td>
</tr>
<tr>
<td>m_addch</td>
<td>m_addstr</td>
</tr>
<tr>
<td>m_clear</td>
<td>m_erase</td>
</tr>
<tr>
<td>m_initscr</td>
<td>m_move</td>
</tr>
<tr>
<td>m_newterm</td>
<td>m_refresh</td>
</tr>
<tr>
<td>map_button</td>
<td>meta</td>
</tr>
<tr>
<td>mouse_off</td>
<td>mouse_on</td>
</tr>
<tr>
<td>mouse_set</td>
<td>move</td>
</tr>
<tr>
<td>mvaddch</td>
<td>mvaddchnstr</td>
</tr>
<tr>
<td>mvaddchstr</td>
<td>mvaddnstr</td>
</tr>
<tr>
<td>mvaddnwstr</td>
<td>mvaddstr</td>
</tr>
<tr>
<td>mvaddwch</td>
<td>mvaddwchnstr</td>
</tr>
<tr>
<td>mvaddwchstr</td>
<td>mvaddwstr</td>
</tr>
<tr>
<td>mvcur</td>
<td>mvdelch</td>
</tr>
<tr>
<td>mvderwin</td>
<td>mvgetch</td>
</tr>
<tr>
<td>mvgetnstr</td>
<td>mvgetstr</td>
</tr>
<tr>
<td>mvgetwch</td>
<td>mvgetwstr</td>
</tr>
<tr>
<td>mvinch</td>
<td>mvinchstr</td>
</tr>
<tr>
<td>mvinchstr</td>
<td>mvinnstr</td>
</tr>
<tr>
<td>mvinnwstr</td>
<td>mvinsch</td>
</tr>
<tr>
<td>mvinsnstr</td>
<td>mvinsnwstr</td>
</tr>
<tr>
<td>mvinsstr</td>
<td>mvinsnstr</td>
</tr>
<tr>
<td>mvinswch</td>
<td>mvinswstr</td>
</tr>
<tr>
<td>mvinwch</td>
<td>mvinwchnstr</td>
</tr>
<tr>
<td>mvinwchstr</td>
<td>mvinwstr</td>
</tr>
<tr>
<td>mvprintw</td>
<td>mvscanw</td>
</tr>
<tr>
<td>mvwaddch</td>
<td>mvwaddchnstr</td>
</tr>
<tr>
<td>mvwaddchstr</td>
<td>mvwaddnstr</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>mvwaddnwstr</td>
<td>mvwaddstr</td>
</tr>
<tr>
<td>mvwaddwch</td>
<td>mvwaddwchnstr</td>
</tr>
<tr>
<td>mvwaddwchstr</td>
<td>mvwaddwstr</td>
</tr>
<tr>
<td>mvwdelch</td>
<td>mvwgetch</td>
</tr>
<tr>
<td>mvwgetnwstr</td>
<td>mvwgetstr</td>
</tr>
<tr>
<td>mvwgetwch</td>
<td>mvwgetwstr</td>
</tr>
<tr>
<td>mvwin</td>
<td>mvwinch</td>
</tr>
<tr>
<td>mvwinchnstr</td>
<td>mvwinchstr</td>
</tr>
<tr>
<td>mvwinnstr</td>
<td>mvwinnwstr</td>
</tr>
<tr>
<td>mvwinsch</td>
<td>mvwinsnstr</td>
</tr>
<tr>
<td>mvwinsnwstr</td>
<td>mvwinsstr</td>
</tr>
<tr>
<td>mvwinstr</td>
<td>mvwinswch</td>
</tr>
<tr>
<td>mvwinwstr</td>
<td>mvwinwch</td>
</tr>
<tr>
<td>mvwinwchstr</td>
<td>mvwinwchstr</td>
</tr>
<tr>
<td>mwwinwstr</td>
<td>mwwprintw</td>
</tr>
<tr>
<td>mwwscanw</td>
<td>napms</td>
</tr>
<tr>
<td>newkey</td>
<td>newpad</td>
</tr>
<tr>
<td>newscreen</td>
<td>newterm</td>
</tr>
<tr>
<td>newwin</td>
<td>nl</td>
</tr>
<tr>
<td>nocbreak</td>
<td>nocrmode</td>
</tr>
<tr>
<td>nodelay</td>
<td>noecho</td>
</tr>
<tr>
<td>nonl</td>
<td>noqiflush</td>
</tr>
<tr>
<td>noraw</td>
<td>notimeout</td>
</tr>
<tr>
<td>overlay</td>
<td>overwrite</td>
</tr>
<tr>
<td>pair_content</td>
<td>pechocchar</td>
</tr>
<tr>
<td>pechowchar</td>
<td>pnoutrefresh</td>
</tr>
<tr>
<td>prefresh</td>
<td>printw</td>
</tr>
<tr>
<td>putp</td>
<td>putwin</td>
</tr>
<tr>
<td>qiflush</td>
<td>raw</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------</td>
</tr>
<tr>
<td>redrawwin</td>
<td>refresh</td>
</tr>
<tr>
<td>request_mouse_pos</td>
<td>reset_prog_mode</td>
</tr>
<tr>
<td>reset_shell_mode</td>
<td>resetty</td>
</tr>
<tr>
<td>restartterm</td>
<td>ripoffline</td>
</tr>
<tr>
<td>savetty</td>
<td>scanw</td>
</tr>
<tr>
<td>scr_dump</td>
<td>scr_init</td>
</tr>
<tr>
<td>scr_restore</td>
<td>scr_set</td>
</tr>
<tr>
<td>scrl</td>
<td>scroll</td>
</tr>
<tr>
<td>scrollok</td>
<td>set_term</td>
</tr>
<tr>
<td>setcurscreen</td>
<td>setscrreg</td>
</tr>
<tr>
<td>setsyx</td>
<td>setterm</td>
</tr>
<tr>
<td>setupterm</td>
<td>slk_attroff</td>
</tr>
<tr>
<td>slk_attron</td>
<td>slk_attrset</td>
</tr>
<tr>
<td>slk_clear</td>
<td>slk_init</td>
</tr>
<tr>
<td>slk_label</td>
<td>slk_noutrefresh</td>
</tr>
<tr>
<td>slk_refresh</td>
<td>slk_restore</td>
</tr>
<tr>
<td>slk_set</td>
<td>slk_start</td>
</tr>
<tr>
<td>slk_touch</td>
<td>standend</td>
</tr>
<tr>
<td>standout</td>
<td>start_color</td>
</tr>
<tr>
<td>subpad</td>
<td>subwin</td>
</tr>
<tr>
<td>syncok</td>
<td>termattrs</td>
</tr>
<tr>
<td>termmame</td>
<td>tgetent</td>
</tr>
<tr>
<td>tgetflag</td>
<td>tgetnum</td>
</tr>
<tr>
<td>tgetstr</td>
<td>tgoto</td>
</tr>
<tr>
<td>tigetflag</td>
<td>tigetnum</td>
</tr>
<tr>
<td>tigetstr</td>
<td>timeout</td>
</tr>
<tr>
<td>touchline</td>
<td>touchwin</td>
</tr>
<tr>
<td>tparm</td>
<td>tputs</td>
</tr>
<tr>
<td>traceoff</td>
<td>traceon</td>
</tr>
</tbody>
</table>
typeahead  unctrl
ungetch  ungetwch
untouchwin  vidattr
vidputs  vidupdate
vvprintw  vvscanw
waddch  waddchnstr
waddchstr  waddnstr
waddnwstr  waddstr
waddwch  waddwchnstr
waddwchstr  waddwstr
wadjcurspos  wattroff
wattron  wattrset
wbkgd  wbkgdset
wborder  wcldear
wcrltobot  wclrtoeol
wcursyncup  wdelch
wdeleteln  wechochar
wechowchar  werase
wgetch  wgetnstr
wgetnwstr  wgetstr
wgetwch  wgetwstr
whline  winch
winchnstr  winchstr
winnstr  winnwstr
winsch  winsdelln
winsertln  winsnstr
winsmwstr  winsstr
winstr  winswch
winswstr  winwch
### libcurses(3LIB)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>winwchnstr</td>
<td>winwchstr</td>
</tr>
<tr>
<td>winwstr</td>
<td>wmouse_position</td>
</tr>
<tr>
<td>wmove</td>
<td>wmovenextch</td>
</tr>
<tr>
<td>wmoveprevch</td>
<td>wnoutrefresh</td>
</tr>
<tr>
<td>wprintw</td>
<td>wredrawln</td>
</tr>
<tr>
<td>wrefresh</td>
<td>wscamw</td>
</tr>
<tr>
<td>wscrl</td>
<td>wsetscrreg</td>
</tr>
<tr>
<td>wstandend</td>
<td>wstandout</td>
</tr>
<tr>
<td>wsyncdown</td>
<td>wsyncup</td>
</tr>
<tr>
<td>wtimeout</td>
<td>wtouchn</td>
</tr>
<tr>
<td>wvline</td>
<td></td>
</tr>
</tbody>
</table>

### Files

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

### Attributes

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

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

### See Also

Intro(3), curses(3CURSES), libcurses(3LIBUCB), libcurses(3XCURSES), attributes(5)
**Name**
libcurses – SunOS/BSD-compatible screen handling and optimization library

**Synopsis**
```
cc [ flag... ] -I /usr/ucbinclude file... -L /usr/libucb -R /usr/libucb -lcurses [ library... ]
```

**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 libcurses.so.1 provides the public interfaces defined below. See *Intro(3)* 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 E0 GT HC
H0 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  
_pfast  _rawmode  _res_flg  _tty  
_tty_ch  _unctrl  box  curscr  
delwin  endwin  getcap  gettmode  
idlok  initscr  longname  mvcurs  
mvprintw  mvscannw  mvwin  mvprintw  
mwscannw  newwin  normtty  overlay  
overwrite  printw  scanw  scroll  
setterm  stdscr  subwin  touchline  
touchwin  ttytype  waddch  waddstr  
wclear  wclrtobot  wclrtoeol  wdelch  
wdelteeln  werase  wgetach  wgetstr  
winsch  winsertln  wmove  wprintw  
wrefresh  wscannw  wstandend  wstandout  

**Files**

/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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

**See Also**

Intro(3), libcurses(3LIB), libcurses(3XCURSES), attributes(5)
# libdat

The `libdat` library provides an application with the User Direct Access Programming Library (uDAPL) 1.2 functions to access the underlying RDMA-able interconnects. Different uDAPL service providers listed in the DAT static registry `dat.conf(4)` can be registered during runtime with the DAT library. After an application opens an interface adapter belonging to a particular service provider, all function calls will be redirected to that service provider's library.

## Description

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

### uDAPL 1.1

- `dat_cno_create`
- `dat_cno_free`
- `dat_cno_modify_agent`
- `dat_cno_query`
- `dat_cno_wait`
- `dat_cr_accept`
- `dat_cr_handoff`
- `dat_cr_query`
- `dat_cr_reject`
- `dat_ep_connect`
- `dat_ep_create`
- `dat_ep_disconnect`
- `dat_ep_dup_connect`
- `dat_ep_free`
- `dat_ep_get_status`
- `dat_ep_modify`
- `dat_ep_post_rdma_read`
- `dat_ep_post_rdma_write`
- `dat_ep_post_recv`
- `dat_ep_post_send`
- `dat_ep_query`
- `dat_ep_reset`
- `dat_evd_clear_unwaitable`
- `dat_evd_create`
- `dat_evd_dequeue`
- `dat_evd_disable`
- `dat_evd_enable`
- `dat_evd_free`
- `dat_evd_modify_cno`
- `dat_evd_post_send`
- `dat_evd_query`
- `dat_evd_resize`
- `dat_evd_set_unwaitable`
- `dat_evd_wait`
- `dat_get_consumer_context`
- `dat_get_handle_type`
- `dat_ia_close`
- `dat_ia_open`
- `dat_ia_query`
- `dat_lmr_create`
The shared object `libdat.so.1` also provides the public interfaces defined below for service providers.

```
dat_registry_add_provider  dat_registry_remove_provider
```

**Files**
- `/usr/lib/libdat.so.1` shared object
- `/usr/lib/64/libdat.so.1` 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWudaplu (user)</td>
</tr>
<tr>
<td></td>
<td>SUNWudapl (root)</td>
</tr>
</tbody>
</table>
See Also  datadm(1M), Intro(3), dat.conf(4), attributes(5)

Notes  The libdat library supports service providers written according to the uDAPL 1.2 specification. A service provider library has to be a dynamic loadable shared object with two public entry points exported:

  dat_provider_init             dat_provider_fini

In terms of installation, the service provider package should include a service_provider.conf(4) file. The datadm(1M) administrative configuration program should be used to add and remove service provider’s entries in the system-wide dat.conf(4).
**Name**  
libdbm – database subroutines library  

**Synopsis**  
```  
cc [flag... ] -I /usr/ucbinclude file... -L /usr/libucb \  
- R /usr/libucb -ldbm [ 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.

<table>
<thead>
<tr>
<th>bitno</th>
<th>blkno</th>
</tr>
</thead>
<tbody>
<tr>
<td>dbmclose</td>
<td>dbminit</td>
</tr>
<tr>
<td>dbrdonly</td>
<td>delete</td>
</tr>
<tr>
<td>dirbuf</td>
<td>dirf</td>
</tr>
<tr>
<td>fetch</td>
<td>firstkey</td>
</tr>
<tr>
<td>hmask</td>
<td>maxbno</td>
</tr>
<tr>
<td>nextkey</td>
<td>pagbuf</td>
</tr>
<tr>
<td>pagf</td>
<td>store</td>
</tr>
</tbody>
</table>

**Files**  
- `/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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

**See Also**  
`Intro(3), dbm(3UCB), attributes(5)`
**Name**  
libdevid – device ID library

**Synopsis**  
cc [ flag... ] file... -ldevid [ library... ]  
#include <devid.h>

**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 `libdevid.so.1` provides the public interfaces defined below. See *Intro(3)* for additional information on shared object interfaces.

- `devid_compare`
- `devid_free`
- `devid_get`
- `devid_sizeof`
- `devid_str_decode`
- `devid_str_encode`
- `devid_str_free`
- `devid_valid`
- `devid_deviceid_to_nmlist`
- `devid_free_nmlist`
- `devid_get_minor_name`
- `devid_str_decode`
- `devid_str_free`

**Files**  
`/lib/libdevid.so.1` shared object.  
`/lib/64/libdevid.so.1` 64-bit shared object.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Stable</td>
</tr>
<tr>
<td>MT-Level</td>
<td>MT-Safe</td>
</tr>
</tbody>
</table>

**See Also**  
pvs(1), *Intro(3)*, attributes(5)
libdevinfo – device information library

Synopsis

cc [ flag... ] file... -ldevinfo [ library... ]
#include <libdevinfo.h>

Description

Functions in this library access device configuration information.

Device configuration data is organized as a tree of device nodes, defined as `di_node_t` in the `libdevinfo` interfaces. Each `di_node_t` represents a physical or logical (pseudo) device. Three types of data are associated with device nodes:

- data defined for all device nodes (attributes)
- properties specific to each device
- minor node data

All device nodes have a set of common attributes, such as a node name, an instance number, and a driver binding name. Common device node attributes are accessed by calling interfaces listed on the `di_binding_name(3DEVINFO)` manual page. Each device node also has a physical path, which is accessed by calling `di_devfs_path(3DEVINFO)`.

Properties provide device specific information for device configuration and usage. Properties can be defined by software (`di_prop_t`) or by firmware (`di_prom_prop_t`). One way to access each `di_prop_t` is to make successive calls to `di_prop_next(3DEVINFO)` until `DI_PROP_NIL` is returned. For each `di_prop_t`, use interfaces on the `di_prop_bytes(3DEVINFO)` manual page to obtain property names and values. Another way to access these properties is to call `di_prop_lookup_bytes(3DEVINFO)` to find the value of a property with a given name. Accessing a `di_prom_prop_t` is similar to accessing a `di_prop_t`, except that the interface names start with `di_prom_prop` and additional calls to `di_prom_init(3DEVINFO)` and `di_prom_fini(3DEVINFO)` are required.

Minor nodes contain information exported by the device for creating special files for the device. Each device node has 0 or more minor nodes associated with it. A list minor nodes (`di_minor_t`) can be obtained by making successive calls to `di_minor_next(3DEVINFO)` until `DI_MINOR_NIL` is returned. For each minor node, `di_minor_devt(3DEVINFO)` and related interfaces are called to get minor node data.

Using `libdevinfo` involves three steps:

- Creating a snapshot of the device tree
- Traversing the device tree to get information of interest
- Destroying the snapshot of the device tree

A snapshot of the device tree is created by calling `di_init(3DEVINFO)` and destroyed by calling `di_fini(3DEVINFO)`. An application can specify the data to be included in the snapshot (full or partial tree, include or exclude properties and minor nodes) and get a handle to the root of the device tree. See `di_init(3DEVINFO)` for details. The application then traverses the device tree in the snapshot to obtain device configuration data.
The device tree is normally traversed through parent-child-sibling linkage. Each device node contains references to its parent, its next sibling, and the first of its children. Given the `di_node_t` returned from `di_init()`, one can find all children by first calling `di_child_node(3DEVINFO)`, followed by successive calls to `di_sibling_node(3DEVINFO)` until `DI_NODE_NIL` is returned. By following this procedure recursively, an application can visit all device nodes contained in the snapshot. Two interfaces, the `di_walk_node(3DEVINFO)` and `di_walk_minor(3DEVINFO)` functions are provided to facilitate device tree traversal. The `di_walk_node()` function visits all device nodes and executes a user-supplied callback function for each node visited. The `di_walk_minor()` function does the same for each minor node in the device tree.

An alternative way to traverse the device tree is through the per-driver device node linkage. Device nodes contain a reference to the next device node bound to the same driver. Given the `di_node_t` returned from `di_init()`, an application can find all device nodes bound to a driver by first calling `di_drv_first_node(3DEVINFO)`, followed by successive calls to `di_drv_next_node(3DEVINFO)` until `DI_NODE_NIL` is returned. Traversing the per-driver device node list works only when the snapshot includes all device nodes.

See `di_init(3DEVINFO)` for examples of `libdevinfo` usage. See Writing Device Drivers for information about Solaris device configuration.

### Interfaces

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

**EXAMPLE 1**  Information accessible through libdevinfo interfaces

The following example illustrates the kind of information accessible through libdevinfo interfaces for a device node representing a hard disk (sd2):

**Attributes**

- node name: sd
- instance: 2
- physical path: /sbus@1f,0/espdma@e,8400000/esp@e,8800000/sd@2,0

**Properties**

- target=2
- lun=0

**Minor nodes**

(disk partition /dev/dsk/c0t2d0s0)
EXAMPLE 1 Information accessible through libdevinfo interfaces  

(name: a  

 dev_t: 0x0080010 (32/16)  

 spectype: IF_BLK (block special)  

 (disk partition /dev/rdsk/c0t2d0s2)  

 (name: c,raw  

 dev_t: 0x0080012 (32/18)  

 spectype: IF_CHR (character special)  

 (Files  

 /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:  

<table>
<thead>
<tr>
<th>ATTRIBUTE</th>
<th>ATTRIBUTE VALUE</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl, SUNWstatl (32–bit)</td>
<td>SUNWcsdx (64–bit)</td>
<td></td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Evolving</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

See Also  

 pvs(1), devlinks(1M), prtconf(1M), Intro(3), di_binding_name(3DEVINFO), di_child_node(3DEVINFO), di_devfs_path(3DEVINFO), di_drv_first_node(3DEVINFO), di_drv_next_node(3DEVINFO), di_fini(3DEVINFO), di_prom_init(3DEVINFO), di_minor_devt(3DEVINFO), di_minor_next(3DEVINFO), di_prom_fini(3DEVINFO), di_prom_init(3DEVINFO), di_prop_bytes(3DEVINFO), di_prop_lookup_bytes(3DEVINFO), di_prop_next(3DEVINFO), di_sibling_node(3DEVINFO), di_walk_minor(3DEVINFO), di_walk_node(3DEVINFO), attributes(5)  

Writing Device Drivers
libdl(3LIB)

Name  libdl – dynamic linking library

Synopsis  cc [ flag... ] file... -ldl [ library... ]

Description  Historically, functions in libdl provided for dynamic linking support. This functionality now resides in libc(3LIB).

This library is maintained to provide backward compatibility for both runtime and compilation environments. The shared object is implemented as a filter on the runtime linker. See ld.so.1(1). New application development need not specify -ldl.

Interfaces  The shared object libdl.so.1 provides the following public interfaces. See Intro(3) for additional information on shared object interfaces.

    dladdr dladdr1
    dlclose dldump
dlerror dlinfo
dlmopen dlopen
dlsym

Files  /lib/libdl.so.1 shared object
       /lib/64/libdl.so.1 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also  ld.so.1(1), pvs(1), Intro(3), libc(3LIB), attributes(5)
libdlpi – Data Link Provider Interface (DLPI) library

**Synopsis**

```
cc [ flag... ] file... -ldlpi [ library... ]
#include <libdlpi.h>
```

**Description**
The `libdlpi` library provides functions that support a programming interface for DLPI applications. The functions support only DLPI Version 2 devices in connectionless mode.

**Interfaces**
The shared object `/lib/libdlpi.so.1` provides the public interfaces defined below. See `Intro(3)` for additional information on shared object interfaces.

```
dlpi_arptype dlpi_bind
dlpi_close dlpi_disabmulti
dlpi_disabnotify dlpi_enabmulti
dlpi_enabnotify dlpi_fd
dlpi_get_physaddr dlpi_iftype
dlpi_info dlpi_linkname
dlpi_mactype dlpi_open
dlpi_promiscoff dlpi_promiscon
dlpi_recv dlpi_send
dlpi_set_physaddr dlpi_set_timeout
dlpi_strerror dlpi_unbind
dlpi_walk
```

**Files**

- `/lib/libdlpi.so.1` shared object
- `/lib/64/libdlpi.so.1` 64–bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Committed</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

**See Also**
`Intro(3), attributes(5)`
**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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWsadmi (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWsadmx (64-bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Obsolete</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

**See Also**  
Intro(3), libdmici(3LIB), libdmimi(3LIB), attributes(5)
libdmici (3LIB)

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

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

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWsadmi (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWsadmx (64-bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Obsolete</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

See Also  Intro(3), libdmi(3LIB), attributes(5)
libdmimi

**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.

- `ConnectToServer`
- `DisconnectToServer`
- `DmiAddComponent`
- `DmiAddGroup`
- `DmiAddLanguage`
- `DmiAddRow`
- `DmiDeleteComponent`
- `DmiDeleteGroup`
- `DmiDeleteLanguage`
- `DmiDeleteRow`
- `DmiGetAttribute`
- `DmiGetConfig`
- `DmiGetMultiple`
- `DmiGetVersion`
- `DmiListAttributes`
- `DmiListClassNames`
- `DmiListComponents`
- `DmiListComponentsByClass`
- `DmiListGroups`
- `DmiListLanguages`
- `DmiRegister`
- `DmiSetAttribute`
- `DmiSetConfig`
- `DmiSetMultiple`
- `DmiUnregister`

**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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWSadmi (32–bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWSadmx (64–bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Obsolete</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>
See Also  Intro(3), libdmi(3LIB), attributes(5)
**Synopsis**  
cc [ flag... ] file... -ldoor [ library... ]  
#include <door.h>

**Description**  
The functions in this library provide programmatic access to doors, including the ability to create and call them. Doors are a fast light-weight RPC mechanism for secure control transfer between processes on the same machine. Conceptually, a thread in one process can issue a call using a door descriptor that causes code to be executed in another process and then returns using the traditional synchronous RPC model. Doors can also be used to pass data and file descriptors between processes.

**Interfaces**  
The shared object `libdoor.so.1` provides the public interfaces defined below. See `Intro(3)` for additional information on shared object interfaces.

- `door_bind`  
- `door_create`  
- `door_info`  
- `door_revoke`  
- `door_ucred`  
- `door_xcreate`  
- `door_call`  
- `door_cred`  
- `door_return`  
- `door_server_create`  
- `door_unbind`

**Files**  
/lib/libdoor.so.1  
shared object  
/lib/64/libdoor.so.1  
64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Evolving</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

**See Also**  
`Intro(3), door_bind(3DOOR), door_call(3DOOR), door_create(3DOOR), door_cred(3DOOR), door_info(3DOOR), door_return(3DOOR), door_revoke(3DOOR), door_server_create(3DOOR), door_ucred(3DOOR), attributes(5)`

Name    libdtrace – DTrace dynamic tracing software library

Description Functions in this library define the interface for interacting with the DTrace dynamic tracing software, including the D language compiler and facilities for enabling probes and consuming trace data.

Interfaces The interfaces provided by libdtrace.so.1 are currently private to the implementation of the Solaris system and DTrace subsystem and are subject to change at any time without notice. Applications using these interfaces might fail to run on future releases. Refer to the Solaris Dynamic Tracing Guide for a description of the public documented interfaces available for the DTrace facility.

Files    /usr/lib/libdtrace.so.1 shared object
           /usr/lib/64/libdtrace.so.1  64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWdtrc</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Private</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

See Also  dtrace(1M), attributes(5), dtrace(7D)

Solaris Dynamic Tracing Guide
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.

    efi_alloc_and_init  efi_alloc_and_read
    efi_free           efi_write

Files /lib/libefi.so.1    shared object
       /lib/64/libefi.so.1  64–bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32–bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64–bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Evolving</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

See Also Intro(3), efi_alloc_and_init(3EXT), attributes(5)
**Name**  
libelf – ELF access library

**Synopsis**  
cc [ flag... ] file... -lelf [ library... ]  
#include <libelf.h>

**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 libelf.so.1 provides the public interfaces defined below. See `Intro(3)` for additional information on shared object interfaces.

```c
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_getshnum    elf_getshstrndx
elf_hash        elf_kind
elf_memory      elf_ndxscn
elf_newdata     elf_newscn
```
```
elf_next  elf_nextscn
elf_rand  elf_rawdata
elf_rawfile  elf_strptr
elf_update  elf_version
gelf_checksum  gelf_fsize
gelf_getcap  gelf_getclass
gelf_getdyn  gelf_getehdr
gelf_getmove  gelf_getphdr
gelf_getrel  gelf_getrela
gelf_getshdr  gelf_getsym
gelf_getsyminfo  gelf_getsymshndx
gelf_newehdr  gelf_newphdr
gelf_update_cap  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_symshndx
gelf_update_syminfo  gelf_xlatetof
gelf_xlatetom  nlist
```

**Files**

- `/lib/libelf.so.1` shared object
- `/lib/64/libelf.so.1` 64-bit shared object

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

**See Also**

pvs(1), Intro(3), elf(3ELF), gelf(3ELF), attributes(5)
libexacct(3LIB)

Name  libexacct — extended accounting file access library

Synopsis  cc [ flag... ] file... -lexacct [ library... ]
          #include <exacct.h>

Description  Functions in this library define the interface for reading and writing extended accounting
              (exacct) files. The <exacct.h> 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 libexacct.so.1 provides the public interfaces defined below. See Intro(3)
              for additional information on shared object interfaces.

              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

Files  /usr/lib/libexacct.so.1   shared object
       /usr/lib/64/libexacct.so.1  64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcs1 (32–bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcs1x (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>
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.

See Also  acctadm(1M), Intro(3), ea_error(3EXACCT), ea_open(3EXACCT), ea_pack_object(3EXACCT), ea_set_item(3EXACCT), attributes(5)
libfmevent(3LIB)

Name  libfmevent – fault management events library

Synopsis  cc [ flag...] file... -L/usr/lib/fm -lfmevent -lnvpair [ library...]  
#include <fm/libfmevent.h>  
#include <libnvpair.h>

Description  This library allows a process to subscribe to a subset of fault management protocol events  
published by the fault management daemon.

Interfaces  The shared object libfmevent.so.1 provides the public interfaces defined below. See  
Intro(3) for additional information on shared object interfaces.

fmev_attr_list  fmev_class
fmev_dup  fmev_errno
fmev_hold  fmev_localtime
fmev_rele  fmev_shdl_alloc
fmev_shdl_fini  fmev_shdl_free
fmev_shdl_init  fmev_shdl_subscribe
fmev_shdl_unsubscribe  fmev_shdl_zalloc
fmev_shdctl.serialized  fmev_shdctl_sigmask
fmev_shdctl thrattr  fmev_shdctl thrcreate
fmev_shdctl thrsetup  fmev_strerror
fmev_time_nsec  fmev_time_sec
fmev timespec

Files  usr/lib/fm/libfmevent.so.1  shared object
usr/lib/fm/64/libfmevent.so.1  64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Architecture</td>
<td>all</td>
</tr>
<tr>
<td>Availability</td>
<td>SUNWfmd</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Committed</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>
See Also  Intro(3), fmev_shdl_init(3FM), libnvpair(3LIB), attributes(5)
libform(3LIB)

Name  libform – forms library

Synopsis  cc [ flag... ] file... -lform [ library... ]

Description  Functions in this library provide forms using libcurses(3LIB) routines.

Interfaces  The shared object libform.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>current_field</td>
<td>data_ahead</td>
</tr>
<tr>
<td>data_behind</td>
<td>dup_field</td>
</tr>
<tr>
<td>dynamic_field_info</td>
<td>field_arg</td>
</tr>
<tr>
<td>field_back</td>
<td>field_buffer</td>
</tr>
<tr>
<td>field_count</td>
<td>field_fore</td>
</tr>
<tr>
<td>field_index</td>
<td>field_info</td>
</tr>
<tr>
<td>field_init</td>
<td>field_just</td>
</tr>
<tr>
<td>field_opts</td>
<td>field_opts_off</td>
</tr>
<tr>
<td>field_opts_on</td>
<td>field_pad</td>
</tr>
<tr>
<td>field_status</td>
<td>field_term</td>
</tr>
<tr>
<td>field_type</td>
<td>field_userptr</td>
</tr>
<tr>
<td>form_driver</td>
<td>form_fields</td>
</tr>
<tr>
<td>form_init</td>
<td>form_opts</td>
</tr>
<tr>
<td>form_opts_off</td>
<td>form_opts_on</td>
</tr>
<tr>
<td>form_page</td>
<td>form_sub</td>
</tr>
<tr>
<td>form_term</td>
<td>form_userptr</td>
</tr>
<tr>
<td>form_win</td>
<td>free_field</td>
</tr>
<tr>
<td>free_fieldtype</td>
<td>free_form</td>
</tr>
<tr>
<td>link_field</td>
<td>link_fieldtype</td>
</tr>
<tr>
<td>move_field</td>
<td>new_field</td>
</tr>
<tr>
<td>new_fieldtype</td>
<td>new_form</td>
</tr>
<tr>
<td>new_page</td>
<td>pos_form_cursor</td>
</tr>
<tr>
<td>post_form</td>
<td>scale_form</td>
</tr>
<tr>
<td>set_current_field</td>
<td>set_field_back</td>
</tr>
</tbody>
</table>
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.so.1  shared object
/usr/lib/64/libform.so.1  64-bit shared object

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

**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.

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

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

```
copylist64
```

**Files**  
/lib/libgen.so.1  shared object

/lib/64/libgen.so.1  64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>ATTRIBUTE TYPE</td>
<td>ATTRIBUTE VALUE</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also Intro(3), attributes(5)
libgen.h(3HEAD)

Name libgen.h, libgen – definitions for pattern matching functions

Synopsis #include <libgen.h>

Description The <libgen.h> header lists definitions used for string pattern-matching and pathname manipulation. See libgen(3LIB).

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also basename(3C), dirname(3C), libgen(3LIB), attributes(5), standards(5)
Name  libgss – Generic Security Services library

Synopsis  cc [ flag... ] file... -lgss [ library... ]
#include <gssapi/gssapi.h>

Description  The functions in this library are the routines that comprise the Generic Security Services library.

When libgss fails to load or initialize a mechanism listed in /etc/gss/mech, a message is sent to syslog(3C).

Interfaces  The shared object libgss.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GSS_C_NT_ANONYMOUS</td>
<td>GSS_C_NT_EXPORT_NAME</td>
</tr>
<tr>
<td>GSS_C_NT_HOSTBASED_SERVICE</td>
<td>GSS_C_NT_MACHINE_UID_NAME</td>
</tr>
<tr>
<td>GSS_C_NT_STRING_UID_NAME</td>
<td>GSS_C_NT_USER_NAME</td>
</tr>
<tr>
<td>gss_accept_sec_context</td>
<td>gss_acquire_cred</td>
</tr>
<tr>
<td>gss_add_cred</td>
<td>gss_add_oid_set_member</td>
</tr>
<tr>
<td>gss_canonicalize_name</td>
<td>gss_compare_name</td>
</tr>
<tr>
<td>gss_context_time</td>
<td>gss_create_empty_oid_set</td>
</tr>
<tr>
<td>gss_delete_sec_context</td>
<td>gss_display_name</td>
</tr>
<tr>
<td>gss_display_status</td>
<td>gss_duplicate_name</td>
</tr>
<tr>
<td>gss_export_name</td>
<td>gss_export_sec_context</td>
</tr>
<tr>
<td>gss_get_mic</td>
<td>gss_import_name</td>
</tr>
<tr>
<td>gss_import_sec_context</td>
<td>gss_indicate_mechs</td>
</tr>
<tr>
<td>gss_init_sec_context</td>
<td>gss_inquire_context</td>
</tr>
<tr>
<td>gss_inquire_cred</td>
<td>gss_inquire_cred_by_mech</td>
</tr>
<tr>
<td>gss_inquire_mechs_for_name</td>
<td>gss_inquire_names_for_mech</td>
</tr>
<tr>
<td>gss_process_context_token</td>
<td>gss_release_buffer</td>
</tr>
<tr>
<td>gss_release_cred</td>
<td>gss_release_name</td>
</tr>
<tr>
<td>gss_release_oid</td>
<td>gss_release_oid_set</td>
</tr>
<tr>
<td>gss_seal</td>
<td>gss_sign</td>
</tr>
<tr>
<td>gss_store_cred</td>
<td>gss_test_oid_set_member</td>
</tr>
<tr>
<td>gss_unseal</td>
<td>gss_unwrap</td>
</tr>
</tbody>
</table>
libgss(3LIB)

    gss_verify  gss_verify_mic
    gss_wrap    gss_wrap_size_limit

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

Attributes  See \texttt{attributes}(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWgss (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWgssx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also  \texttt{pvs(1)}, \texttt{Intro(2)}, \texttt{Intro(3)}, \texttt{syslog(3C)}, \texttt{attributes(5)}

\textit{Developer's Guide to Oracle Solaris Security}
The functions in this library access Fibre Channel HBA data.

Fibre Channel HBA information is provided through a standard interface in a vendor independent manner. This common interface provides access to the following information:

- Local HBA attributes
- Local HBA port attributes and statistics
- Mapping between FCP-2 discovered devices and operating system SCSI information
- Discovered devices port attributes
- SCSI commands for discovered FCP-2 devices (Report LUNS, Read Capacity, and Inquiry)
- Common Transport commands to discover Fabric details

The shared object \libhbaapi.so.1 provides the public interfaces defined below. See \Intro{(3)} for additional information on shared object interfaces.

- HBA_CloseAdapter
- HBA_FreeLibrary
- HBA_GetAdapterAttributes
- HBA_GetAdapterName
- HBA_GetAdapterPortAttributes
- HBA_GetBindingCapability
- HBA_GetBindingSupport
- HBA_GetDiscoveredPortAttributes
- HBA_GetEventBuffer
- HBA_GetFC4Statistics
- HBA_GetFCPStatistics
- HBA_GetFcpPersistentBinding
- HBA_GetFcpTargetMapping
- HBA_GetFcpTargetMappingV2
- HBA_GetNumberOfAdapters
- HBA_GetPersistentBindingV2
- HBA_GetPortAttributesByWWN
- HBA_GetRnIdMgmtInfo
- HBA_GetVersion
- HBA_GetVendorLibraryAttributes
- HBA_LoadLibrary
- HBA_OpenAdapter
- HBA_OpenAdapterByWWN
- HBA_RefreshInformation
- HBA_RegisterForAdapterAddEvents
- HBA_RegisterForAdapterEvents
- HBA_RegisterForAdapterPortStatEvents
- HBA_RegisterForLinkEvents
Client applications link with the Common Library (using `-lHBAAPI`) to access the interfaces. The Common Library dynamically loads individual Vendor-Specific Libraries (VSL) listed in `/etc/hba.conf` described on the `hba.conf(4)`.

Using the `libhbaapi` involves the following steps:

1. Optionally determining the version of the library by calling `HBA_GetVersion(3HBAAPI)`.
2. Initializing the Common Library by calling `HBA_LoadLibrary(3HBAAPI)`.
3. Determine the number of HBAs known to the common library by calling `HBA_GetNumberOfAdapters(3HBAAPI)`.
4. Determine each HBA name in turn by calling `HBA_GetAdapterName(3HBAAPI)`.
5. Open each HBA in turn by calling `HBA_OpenAdapter(3HBAAPI)`.
6. Operate on a given HBA by calling the following:
   - `HBA_GetAdapterAttributes(3HBAAPI)`
   - `HBA_GetAdapterPortAttributes(3HBAAPI)`
   - `HBA_GetDiscoveredPortAttributes(3HBAAPI)`
   - `HBA_GetPortAttributesByWWN(3HBAAPI)`
   - `HBA_SendCTPassThru(3HBAAPI)`
   - `HBA_SendCTPassThruV2(3HBAAPI)`
   - `HBA_GetEventBuffer(3HBAAPI)`
   - `HBA_SetRNIDMgmtInfo(3HBAAPI)`
   - `HBA_GetRNIDMgmtInfo(3HBAAPI)`
   - `HBA_SendRNID(3HBAAPI)`
   - `HBA_SendRNIDV2(3HBAAPI)`
   - `HBA_RefreshInformation(3HBAAPI)`
7. Close open HBAs by calling `HBA_CloseAdapter(3HBAAPI)`.
8. Unload the library by calling `HBA_FreeLibrary(3HBAAPI)`.

**Errors** Errors are generally returned from the underlying VSL and can include any of the following values:

- **HBA_STATUS_OK** Request completed successfully. (No Error)
- **HBA_STATUS_ERROR** Non-specific error encountered.
- **HBA_STATUS_ERROR_NOT_SUPPORTED** The VSL does not support this interface.
- **HBA_STATUS_ERROR_INVALID_HANDLE** The `handle` argument does not refer to an open HBA handle.
- **HBA_STATUS_ERROR_ARG** An argument in the request was invalid.
- **HBA_STATUS_ERROR_ILLEGAL_WWN** A WWN in the request was not recognized.
- **HBA_STATUS_ERROR_ILLEGAL_INDEX** An index in the request was not recognized.
- **HBA_STATUS_ERROR_MORE_DATA** A larger buffer is required to complete the requested operation.
- **HBA_STATUS_ERROR_STALE_DATA** The state of the HBA has changed, possibly due to Dynamic Reconfiguration or devices being added or removed. The caller should call `HBA_RefreshInformation(3HBAAPI)` and reissue any discovery logic to reset all indexes related to this HBA.
- **HBA_STATUS_SCSI_CHECK_CONDITION** A SCSI check-condition was encountered during the I/O operation. Not all VSLs report this error value. Some might return `HBA_STATUS_ERROR` when a check-condition is encountered, or `HBA_STATUS_OK`.

Library Interfaces and Headers 167
HBA_STATUS_ERROR_BUSY
The requested device is busy. A retry might be effective.

HBA_STATUS_ERROR_TRY_AGAIN
The requested I/O timed out. A retry might be effective.

HBA_STATUS_ERROR_UNAVAILABLE
The requested HBA has been removed or deactivated.

All other error values are reserved.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcfcl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcfclx (64-bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Standard: FC-MI 1.92 (API version 1)</td>
</tr>
<tr>
<td></td>
<td>Standard: FC-HBA Version 4 (API version 2)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also
HBA_GetAdapterAttributes(3HBAAPI), HBA_GetAdapterName(3HBAAPI),
HBA_GetAdapterPortAttributes(3HBAAPI), HBA_GetBindingCapability(3HBAAPI),
HBA_GetDiscoveredPortAttributes(3HBAAPI), HBA_GetEventBuffer(3HBAAPI),
HBA_GetFcpPersistentBinding(3HBAAPI), HBA_GetFcpTargetMapping(3HBAAPI),
HBA_GetNumberOfAdapters(3HBAAPI), HBA_GetPortAttributesByWNN(3HBAAPI),
HBA_GetPortStatistics(3HBAAPI), HBA_GetVersion(3HBAAPI),
HBA_GetWrapperLibraryAttributes(3HBAAPI), HBA_LoadLibrary(3HBAAPI),
HBA_OpenAdapter(3HBAAPI), HBA_RefreshInformation(3HBAAPI),
HBA_RegisterForAdapterEvents(3HBAAPI), HBA_SendCTPassThru(3HBAAPI),
HBA_SendRLS(3HBAAPI), HBA_SendScsiInquiry(3HBAAPI),
HBA_SetRNIDMgmtInfo(3HBAAPI), hba.conf(4), attributes(5)

T11 FC-MI Specification
Name  libidnkit – IDN conversion library

Synopsis  cc [ flag... ] file... -lidnkit [ library... ]
          #include <idn/api.h>

Description  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.

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.

Interfaces  The shared object libidnkit.so.1 provides the public interfaces defined below. See Intro(3)
for additional information on shared object interfaces.

          idn_decodename          idn_decodename2
          idn_enable             idn_encodename
          idn_nameinit

Files  /usr/lib/libidnkit.so.1  shared object
       /usr/lib/64/libidnkit.so.1  64–bit shared object

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

<table>
<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>

See Also  Intro(3),iconv(3C),idn_decodename(3EXT),setlocale(3C),hosts(4),attributes(5),
environ(5)

RFC 3490  Internationalizing Domain Names in Applications (IDNA)

RFC 3491  Nameprep: A Stringprep Profile for Internationalized Domain Names
         (IDN)

RFC 3492  Punycode: A Bootstring encoding of Unicode for Internationalized
         Domain Names in Applications (IDNA)
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
http://www.unicode.org

International Language Environments Guide (for this version of Solaris)

Copyright © 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,

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 ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.
libintl (3LIB)

**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 is implemented as a filter on libc.so.1. New application development need not specify -lintl.

**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**
/lib/libintl.so.1   a filter on /lib/libc.so.1
/lib/64/libintl.so.1 a filter on /lib/64/libc.so.1

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

```
<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcs1 (32–bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcsdx (64–bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe with exceptions</td>
</tr>
</tbody>
</table>
```

**See Also**
pvs(1), Intro(3), gettext(3C), libc(3LIB), attributes(5)
**Name**  libkrb5 – MIT Kerberos 5 library

**Synopsis**
```
c -I/usr/include/kerberosv5 [ flag... ] file... -lkrb5 [ library...]  
#include <krb5.h>  
#include <com_err.h>  
```

**Description**
The functions in this library are the routines that comprise the MIT Kerberos 5 library.

**Interfaces**
The shared object `libkrb5.so` provides the public interface defined below.

The `krb5` library is provided as a convenience to allow native `krb5` applications to be built and to run. Compatibility between Solaris releases of the `krb5` interface is not guaranteed. For new applications that require these features, `libgss(3LIB)` is recommended.

For detailed documentation on the `krb5` interface, see the MIT Kerberos 5 web site at `http://web.mit.edu/kerberos`.

The `krb5_cc_gen_new` routine, listed in `krb5.h` section, is flawed and should be avoided. Until a new routine is available from MIT, the following can be done:

```c
char ccname[40];  
ext int tmpfd;  

snprintf(ccname,sizeof(ccname),"FILE:/tmp/krb5cc_%d_XXXXXX",geteuid());  

if ((tmpfd = mkstemp(ccname+strlen("FILE:")))==-1) {  
    log("mkstemp(): %.100s", strerror(errno));  
    problem = errno;  
    goto fail;  
}  

if (fchmod(tmpfd,S_IRUSR | S_IWUSR) == -1) {  
    log("fchmod(): %.100s", strerror(errno));  
    close(tmpfd);  
    problem = errno;  
    goto fail;  
}  

close(tmpfd);  
problem = krb5_cc_resolve(authctxt->krb5_ctx, ccname, &ccache);  
...
```

**fail:**

The `krb5_string_to_key` and `krb5_string_to_key` routines, listed in `krb5.h` section, are part of the old cryptosystem and should not be used in new apps.

**com_err.h**

```c
com_err  
com_err_va  
error_message  
```
krb5.h

krb5_address_compare
krb5_address_order
krb5_address_search
krb5_aname_to_localname
krb5_appdefault_boolean
krb5_appdefault_string
krb5_auth_con_free
krb5_auth_con_genaddrs
krb5_auth_con_get_checksum_func
krb5_auth_con_getaddrs
krb5_auth_con_getauthenticator
krb5_auth_con_getflags
krb5_auth_con_getkey
krb5_auth_con_getlocalseqnumber
krb5_auth_con_getrcache
krb5_auth_con_getrecvsubkey
krb5_auth_con_getremoteseqnumber
krb5_auth_con_getsendsubkey
krb5_auth_con_init
krb5_auth_con_set_checksum_func
krb5_auth_con_setaddrs
krb5_auth_con_setflags
krb5_auth_con_setports
krb5_auth_con_setrecvsubkey
krb5_auth_con_setrcache
krb5_auth_con_setsendsubkey
krb5_auth_con_setuseruserkey
krb5_build_principal
krb5_build_principal_ext
krb5_c_block_size
krb5_c_checksum_length
krb5_c_decrypt
krb5_c_encrypt
krb5_c_encrypt_length
krb5_c_encrypt_type_compare
krb5_c_free_state
krb5_c_init_state
krb5_c_is_coll_proof_cksum
krb5_c_is_keyed_cksum
krb5_c_keyed_checksum_types
krb5_c_make_checksum
krb5_c_make_random_key
krb5_c_random_make_octets
krb5_c_string_to_key
krb5_c_string_to_key_with_params
krb5_c_valid_cksumtype
krb5_c_valid_enctype
krb5_c_verify_checksum
krb5_cc_close
krb5_cc_copy_creds
krb5_cc_default
krb5_cc_default_name
krb5_cc_destroy
krb5_cc_end_seq_get
krb5_cc_gen_new
krb5_cc_get_name
krb5_cc_get_principal
krb5_cc_get_type
krb5_cc_initialize
krb5_cc_next_cred
krb5_cc_remove_cred
krb5_cc_resolve
krb5_cc_retrieve_cred
krb5_cc_set_default_name
krb5_cc_set_flags
krb5_cc_start_seq_get
krb5_cc_store_cred
krb5_change_password
krb5_cksumtype_to_string
krb5_copy_addresses
krb5_copy_authdata
krb5_copy_authenticator
krb5_copy_checksum
krb5_copy_creds
krb5_copy_data
krb5_copy_keyblock
krb5_copy_keyblock_contents
krb5_copy_principal
krb5_copy_ticket
krb5_decode_ticket
krb5_deltat_to_string
krb5_enctype_to_string
krb5_free_addresses
krb5_free_ap_rep_enc_part
krb5_free_authdata
krb5_free_authenticator
krb5_free_checksum
krb5_free_checksum_contents
krb5_free_cksumtypes
krb5_free_context
krb5_free_cred_contents
krb5_free_creds
krb5_free_data
krb5_free_data_contents
krb5_free_default_realm
krb5_free_error
krb5_free_host_realm
krb5_free_keyblock
krb5_free_keyblock_contents
krb5_free_keytab_entry_contents
krb5_free_principal
krb5_free.realm_string
krb5_free.tgt_creds
krb5_free.ticket
krb5_free.unparsed_name
krb5_fwd.tgt_creds
krb5_get.credentials
krb5_get.credentials_renew
krb5_get.credentials_validate
krb5_get.default_realm
krb5_get.host_realm
krb5_get.init.creds_keytab
krb5_get.init.creds_opt_init
krb5_get.init.creds_opt_set_address_list
krb5_get.init.creds_opt_set_etype_list
krb5_get.init.creds_opt_set_forwardable
krb5_get.init.creds_opt_set_preauth_list
krb5_get.init.creds_opt_set_proxiable
krb5_get.init.creds_opt_set_renew_life
krb5_get.init.creds_opt_set_salt
krb5_get.init.creds_opt_set_tkt_life
krb5_get.key_data
krb5_get.key_enctype
krb5_get.key_length
krb5_get.init.creds_password
krb5_get.permitted_enctypes
krb5_get.profile
krb5_get.prompt_types
krb5_get.renewed_creds
krb5_get.server.rcache
krb5_get.validated_creds
krb5_init.allocated_keyblock
krb5_init_context
krb5_init_keyblock
krb5_init_secure_context
krb5_is_thread_safe
krb5_kt_add_entry
krb5_kt_close
krb5_kt_default
krb5_kt_default_name
krb5_kt_end_seq_get
krb5_kt_get_entry
krb5_kt_get_name
krb5_kt_get_type
krb5_kt_next_entry
krb5_kt_read_service_key
krb5_kt_remove_entry
krb5_kt_resolve
krb5_kt_start_seq_get
krb5_kuserok
krb5_mk_error
krb5_mk_ncred
krb5_mk_priv
krb5_mk_rep
krb5_mk_req
krb5_mk_req_extended
krb5_mk_safe
krb5_mk_1cred
krb5_os_localaddr
krb5_parse_name
krb5_principal_compare
krb5_principal2salt
krb5_prompter_posix
krb5_rd_credential
krb5_rd_error
krb5_rd_priv
krb5_rd_rep
krb5_rd_req
krb5_rd_safe
krb5_read_password
krb5_realm_compare
krb5_realm_iterator
krb5_realm_iterator_create
krb5_realm_iterator_free
krb5_recvauth
krb5_recvauth_version
Files

*Files* /usr/lib/libkrb5.so.1 shared object
/usr/lib/64/libkrb5.so.1 64-bit shared object

Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWkrbu (32-bit and 64-bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>External</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>
See Also  krb5-config(1), libgss(3LIB), attributes(5)
libkstat(3LIB)

Name libkstat – kernel statistics library

Synopsis cc [flag...] file... -lkstat [library...] 
#include <kstat.h>

Description Functions in this library provide a general-purpose mechanism for providing kernel statistics to users.

Interfaces The shared object libkstat.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

kstat_chain_update kstat_close
kstat_data_lookup kstat_lookup
kstat_open kstat_read
kstat_write

Files /lib/libkstat.so.1 shared object
/lib/64/libkstat.so.1 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Stable</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

SeeAlso 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.

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

```
kvm_close  kvm_getcmd
kvm_getproc  kvm_getu
kvm_kread  kvm_kwrite
kvm_nextproc  kvm_nlist
kvm_open  kvm_read
kvm_setproc  kvm_uread
kvm_urewrite  kvm_write
```

**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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td><code>kvm_read()</code> and <code>kvm_write()</code> are Obsolete; the remaining functions are Stable.</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

**See Also**  `pvs(1), Intro(3), attributes(5)`
libl(3LIB)

Name  libl – lex library

Synopsis  cc [ flag... ] file... [ library... ]

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.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>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWctpls</td>
</tr>
<tr>
<td>MT-Level</td>
<td>MT–Safe</td>
</tr>
</tbody>
</table>

See Also  Intro(3), attributes(5)
liblgrp (3LIB)

Name  liblgrp – locality group library

Synopsis  cc [ flag... ] file... -llgrp [ library... ]
          #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.

  lgrp_affinity_get lgrp_affinity_set
  lgrp_children lgrp_cookie_stale
  lgrp_cpus lgrp_fini
  lgrp_home lgrp_init
  lgrp_latency lgrp_latency_cookie
  lgrp_mem_size lgrp_nlgrps
  lgrp_parents lgrp_resourcess
  lgrp_root lgrp_version
  lgrp_view

Files  /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>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>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>

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_resourcess(3LGRP), lgrp_root(3LGRP), lgrp_version(3LGRP), lgrp_view(3LGRP), attributes(5)
Name   libm – C math library

Synopsis  c99 [ flag... ] file... -lm [ library... ]

Description  Functions in this library provide common elementary mathematical functions and floating point environment routines defined by System V, ANSI C, POSIX, and so on. See standards(5). Additional functions in this library provide extended support for handling floating point exceptions.

Interfaces  The shared object libm.so.2 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

acos  acosf
acosh  acoshf
acoshl  acosl
asin  asinf
asinh  asinhf
asinhl  asinl
atan  atan2
atan2f  atan2l
atanf  atanhl
atanhf  atanl
atanl  cabs
cabsf  cabsl
cacos  cacosf
cacosh  cacoshf
cacoshl  cacosl
carg  cargf
cargl  casin
casinf  casinh
casinhf  casinhl
casinl  catan
catanf  catanh
catanhf  catanhl
<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>catanl</td>
</tr>
<tr>
<td>cbtf</td>
</tr>
<tr>
<td>ccos</td>
</tr>
<tr>
<td>ccosh</td>
</tr>
<tr>
<td>ccoshl</td>
</tr>
<tr>
<td>ceil</td>
</tr>
<tr>
<td>ceilf</td>
</tr>
<tr>
<td>ceill</td>
</tr>
<tr>
<td>cexp</td>
</tr>
<tr>
<td>cexpf</td>
</tr>
<tr>
<td>cimag</td>
</tr>
<tr>
<td>cimagl</td>
</tr>
<tr>
<td>clog</td>
</tr>
<tr>
<td>clogf</td>
</tr>
<tr>
<td>conj</td>
</tr>
<tr>
<td>conjf</td>
</tr>
<tr>
<td>conjl</td>
</tr>
<tr>
<td>copy sign</td>
</tr>
<tr>
<td>copy signf</td>
</tr>
<tr>
<td>cos</td>
</tr>
<tr>
<td>cosf</td>
</tr>
<tr>
<td>cosh</td>
</tr>
<tr>
<td>coshf</td>
</tr>
<tr>
<td>coshl</td>
</tr>
<tr>
<td>cosl</td>
</tr>
<tr>
<td>cpow</td>
</tr>
<tr>
<td>cpowf</td>
</tr>
<tr>
<td>cpowl</td>
</tr>
<tr>
<td>cproj</td>
</tr>
<tr>
<td>cprojf</td>
</tr>
<tr>
<td>cprojl</td>
</tr>
<tr>
<td>creal</td>
</tr>
<tr>
<td>crealf</td>
</tr>
<tr>
<td>creall</td>
</tr>
<tr>
<td>csin</td>
</tr>
<tr>
<td>csinf</td>
</tr>
<tr>
<td>csinh</td>
</tr>
<tr>
<td>csinhf</td>
</tr>
<tr>
<td>csinl</td>
</tr>
<tr>
<td>csqrt</td>
</tr>
<tr>
<td>csqrf</td>
</tr>
<tr>
<td>csqrl</td>
</tr>
<tr>
<td>ctan</td>
</tr>
<tr>
<td>ctanf</td>
</tr>
<tr>
<td>ctanh</td>
</tr>
<tr>
<td>ctanhf</td>
</tr>
<tr>
<td>ctanhl</td>
</tr>
<tr>
<td>ctanl</td>
</tr>
</tbody>
</table>
erf
erfc
erfcf
erfcl
erff
erfl
exp
exp2
exp2f
exp2l
expf
expl
expm1
expm1f
expm1l
fabs
fabsf
fabsl
fdim
fdimf
fdiml
feclearexcept
fegetenv
fegetexceptflag
fegetround
feraiseexcept
fesetenv
fesetexceptflag
fesetround
fetestexcept
feupdateenv
fex_get_handling
fex_get_log
fex_get_log_depth
fex_getexcepthandler
fex_log_entry
fex_merge_flags
fex_set_handling
fex_set_log
fex_set_log_depth
fex_setexcepthandler
floor
floorf
floorl
fma
fmaf
fmax
fmaxf
fmaxl
fmin
fminf
fminl
fmod
fmodf
fmodl
frexp
frexp  frexpl
gamma  gamma_r
gammaf  gammaf_r
gammad  gammal_r
hypot  hypotf
hypotl  ilogb
ilogbf  ilogbl
isnan  j0
j0f  j0l
j1  j1f
j1l  jn
jnf  jnl
ldexp  ldexpf
ldexpl  lgamma
lgamma_r  lgammaf
lgammaf_r  lgammal
lgammal_r  llrint
llrintf  llrintl
llround  llroundf
llroundl  log
log10  log10f
log10l  log1p
log1pf  log1pl
log2  log2f
log2l  logb
logbf  logbl
logf  logl
lrint  lrintf
lrintl  lround
<table>
<thead>
<tr>
<th>Function</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>lroundf</td>
<td>lroundl</td>
</tr>
<tr>
<td>matherr</td>
<td>modf</td>
</tr>
<tr>
<td>modff</td>
<td>modfl</td>
</tr>
<tr>
<td>nan</td>
<td>nanf</td>
</tr>
<tr>
<td>nanl</td>
<td>nearbyint</td>
</tr>
<tr>
<td>nearbyintf</td>
<td>nearbyintl</td>
</tr>
<tr>
<td>nextafter</td>
<td>nextafterf</td>
</tr>
<tr>
<td>nextafterl</td>
<td>nexttoward</td>
</tr>
<tr>
<td>nexttowardf</td>
<td>nexttowardl</td>
</tr>
<tr>
<td>pow</td>
<td>powf</td>
</tr>
<tr>
<td>powl</td>
<td>remainder</td>
</tr>
<tr>
<td>remainderf</td>
<td>remainderl</td>
</tr>
<tr>
<td>remquo</td>
<td>remquof</td>
</tr>
<tr>
<td>remquol</td>
<td>rint</td>
</tr>
<tr>
<td>rintf</td>
<td>rintl</td>
</tr>
<tr>
<td>round</td>
<td>roundf</td>
</tr>
<tr>
<td>roundl</td>
<td>scalb</td>
</tr>
<tr>
<td>scalbf</td>
<td>scalbl</td>
</tr>
<tr>
<td>scalbln</td>
<td>scalblnf</td>
</tr>
<tr>
<td>scalblnl</td>
<td>scalbn</td>
</tr>
<tr>
<td>scalbnf</td>
<td>scalbnl</td>
</tr>
<tr>
<td>signgam</td>
<td>signgamf</td>
</tr>
<tr>
<td>signgaml</td>
<td>significand</td>
</tr>
<tr>
<td>significandf</td>
<td>significandl</td>
</tr>
<tr>
<td>sin</td>
<td>sincos</td>
</tr>
<tr>
<td>sincosf</td>
<td>sincosl</td>
</tr>
<tr>
<td>sinf</td>
<td>sinh</td>
</tr>
<tr>
<td>sinhf</td>
<td>sinhl</td>
</tr>
<tr>
<td>sinl</td>
<td>sqrt</td>
</tr>
</tbody>
</table>
The following interfaces are unique to the x86 and x64 versions of this library:

fegetprec  fesetprec

**Accuracy**  ISO/IEC 9899:1999, also known as C99, specifies the functions listed in the following tables and states that the accuracy of these functions is “implementation-defined”. The information below characterizes the accuracy of these functions as implemented in `libm.so.2`. For each function, the tables provide an upper bound on the largest error possible for any argument and the largest error actually observed among a large sample of arguments. Errors are expressed in “units in the last place”, or ulps, relative to the exact function value for each argument (regarding the argument as exact). Ulps depend on the precision of the floating point format: if \( y \) is the exact function value, \( x \) and \( x' \) are adjacent floating point numbers such that \( x < y < x' \), and \( x'' \) is the computed function value, then provided \( x, x' \), and \( x'' \) all lie in the same binade, the error in \( x'' \) is \( |y - x''| / |x - x'| \) ulps. In particular, when the error is less than one ulp, the computed value is one of the two floating point numbers adjacent to the exact value.

The bounds and observed errors listed below apply only in the default floating point modes. Specifically, on SPARC, these bounds assume the rounding direction is round-to-nearest and non-standard mode is disabled. On x86, the bounds assume the rounding direction is round-to-nearest and the rounding precision is round-to-64-bits. Moreover, on x86, floating point function values are returned in a floating point register in extended double precision format, but the bounds below assume that the result value is then stored to memory in the format corresponding to the function’s type. On x64, the bounds assume the rounding
direction in both the x87 floating point control word and the MXCSR is round-to-nearest, the rounding precision in the x87 control word is round-to-64-bits, and the FTZ and DAZ modes are disabled.

The error bounds listed below are believed to be correct, but smaller bounds might be proved later. The observed errors are the largest ones currently known, but larger errors might be discovered later. Numbers in the notes column refer to the notes following the tables.

### Real Functions

**Single precision real functions (SPARC, x86, and x64)**

<table>
<thead>
<tr>
<th>function</th>
<th>error bound</th>
<th>largest error</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>acosf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>acoshf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>asinf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>asinhf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>atanf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>atan2f</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>atanhf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>cbrtf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>cosf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>coshf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>erff</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>erfcf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>expf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>exp2f</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>expmf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>hypotf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>lgammaf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>logf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>log10f</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>log1pf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>function</td>
<td>error bound (ulps)</td>
<td>largest error observed (ulps)</td>
<td>notes</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------</td>
<td>-------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>log2f</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>powf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>sinf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>sinhf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>sqrtf</td>
<td>0.5</td>
<td>0.500</td>
<td>[1]</td>
</tr>
<tr>
<td>tanf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>tanhf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>tgammaf</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
</tbody>
</table>

Double precision real functions (SPARC and x64)

<table>
<thead>
<tr>
<th>function</th>
<th>error bound (ulps)</th>
<th>largest error observed (ulps)</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>acos</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>acosh</td>
<td>4.0</td>
<td>1.878</td>
<td></td>
</tr>
<tr>
<td>asin</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>asinh</td>
<td>7.0</td>
<td>1.653</td>
<td></td>
</tr>
<tr>
<td>atan</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>atanh</td>
<td>2.5</td>
<td>1.475</td>
<td></td>
</tr>
<tr>
<td>cbrt</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>cos</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>cosh</td>
<td>3.0</td>
<td>1.168</td>
<td></td>
</tr>
<tr>
<td>erf</td>
<td>4.0</td>
<td>0.959</td>
<td></td>
</tr>
<tr>
<td>erfc</td>
<td>6.0</td>
<td>2.816</td>
<td></td>
</tr>
<tr>
<td>exp</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>exp2</td>
<td>2.0</td>
<td>1.050</td>
<td></td>
</tr>
<tr>
<td>expm1</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>function</td>
<td>error bound (ulps)</td>
<td>largest error observed (ulps)</td>
<td>notes</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------</td>
<td>-------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>hypot</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>lgamma</td>
<td>61.5</td>
<td>5.629</td>
<td>[2]</td>
</tr>
<tr>
<td>log</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>log10</td>
<td>3.5</td>
<td>1.592</td>
<td></td>
</tr>
<tr>
<td>log1p</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>log2</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>pow</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>sin</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>sinh</td>
<td>4.0</td>
<td>2.078</td>
<td></td>
</tr>
<tr>
<td>sqrt</td>
<td>0.5</td>
<td>0.500</td>
<td>[1]</td>
</tr>
<tr>
<td>tan</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>tanh</td>
<td>3.5</td>
<td>2.136</td>
<td></td>
</tr>
<tr>
<td>tgamma</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
</tbody>
</table>

Double precision real functions (x86)

<table>
<thead>
<tr>
<th>function</th>
<th>error bound (ulps)</th>
<th>largest error observed (ulps)</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>acos</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>acosh</td>
<td>4.0</td>
<td>1.694</td>
<td></td>
</tr>
<tr>
<td>asin</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>asinh</td>
<td>7.0</td>
<td>1.493</td>
<td></td>
</tr>
<tr>
<td>atan</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>atan2</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>atanh</td>
<td>4.0</td>
<td>1.445</td>
<td></td>
</tr>
<tr>
<td>cbrc</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>cos</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>cosh</td>
<td>3.0</td>
<td>1.001</td>
<td></td>
</tr>
</tbody>
</table>
### Quadruple precision real functions (SPARC)

<table>
<thead>
<tr>
<th>function</th>
<th>error bound (ulps)</th>
<th>largest error observed (ulps)</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>acosl</td>
<td>3.5</td>
<td>1.771</td>
<td></td>
</tr>
<tr>
<td>acoshl</td>
<td>8.0</td>
<td>1.275</td>
<td></td>
</tr>
<tr>
<td>asinl</td>
<td>4.0</td>
<td>2.007</td>
<td></td>
</tr>
<tr>
<td>asinhl</td>
<td>9.0</td>
<td>1.823</td>
<td></td>
</tr>
<tr>
<td>atanl</td>
<td>1.0</td>
<td>&lt; 1</td>
<td></td>
</tr>
<tr>
<td>function</td>
<td>error bound (ulps)</td>
<td>largest error observed (ulps)</td>
<td>notes</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------</td>
<td>-------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>atan2l</td>
<td>2.5</td>
<td>1.102</td>
<td></td>
</tr>
<tr>
<td>atanh1</td>
<td>4.0</td>
<td>1.970</td>
<td></td>
</tr>
<tr>
<td>cbrtl</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>cosl</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>coshl</td>
<td>3.5</td>
<td>0.985</td>
<td></td>
</tr>
<tr>
<td>erf1</td>
<td>2.0</td>
<td>0.779</td>
<td></td>
</tr>
<tr>
<td>erfcl</td>
<td>68.5</td>
<td>13.923</td>
<td></td>
</tr>
<tr>
<td>exp1</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>exp2l</td>
<td>2.0</td>
<td>0.714</td>
<td></td>
</tr>
<tr>
<td>expm1l</td>
<td>2.0</td>
<td>1.020</td>
<td></td>
</tr>
<tr>
<td>hypotl</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>lgamml</td>
<td>18.5</td>
<td>2.916</td>
<td>[2]</td>
</tr>
<tr>
<td>log1</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>log10l</td>
<td>3.5</td>
<td>1.156</td>
<td></td>
</tr>
<tr>
<td>log1pl</td>
<td>2.0</td>
<td>1.216</td>
<td></td>
</tr>
<tr>
<td>log2l</td>
<td>3.5</td>
<td>1.675</td>
<td></td>
</tr>
<tr>
<td>powl</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>sinl</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
<tr>
<td>sinh1</td>
<td>4.5</td>
<td>1.589</td>
<td></td>
</tr>
<tr>
<td>sqrt1</td>
<td>0.5</td>
<td>0.500</td>
<td>[1]</td>
</tr>
<tr>
<td>tanl</td>
<td>4.5</td>
<td>2.380</td>
<td></td>
</tr>
<tr>
<td>tanhl</td>
<td>4.5</td>
<td>1.692</td>
<td></td>
</tr>
<tr>
<td>tgammal</td>
<td>1.0</td>
<td>&lt;1</td>
<td></td>
</tr>
</tbody>
</table>

Extended precision real functions (x86 and x64)
<table>
<thead>
<tr>
<th>function</th>
<th>error bound (ulps)</th>
<th>largest error observed (ulps)</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>acosl</td>
<td>3.0</td>
<td>1.868</td>
<td></td>
</tr>
<tr>
<td>acoshl</td>
<td>8.0</td>
<td>2.352</td>
<td></td>
</tr>
<tr>
<td>asinl</td>
<td>3.0</td>
<td>1.716</td>
<td></td>
</tr>
<tr>
<td>asinhl</td>
<td>9.0</td>
<td>2.346</td>
<td></td>
</tr>
<tr>
<td>atanl</td>
<td>1.0</td>
<td>&lt; 1</td>
<td></td>
</tr>
<tr>
<td>atan2l</td>
<td>1.0</td>
<td>&lt; 1</td>
<td></td>
</tr>
<tr>
<td>atanhl</td>
<td>4.0</td>
<td>2.438</td>
<td></td>
</tr>
<tr>
<td>cbrtl</td>
<td>1.0</td>
<td>&lt; 1</td>
<td></td>
</tr>
<tr>
<td>cosl</td>
<td>1.0</td>
<td>&lt; 1</td>
<td></td>
</tr>
<tr>
<td>coshl</td>
<td>3.5</td>
<td>1.288</td>
<td></td>
</tr>
<tr>
<td>erfl</td>
<td>1.0</td>
<td>&lt; 1</td>
<td></td>
</tr>
<tr>
<td>erfcl</td>
<td>78.5</td>
<td>13.407</td>
<td></td>
</tr>
<tr>
<td>expl</td>
<td>3.5</td>
<td>1.291</td>
<td></td>
</tr>
<tr>
<td>exp2l</td>
<td>1.5</td>
<td>0.807</td>
<td></td>
</tr>
<tr>
<td>expm1l</td>
<td>4.0</td>
<td>1.936</td>
<td></td>
</tr>
<tr>
<td>hypotl</td>
<td>3.5</td>
<td>2.087</td>
<td></td>
</tr>
<tr>
<td>lgamma1</td>
<td>22.5</td>
<td>4.197</td>
<td>[2]</td>
</tr>
<tr>
<td>logl</td>
<td>2.0</td>
<td>0.881</td>
<td></td>
</tr>
<tr>
<td>log10l</td>
<td>2.0</td>
<td>1.284</td>
<td></td>
</tr>
<tr>
<td>log1pl</td>
<td>5.0</td>
<td>2.370</td>
<td></td>
</tr>
<tr>
<td>log2l</td>
<td>1.0</td>
<td>&lt; 1</td>
<td></td>
</tr>
<tr>
<td>powl</td>
<td>32770.0</td>
<td>4478.132</td>
<td></td>
</tr>
<tr>
<td>sinl</td>
<td>1.0</td>
<td>&lt; 1</td>
<td></td>
</tr>
<tr>
<td>sinh1</td>
<td>4.5</td>
<td>2.356</td>
<td></td>
</tr>
<tr>
<td>sqrt1</td>
<td>0.5</td>
<td>0.500</td>
<td>[1]</td>
</tr>
<tr>
<td>tanl</td>
<td>4.5</td>
<td>2.366</td>
<td></td>
</tr>
<tr>
<td>tanhl</td>
<td>4.5</td>
<td>2.417</td>
<td></td>
</tr>
</tbody>
</table>
function | error bound (ulps) | largest error observed (ulps) | notes
--- | --- | --- | ---
tgammal | 1.0 | <1 | 

Notes:

[1] On SPARC and x64, sqrtf, sqrt, and sqrtl are correctly rounded in accordance with IEEE 754. On x86, sqrtl is correctly rounded, sqrtf is correctly rounded provided the result is narrowed to single precision as discussed above, but sqrt might not be correctly rounded due to “double rounding”: when the intermediate value computed to extended precision lies exactly halfway between two representable numbers in double precision, the result of rounding the intermediate value to double precision is determined by the round-ties-to-even rule. If this rule causes the second rounding to round in the same direction as the first, the net rounding error can exceed 0.5 ulps. (The error is bounded instead by 0.5*(1 + 2^-11) ulps.)

[2] Error bounds for lgamma and lgammal apply only for positive arguments.

Complex functions

The real-valued complex functions cabs f, cabs, cabs l, carg f, carg, and carg l are equivalent to the real functions hypot f, hypot, hypot l, atan2 f, atan2, and atan2 l, respectively. The error bounds and observed errors given above for the latter functions also apply to the former.

The complex functions listed below are complex-valued. For each function, the error bound shown applies separately to both the real and imaginary parts of the result. (For example, both the real and imaginary parts of ccos f(z) are accurate to within 1 ulp regardless of their magnitudes.) Similarly, the largest observed error shown is the largest error found in either the real or the imaginary part of the result.

Single precision complex functions (SPARC and x64)

<table>
<thead>
<tr>
<th>function</th>
<th>error bound (ulps)</th>
<th>largest error observed (ulps)</th>
<th>notes</th>
</tr>
</thead>
</table>
cacosf, cacoshf | 1 | <1 | [1] |
casinf, casinhf | 1 | <1 | |
catanf, catanhf | 6 | <1 | |
cosf, ccoshf | 10 | 2.012 | |
cexpf | 3 | 2.239 | |
clogf | 3 | <1 | |
cpolf | — | <1 | [2] |
<table>
<thead>
<tr>
<th>function</th>
<th>error bound (ulps)</th>
<th>largest error observed (ulps)</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>csinf, csinhf</td>
<td>10</td>
<td>2.009</td>
<td></td>
</tr>
<tr>
<td>csqrtf</td>
<td>4</td>
<td>&lt; 1</td>
<td></td>
</tr>
<tr>
<td>ctanf, ctanhf</td>
<td>13</td>
<td>6.987</td>
<td></td>
</tr>
</tbody>
</table>

**Single precision complex functions (x86)**

<table>
<thead>
<tr>
<th>function</th>
<th>error bound (ulps)</th>
<th>largest error observed (ulps)</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacosf, cacoshf</td>
<td>1</td>
<td>&lt; 1</td>
<td>[1]</td>
</tr>
<tr>
<td>casinf, casinhf</td>
<td>1</td>
<td>&lt; 1</td>
<td></td>
</tr>
<tr>
<td>catanf, catanhf</td>
<td>6</td>
<td>&lt; 1</td>
<td></td>
</tr>
<tr>
<td>ccosf, ccoshf</td>
<td>10</td>
<td>1.984</td>
<td></td>
</tr>
<tr>
<td>cexpf</td>
<td>3</td>
<td>1.984</td>
<td></td>
</tr>
<tr>
<td>clogf</td>
<td>3</td>
<td>&lt; 1</td>
<td></td>
</tr>
<tr>
<td>cpowf</td>
<td>—</td>
<td>&lt; 1</td>
<td>[2]</td>
</tr>
<tr>
<td>csinf, csinhf</td>
<td>10</td>
<td>1.973</td>
<td></td>
</tr>
<tr>
<td>csqrtf</td>
<td>4</td>
<td>&lt; 1</td>
<td></td>
</tr>
<tr>
<td>ctanf, ctanhf</td>
<td>13</td>
<td>4.657</td>
<td></td>
</tr>
</tbody>
</table>

**Double precision complex functions (SPARC and x64)**

<table>
<thead>
<tr>
<th>function</th>
<th>error bound (ulps)</th>
<th>largest error observed (ulps)</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacos, cacosh</td>
<td>9</td>
<td>3.831</td>
<td>[1]</td>
</tr>
<tr>
<td>casin, casinh</td>
<td>9</td>
<td>3.732</td>
<td></td>
</tr>
<tr>
<td>catan, catanh</td>
<td>6</td>
<td>4.179</td>
<td></td>
</tr>
<tr>
<td>ccos, ccosh</td>
<td>10</td>
<td>3.832</td>
<td></td>
</tr>
<tr>
<td>cexp</td>
<td>3</td>
<td>2.255</td>
<td></td>
</tr>
<tr>
<td>clog</td>
<td>3</td>
<td>2.870</td>
<td></td>
</tr>
<tr>
<td>function</td>
<td>error bound (ulps)</td>
<td>largest error observed (ulps)</td>
<td>notes</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------</td>
<td>-------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>cpow</td>
<td>-</td>
<td>-</td>
<td>[2]</td>
</tr>
<tr>
<td>csin, csinh</td>
<td>10</td>
<td>3.722</td>
<td></td>
</tr>
<tr>
<td>csqrt</td>
<td>4</td>
<td>3.204</td>
<td></td>
</tr>
<tr>
<td>ctan, ctanh</td>
<td>13</td>
<td>7.143</td>
<td></td>
</tr>
</tbody>
</table>

### Double precision complex functions (x86)

<table>
<thead>
<tr>
<th>function</th>
<th>error bound (ulps)</th>
<th>largest error observed (ulps)</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacos, cacosh</td>
<td>9</td>
<td>3.624</td>
<td>[1]</td>
</tr>
<tr>
<td>casin, casinh</td>
<td>9</td>
<td>3.624</td>
<td></td>
</tr>
<tr>
<td>ccatanh, ccatanh</td>
<td>6</td>
<td>2.500</td>
<td></td>
</tr>
<tr>
<td>ccos, ccosh</td>
<td>10</td>
<td>2.929</td>
<td></td>
</tr>
<tr>
<td>cexp</td>
<td>3</td>
<td>2.147</td>
<td></td>
</tr>
<tr>
<td>clog</td>
<td>3</td>
<td>1.927</td>
<td></td>
</tr>
<tr>
<td>cpow</td>
<td>-</td>
<td>-</td>
<td>[2]</td>
</tr>
<tr>
<td>csin, csinh</td>
<td>10</td>
<td>2.918</td>
<td></td>
</tr>
<tr>
<td>csqrt</td>
<td>4</td>
<td>1.914</td>
<td></td>
</tr>
<tr>
<td>ctan, ctanh</td>
<td>13</td>
<td>4.630</td>
<td></td>
</tr>
</tbody>
</table>

### Quadruple precision complex functions (SPARC)

<table>
<thead>
<tr>
<th>function</th>
<th>error bound (ulps)</th>
<th>largest error observed (ulps)</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacosl, cacoshl</td>
<td>9</td>
<td>3</td>
<td>[1]</td>
</tr>
<tr>
<td>casinl, casinhl</td>
<td>9</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>ccatanhl, ccatanhl</td>
<td>6</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>ccosl, ccoshl</td>
<td>10</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>cexp1</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
</tbody>
</table>
## Extended precision complex functions (x86 and x64)

<table>
<thead>
<tr>
<th>function</th>
<th>error bound (ulps)</th>
<th>largest error observed (ulps)</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>cacosl, cacoshl</td>
<td>9</td>
<td>2</td>
<td>[1]</td>
</tr>
<tr>
<td>casinl, casinhl</td>
<td>9</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>catanl, catanhl</td>
<td>6</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>ccosl, ccoshl</td>
<td>10</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>cexpl</td>
<td>3</td>
<td>2.699</td>
<td></td>
</tr>
<tr>
<td>clogl</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>cpowl</td>
<td>-</td>
<td>-</td>
<td>[2]</td>
</tr>
<tr>
<td>csinl, csinhl</td>
<td>10</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>csqrtl</td>
<td>4</td>
<td>1.452</td>
<td></td>
</tr>
<tr>
<td>ctanl, ctanhl</td>
<td>13</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

### Notes:

1. The complex hyperbolic trigonometric functions are equivalent by symmetries to their circular trigonometric counterparts. Because the implementations of these functions exploit these symmetries, corresponding functions have the same error bounds and observed errors.

2. For large arguments, the results computed by `cpowf`, `cpow`, and `cpowl` can have unbounded relative error. It might be possible to give error bounds for specific domains, but no such bounds are currently available. The observed errors shown are for the domain \( \{(z, w) : \max(|\Re z|, |\Im z|, |\Re w|, |\Im w|) \leq 1 \} \).
### Files

<table>
<thead>
<tr>
<th>Files</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/lib/libm.so.2</td>
<td>shared object</td>
</tr>
<tr>
<td>/lib/64/libm.so.2</td>
<td>64-bit shared object</td>
</tr>
</tbody>
</table>

### Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWlibmsr</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe with exceptions</td>
</tr>
</tbody>
</table>

As described on the lgamma(3M) manual page, gamma() and lgamma() and their float and long double counterparts are Unsafe. All other functions in libm.so.2 are MT-Safe.

### See Also

Intro(3), lgamma(3M), math.h(3HEAD), attributes(5), standards(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.so.1 shared object
/usr/lib/64/libmail.so.1 64–bit shared object
```

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsvl (32–bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcsvl (64–bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

**See Also** `Intro(3), maillock(3MAIL), attributes(5)`
Name  libmalloc – memory allocation library

Synopsis  cc [ flag... ] file... -lmalloc [ library... ]

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 libmalloc.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

    calloc  cfree
    free   mallinfo
    malloc mallopt
    realloc valloc

Files  /usr/lib/libmalloc.so.1 shared object
       /usr/lib/64/libmalloc.so.1 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also  Intro(3), malloc(3MALLOC), attributes(5)
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          malloc
memalign        realloc
valloc

Files

/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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also

pvs(1), mmap(2), sbrk(2), Intro(3), malloc(3C), malloc(3MALLOCC), mapmalloc(3MALLOCC), attributes(5)
libmd – Message Digest library

Synopsis: cc [ flag... ] file... -lmd [ library... ]
#include <md4.h>
#include <md5.h>
#include <sha1.h>
#include <sha2.h>

Description: Functions in this library provide hashing routines for MD4 (RFC1320), MD5 (RFC1321), SHA1 (RFC3174), SHA256 (FIPS 180-2), SHA384 (FIPS 180-2), SHA512 (FIPS 180-2).

Interfaces: The shared object /lib/libmd.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

MD4Final
MD4Init
MD4Update
md5_calc
MD5Final
MD5Init
MD5Update
SHA1Final
SHA1Init
SHA1Update
SHA2Final
SHA2Init
SHA2Update

Security: The MD4 and MD5 algorithms are currently considered weak for cryptographic use. The algorithms should be used only for compatibility with legacy systems or protocols.

The SHA1 algorithm is also believed to have some weaknesses. Migration to one of the SHA2 algorithms—including SHA256, SHA386 or SHA512— is highly recommended when compatibility with data formats and on wire protocols is permitted.

Files: /lib/libmd.so.1 shared object
/lib/64/libmd.so.1 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Committed</td>
</tr>
<tr>
<td>MT-Level</td>
<td>MT-Safe</td>
</tr>
</tbody>
</table>
**Name**  libmd5 – MD5 hashing library

**Synopsis**  cc [ flag... ] file... -lmd5 [ library... ]  
#include <md5.h>

**Description**  Functions in this library provide MD5 hashing routines.

**Interfaces**  The shared object libmd5.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

```
MD5Final
MD5Init
MD5Update
 md5_calc
```

**Files**  /lib/libmd5.so.1 shared object  
/lib/64/libmd5.so.1  64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>MT-Safe</td>
</tr>
</tbody>
</table>

**See Also**  Intro(3), attributes(5)
Name  libmenu – menus library

Synopsis  cc [ flag... ] file... -lmenu [ library... ]

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
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
                unpost_menu

Files  /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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

See Also  Intro(3), libcurses(3LIB), attributes(5)
### libmlib

**Name**  
libmlib – mediaLib library

**Synopsis**  
`cc [ flag... ] file... -lmlib [ library... ]`

```c
#include <mlib.h>
```

**Description**  
Interfaces in this library provide functions for multimedia processing. When executed on an UltraSPARC platform, these functions take advantage of the VIS Instruction Set. When executed on an AMD64 platform, these functions take advantage of the MMX/SSE/SSE2 instructions.

**Interfaces**  
The shared object `libmlib.so.2` provides the public interfaces defined below. See `Intro(3)` for additional information on shared object interfaces.

#### System Functions
- `mlib_free`
- `mlib_malloc`
- `mlib_memcpy`
- `mlib_memmove`
- `mlib_memset`
- `mlib_realloc`
- `mlib_version`

#### Algebra Functions
- `mlib_MatrixAdd_S16C_Mod`
- `mlib_MatrixAdd_S16C_S16C_Mod`
- `mlib_MatrixAdd_S16C_S16C_Sat`
- `mlib_MatrixAdd_S16C_S8C_Mod`
- `mlib_MatrixAdd_S16C_S8C_Sat`
- `mlib_MatrixAdd_S16C_Sat`
- `mlib_MatrixAdd_S16C_U8C_Mod`
- `mlib_MatrixAdd_S16C_U8C_Sat`
- `mlib_MatrixAdd_S16_Mod`
- `mlib_MatrixAdd_S16_S16_Mod`
- `mlib_MatrixAdd_S16_S16_Sat`
- `mlib_MatrixAdd_S16_S8_Mod`
- `mlib_MatrixAdd_S16_S8_Sat`
- `mlib_MatrixAdd_S16_Sat`
- `mlib_MatrixAdd_S16_U8_Mod`
- `mlib_MatrixAdd_S16_U8_Sat`
- `mlib_MatrixAdd_S32C_Mod`
- `mlib_MatrixAdd_S32C_S16C_Mod`
- `mlib_MatrixAdd_S32C_S16C_Sat`
- `mlib_MatrixAdd_S32C_S32C_Mod`
- `mlib_MatrixAdd_S32C_S32C_Sat`
- `mlib_MatrixAdd_S32C_Sat`
- `mlib_MatrixAdd_S32C_S32C_Mod`
- `mlib_MatrixAdd_S32_S16_Mod`
- `mlib_MatrixAdd_S32_S16_Sat`
- `mlib_MatrixAdd_S32_S32_Mod`
- `mlib_MatrixAdd_S32_S32_Sat`
libmlib(3LIB)

- mlib_MatrixAdd_S32_S32_Sat
- mlib_MatrixAdd_S32_Sat
- mlib_MatrixAdd_S8C_Mod
- mlib_MatrixAdd_S8C_S8C_Mod
- mlib_MatrixAdd_S8C_S8C_Sat
- mlib_MatrixAdd_S8C_Sat
- mlib_MatrixAdd_S8_S8_Mod
- mlib_MatrixAdd_S8_S8_Sat
- mlib_MatrixAdd_S8_Sat
- mlib_MatrixAddS_S16C_Mod
- mlib_MatrixAddS_S16C_S16C_Mod
- mlib_MatrixAddS_S16C_S16C_Sat
- mlib_MatrixAddS_S16C_S8C_Mod
- mlib_MatrixAddS_S16C_S8C_Sat
- mlib_MatrixAddS_S16C_Sat
- mlib_MatrixAddS_S16C_U8C_Mod
- mlib_MatrixAddS_S16C_U8C_Sat
- mlib_MatrixAddS_S16_Mod
- mlib_MatrixAddS_S16_S16_Mod
- mlib_MatrixAddS_S16_S16_Sat
- mlib_MatrixAddS_S16_S8_Mod
- mlib_MatrixAddS_S16_S8_Sat
- mlib_MatrixAddS_S16_Sat
- mlib_MatrixAddS_S16_U8_Mod
- mlib_MatrixAddS_S16_U8_Sat
- mlib_MatrixAddS_S32C_Mod
- mlib_MatrixAddS_S32C_S16C_Mod
- mlib_MatrixAddS_S32C_S16C_Sat
- mlib_MatrixAddS_S32C_S32C_Mod
- mlib_MatrixAddS_S32C_S32C_Sat
- mlib_MatrixAddS_S32C_Sat
- mlib_MatrixAddS_S32_S16_Mod
- mlib_MatrixAddS_S32_S16_Sat
- mlib_MatrixAddS_S32_S32_Mod
- mlib_MatrixAddS_S32_S32_Sat
- mlib_MatrixAddS_S32_Sat
- mlib_MatrixAddS_S8C_Mod
- mlib_MatrixAddS_S8C_S8C_Mod
- mlib_MatrixAddS_S8C_S8C_Sat
- mlib_MatrixAddS_S8C_Sat
- mlib_MatrixAddS_S8_S8_Mod
- mlib_MatrixAddS_S8_S8_Sat
- mlib_MatrixAddS_S8_Sat
- mlib_MatrixAddS_S8_S8_Sat
- mlib_MatrixAddS_S8_Sat
- mlib_MatrixAddS_U8C_Mod
- mlib_MatrixAddS_U8C_Sat
- mlib_MatrixAddS_U8C_U8C_Mod
- mlib_MatrixAddS_U8C_U8C_Sat
- mlib_MatrixAddS_U8_Mod
- mlib_MatrixAddS_U8_Sat
- mlib_MatrixAddS_U8_U8_Mod
- mlib_MatrixAddS_U8_U8_Sat
- mlib_MatrixAdd_U8C_Mod
- mlib_MatrixAdd_U8C_Sat
- mlib_MatrixAdd_U8C_U8C_Mod
- mlib_MatrixAdd_U8C_U8C_Sat
- mlib_MatrixAdd_U8_Mod
- mlib_MatrixAdd_U8_Sat
- mlib_MatrixAdd_U8_U8_Mod
- mlib_MatrixAdd_U8_U8_Sat
- mlib_MatrixAve_S16
- mlib_MatrixAve_S16C
- mlib_MatrixAve_S16C_S16C
- mlib_MatrixAve_S16C_S8C
- mlib_MatrixAve_S16C_U8C
- mlib_MatrixAve_S16_S16
- mlib_MatrixAve_S16_S8
- mlib_MatrixAve_S16_U8
- mlib_MatrixAve_S32
- mlib_MatrixAve_S32C
- mlib_MatrixAve_S32C_S16C
- mlib_MatrixAve_S32C_S32C
- mlib_MatrixAve_S32_S16
- mlib_MatrixAve_S32_S32
- mlib_MatrixAve_S8
- mlib_MatrixAve_S8C
- mlib_MatrixAve_S8C_S8C
- mlib_MatrixAve_S8_S8C
- mlib_MatrixAve_U8
- mlib_MatrixAve_U8C
- mlib_MatrixAve_U8C_U8C
- mlib_MatrixMaximum_D64
- mlib_MatrixMaximum_F32
- mlib_MatrixMaximumMag_D64C
- mlib_MatrixMaximumMag_F32C
libmlib(3LIB)

- mlib_MatrixMaximumMag_S16C
- mlib_MatrixMaximumMag_S32C
- mlib_MatrixMaximumMag_S8C
- mlib_MatrixMaximumMag_U8C
- mlib_MatrixMaximum_S16
- mlib_MatrixMaximum_S32
- mlib_MatrixMaximum_S8
- mlib_MatrixMaximum_U8
- mlib_MatrixMinimum_D64
- mlib_MatrixMinimum_F32
- mlib_MatrixMinimumMag_D64C
- mlib_MatrixMinimumMag_F32C
- mlib_MatrixMinimumMag_S16C
- mlib_MatrixMinimumMag_S32C
- mlib_MatrixMinimumMag_S8C
- mlib_MatrixMinimumMag_U8C
- mlib_MatrixMinimum_S16
- mlib_MatrixMinimum_S32
- mlib_MatrixMinimum_S8
- mlib_MatrixMinimum_U8
- mlib_MatrixMul_S16C_S16C_Mod
- mlib_MatrixMul_S16C_S16C_Sat
- mlib_MatrixMul_S16C_S8C_Mod
- mlib_MatrixMul_S16C_S8C_Sat
- mlib_MatrixMul_S16C_U8C_Mod
- mlib_MatrixMul_S16C_U8C_Sat
- mlib_MatrixMul_S16_S16_Mod
- mlib_MatrixMul_S16_S16_Sat
- mlib_MatrixMul_S16_S8_Mod
- mlib_MatrixMul_S16_S8_Sat
- mlib_MatrixMul_S16_U8_Mod
- mlib_MatrixMul_S16_U8_Sat
- mlib_MatrixMul_S32C_S16C_Mod
- mlib_MatrixMul_S32C_S16C_Sat
- mlib_MatrixMul_S32C_S32C_Mod
- mlib_MatrixMul_S32C_S32C_Sat
- mlib_MatrixMul_S32_S16_Mod
- mlib_MatrixMul_S32_S16_Sat
- mlib_MatrixMul_S32_S32_Mod
- mlib_MatrixMul_S32_S32_Sat
- mlib_MatrixMul_S8C_S8C_Mod
- mlib_MatrixMul_S8C_S8C_Sat
- mlib_MatrixMul_S8_S8_Mod
- mlib_MatrixMul_S8_S8_Sat
- mlib_MatrixMulShift_S16C_S16C_Mod
- mlib_MatrixMulShift_S16C_S16C_Sat
- mlib_MatrixMulShift_S16_S16_Mod
- mlib_MatrixMulShift_S16_S16_Sat
- mlib_MatrixMulS_S16C_Mod
- mlib_MatrixMulS_S16C_S16C_Mod
- mlib_MatrixMulS_S16C_S16C_Sat
- mlib_MatrixMulS_S16C_S8C_Mod
- mlib_MatrixMulS_S16C_S8C_Sat
- mlib_MatrixMulS_S16C_Sat
- mlib_MatrixMulS_S16C_U8C_Mod
- mlib_MatrixMulS_S16C_U8C_Sat
- mlib_MatrixMulS_S16_Mod
- mlib_MatrixMulS_S16_S16_Mod
- mlib_MatrixMulS_S16_S16_Sat
- mlib_MatrixMulS_S16_S8_Mod
- mlib_MatrixMulS_S16_S8_Sat
- mlib_MatrixMulS_S16_U8_Mod
- mlib_MatrixMulS_S16_U8_Sat
- mlib_MatrixMulS_S32C_Mod
- mlib_MatrixMulS_S32C_S16C_Mod
- mlib_MatrixMulS_S32C_S16C_Sat
- mlib_MatrixMulS_S32C_S32C_Mod
- mlib_MatrixMulS_S32C_S32C_Sat
- mlib_MatrixMulS_S32C_Sat
- mlib_MatrixMulS_S8C_Mod
- mlib_MatrixMulS_S8C_S8C_Mod
- mlib_MatrixMulS_S8C_S8C_Sat
- mlib_MatrixMulS_S8C_Sat
- mlib_MatrixMulS_S8_Mod
- mlib_MatrixMulS_S8_S8_Mod
- mlib_MatrixMulS_S8_S8_Sat
- mlib_MatrixMulS_S8_Sat
- mlib_MatrixMulSShift_S16C_Mod
- mlib_MatrixMulSShift_S16C_S16C_Mod
- mlib_MatrixMulSShift_S16C_S16C_Sat
- mlib_MatrixMulSShift_S16C_Sat
- mlib_MatrixMulSShift_S16_Mod
- mlib_MatrixMulSShift_S16_S16_Mod
- mlib_MatrixMulSShift_S16_S16_Sat
- mlib_MatrixMulSShift_S16_Sat
- mlib_MatrixMulSShift_S32C_Mod
- mlib_MatrixMulSShift_S32C_S32C_Mod
- mlib_MatrixMulSShift_S32C_S32C_Sat
- mlib_MatrixMulSShift_S32C_Sat
- mlib_MatrixMulSShift_S8C_Mod
- mlib_MatrixMulSShift_S8C_S8C_Mod
- mlib_MatrixMulSShift_S8C_S8C_Sat
- mlib_MatrixMulSShift_S8C_Sat
- mlib_MatrixMulSShift_U8C_Mod
- mlib_MatrixMulSShift_U8C_Sat
- mlib_MatrixMulSShift_U8C_U8C_Mod
- mlib_MatrixMulSShift_U8C_U8C_Sat
- mlib_MatrixMulS_U8C_Mod
- mlib_MatrixMulS_U8C_Sat
- mlib_MatrixMulS_U8C_U8C_Mod
- mlib_MatrixMulS_U8C_U8C_Sat
- mlib_MatrixMul_U8C_U8C_Mod
- mlib_MatrixMul_U8C_U8C_Sat
- mlib_MatrixScale_S16C_Mod
- mlib_MatrixScale_S16C_S16C_Mod
- mlib_MatrixScale_S16C_S16C_Sat
- mlib_MatrixScale_S16C_Sat
- mlib_MatrixScale_S8C_Mod
- mlib_MatrixScale_S8C_S8C_Mod
- mlib_MatrixScale_S8C_S8C_Sat
- mlib_MatrixScale_S8C_Sat
- mlib_MatrixScale_S16C_S8C_Sat
- mlib_MatrixScale_S16C_Sat
- mlib_MatrixScale_S16C_U8C_Mod
- mlib_MatrixScale_S16C_U8C_Sat
- mlib_MatrixScale_S16_Mod
- mlib_MatrixScale_S16_S16_Mod
- mlib_MatrixScale_S16_S16_Sat
- mlib_MatrixScale_S16_S8_Mod
- mlib_MatrixScale_S16_S8_Sat
- mlib_MatrixScale_S16_Sat
- mlib_MatrixScale_S16_U8_Mod
- mlib_MatrixScale_S16_U8_Sat
- mlib_MatrixScale_S32C_Mod
- mlib_MatrixScale_S32C_S16C_Mod
- mlib_MatrixScale_S32C_S16C_Sat
- mlib_MatrixScale_S32C_S32C_Mod
- mlib_MatrixScale_S32C_S32C_Sat
- mlib_MatrixScale_S32C_Sat
- mlib_MatrixScale_S32_Mod
- mlib_MatrixScale_S32_S16_Mod
- mlib_MatrixScale_S32_S16_Sat
- mlib_MatrixScale_S32_S32_Mod
- mlib_MatrixScale_S32_S32_Sat
- mlib_MatrixScale_S32_Sat
- mlib_MatrixScale_S8C_Mod
- mlib_MatrixScale_S8C_S8C_Mod
- mlib_MatrixScale_S8C_S8C_Sat
- mlib_MatrixScale_S8C_Sat
- mlib_MatrixScale_S8_Mod
- mlib_MatrixScale_S8_S8_Mod
- mlib_MatrixScale_S8_S8_Sat
- mlib_MatrixScale_S8_Sat
- mlib_MatrixScale_U8C_Mod
- mlib_MatrixScale_U8C_Sat
- mlib_MatrixScale_U8C_U8C_Mod
- mlib_MatrixScale_U8C_U8C_Sat
- mlib_MatrixScale_U8_Mod
- mlib_MatrixScale_U8_Sat
- mlib_MatrixScale_U8_U8_Mod
- mlib_MatrixScale_U8_U8_Sat
- mlib_MatrixSub_S16C_Mod
- mlib_MatrixSub_S16C_S16C_Mod
- mlib_MatrixSub_S16C_S16C_Sat
- mlib_MatrixSub_S16C_S8C_Mod
- mlib_MatrixSub_S8_S8_Mod
- mlib_MatrixSub_S8_S8_Sat
- mlib_MatrixSub_S8_Sat
- mlib_MatrixSub_U8_Mod
- mlib_MatrixSub_U8_Sat
- mlib_MatrixSub_U8_U8_Mod
- mlib_MatrixSub_U8_U8_Sat
<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>mlib_MatrixSub_S16C_S8C_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S16C_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S16C_U8C_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S16C_U8C_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S16_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S16_S16_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S16_S16_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S16_S8_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S16_S8_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S16_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S16_U8_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S16_U8_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S32C_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S32C_S16C_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S32C_S16C_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S32C_S32C_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S32C_S32C_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S32C_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S32_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S32_S16_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S32_S16_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S32_S32_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S32_S32_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S32_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S8C_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S8C_S8C_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S8C_S8C_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S8C_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S8_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S8_S8_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S8_S8_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSub_S8_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSubS_S16C_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSubS_S16C_S16C_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSubS_S16C_S16C_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSubS_S16C_S8C_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSubS_S16C_S8C_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSubS_S16C_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSubS_S16C_U8C_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSubS_S16C_U8C_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSubS_S16_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSubS_S16_S16_Mod</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSubS_S16_S16_Sat</code></td>
</tr>
<tr>
<td><code>mlib_MatrixSubS_S16_S8_Mod</code></td>
</tr>
</tbody>
</table>
- mlib_MatrixSubS_S16_S8_Sat
- mlib_MatrixSubS_S16_Sat
- mlib_MatrixSubS_S16_U8_Mod
- mlib_MatrixSubS_S16_U8_Sat
- mlib_MatrixSubS_S32C_Mod
- mlib_MatrixSubS_S32C_S16C_Mod
- mlib_MatrixSubS_S32C_S16C_Sat
- mlib_MatrixSubS_S32C_S32C_Mod
- mlib_MatrixSubS_S32C_S32C_Sat
- mlib_MatrixSubS_S32C_Sat
- mlib_MatrixSubS_S32_Mod
- mlib_MatrixSubS_S32_S16_Mod
- mlib_MatrixSubS_S32_S16_Sat
- mlib_MatrixSubS_S32_S32_Mod
- mlib_MatrixSubS_S32_S32_Sat
- mlib_MatrixSubS_S32_S32_Sat
- mlib_MatrixSubS_S32_Sat
- mlib_MatrixSubS_S8C_Mod
- mlib_MatrixSubS_S8C_S8C_Mod
- mlib_MatrixSubS_S8C_S8C_Sat
- mlib_MatrixSubS_S8C_Sat
- mlib_MatrixSubS_S8_Mod
- mlib_MatrixSubS_S8_S8_Mod
- mlib_MatrixSubS_S8_S8_Sat
- mlib_MatrixSubS_S8_Sat
- mlib_MatrixSubS_U8C_Mod
- mlib_MatrixSubS_U8C_Sat
- mlib_MatrixSubS_U8C_U8C_Mod
- mlib_MatrixSubS_U8C_U8C_Sat
- mlib_MatrixSubS_U8_Mod
- mlib_MatrixSubS_U8_Sat
- mlib_MatrixSubS_U8_U8_Mod
- mlib_MatrixSubS_U8_U8_Sat
- mlib_MatrixSub_U8C_Mod
- mlib_MatrixSub_U8C_Sat
- mlib_MatrixSub_U8C_U8C_Mod
- mlib_MatrixSub_U8C_U8C_Sat
- mlib_MatrixSub_U8_Mod
- mlib_MatrixSub_U8_Sat
- mlib_MatrixSub_U8_U8_Mod
- mlib_MatrixSub_U8_U8_Sat
- mlib_MatrixTranspose_S16
- mlib_MatrixTranspose_S16C
- mlib_MatrixTranspose_S16C_S16C
- mlib_MatrixTranspose_S16_S16
- mlib_MatrixTranspose_S32
- mlib_MatrixTranspose_S32C
- mlib_MatrixTranspose_S32C_S32C
- mlib_MatrixTranspose_S32_S32
- mlib_MatrixTranspose_S8
- mlib_MatrixTranspose_S8C
- mlib_MatrixTranspose_S8C_S8C
- mlib_MatrixTranspose_S8_S8
- mlib_MatrixTranspose_U8
- mlib_MatrixTranspose_U8C
- mlib_MatrixTranspose_U8C_U8C
- mlib_MatrixTranspose_U8_U8
- mlib_MatrixUnit_S16
- mlib_MatrixUnit_S16C
- mlib_MatrixUnit_S32
- mlib_MatrixUnit_S32C
- mlib_MatrixUnit_S8
- mlib_MatrixUnit_S8C
- mlib_MatrixUnit_U8
- mlib_MatrixUnit_U8C
- mlib_VectorAdd_S16C_Mod
- mlib_VectorAdd_S16C_S16C_Mod
- mlib_VectorAdd_S16C_S16C_Sat
- mlib_VectorAdd_S16C_S8C_Mod
- mlib_VectorAdd_S16C_S8C_Sat
- mlib_VectorAdd_S16C_Sat
- mlib_VectorAdd_S16C_U8C_Mod
- mlib_VectorAdd_S16C_U8C_Sat
- mlib_VectorAdd_S16_Mod
- mlib_VectorAdd_S16_S16_Mod
- mlib_VectorAdd_S16_S16_Sat
- mlib_VectorAdd_S16_S8_Mod
- mlib_VectorAdd_S16_S8_Sat
- mlib_VectorAdd_S16_Sat
- mlib_VectorAdd_S16_U8_Mod
- mlib_VectorAdd_S16_U8_Sat
- mlib_VectorAdd_S32C_Mod
- mlib_VectorAdd_S32C_S16C_Mod
- mlib_VectorAdd_S32C_S16C_Sat
- mlib_VectorAdd_S32C_S32C_Mod
- mlib_VectorAdd_S32C_S32C_Sat
- mlib_VectorAdd_S32C_Sat
- mlib_VectorAdd_S32C_U8C_Mod
- mlib_VectorAdd_S32C_U8C_Sat
- mlib_VectorAdd_S32_Mod
- mlib_VectorAdd_S32_S16C_Mod
- mlib_VectorAdd_S32_S16C_Sat
- mlib_VectorAdd_S32_S32C_Mod
- mlib_VectorAdd_S32_S32C_Sat
- mlib_VectorAdd_S32_Sat
- mlib_VectorAdd_S32_U8C_Mod
- mlib_VectorAdd_S32_U8C_Sat
- mlib_VectorAdd_S32_S16C_Mod
- mlib_VectorAdd_S32_S16C_Sat
- mlib_VectorAdd_S32_S16C_Sat
- mlib_VectorAdd_S32_S32C_Mod
- mlib_VectorAdd_S32_S32C_Sat
- `mlib_VectorAdd_S32_S16_Sat`
- `mlib_VectorAdd_S32_S32_Mod`
- `mlib_VectorAdd_S32_S32_Sat`
- `mlib_VectorAdd_S32_Sat`
- `mlib_VectorAdd_S8C_Mod`
- `mlib_VectorAdd_S8C_S8C_Mod`
- `mlib_VectorAdd_S8C_S8C_Sat`
- `mlib_VectorAdd_S8C_Sat`
- `mlib_VectorAdd_S8_S8_Mod`
- `mlib_VectorAdd_S8_S8_Sat`
- `mlib_VectorAdd_S8_Sat`
- `mlib_VectorAddS_S16C_Mod`
- `mlib_VectorAddS_S16C_S16C_Mod`
- `mlib_VectorAddS_S16C_S16C_Sat`
- `mlib_VectorAddS_S16C_S32C_Mod`
- `mlib_VectorAddS_S16C_S32C_Sat`
- `mlib_VectorAddS_S16C_Sat`
- `mlib_VectorAddS_S16C_U8C_Mod`
- `mlib_VectorAddS_S16C_U8C_Sat`
- `mlib_VectorAddS_S16_Mod`
- `mlib_VectorAddS_S16_S16_Mod`
- `mlib_VectorAddS_S16_S16_Sat`
- `mlib_VectorAddS_S16_S8_Mod`
- `mlib_VectorAddS_S16_S8_Sat`
- `mlib_VectorAddS_S16_Sat`
- `mlib_VectorAddS_S8C_Mod`
- `mlib_VectorAddS_S8C_S8C_Mod`
- `mlib_VectorAddS_S8C_S8C_Sat`
- `mlib_VectorAddS_S8C_Sat`
- `mlib_VectorAddS_S8_Mod`
- `mlib_VectorAddS_S8_S8_Mod`
- `mlib_VectorAddS_S8_S8_Sat`
- `mlib_VectorAddS_S8_Sat`
- `mlib_VectorAddS_U8C_Mod`
- `mlib_VectorAddS_U8C_Sat`
- `mlib_VectorAddS_U8C_U8C_Mod`
- `mlib_VectorAddS_U8C_U8C_Sat`
- `mlib_VectorAddS_U8_Mod`
- `mlib_VectorAddS_U8_Sat`
- `mlib_VectorAddS_U8_U8_Mod`
- `mlib_VectorAddS_U8_U8_Sat`
- `mlib_VectorAdd_U8C_Mod`
- `mlib_VectorAdd_U8C_Sat`
- `mlib_VectorAdd_U8C_U8C_Mod`
- `mlib_VectorAdd_U8C_U8C_Sat`
- `mlib_VectorAdd_U8_Mod`
- `mlib_VectorAdd_U8_Sat`
- `mlib_VectorAdd_U8_U8_Mod`
- `mlib_VectorAdd_U8_U8_Sat`
- `mlib_VectorAng_S16C`
- `mlib_VectorAng_S32C`
- `mlib_VectorAng_S8C`
- `mlib_VectorAng_U8C`
- `mlib_VectorAve_S16`
- `mlib_VectorAve_S16C`
- `mlib_VectorAve_S16C_S16C`
- `mlib_VectorAve_S16C_S8C`
- `mlib_VectorAve_S16C_U8C`
- `mlib_VectorAve_S16_S16`
- `mlib_VectorAve_S16_S8`
- `mlib_VectorAve_S16_U8`
- `mlib_VectorAve_S32`
- `mlib_VectorAve_S32C`
- `mlib_VectorAve_S32C_S16C`
- `mlib_VectorAve_S32C_S32C`
- `mlib_VectorAve_S32_S16`
- `mlib_VectorAve_S32_S32`
- `mlib_VectorAve_S8`
- `mlib_VectorAve_S8C`
- `mlib_VectorAve_S8C_S8C`
- `mlib_VectorAve_S8_S8`
- `mlib_VectorAve_U8`
- `mlib_VectorAve_U8C`
- mlib_VectorAve_U8C_U8C
- mlib_VectorAve_U8_U8
- mlib_VectorConjRev_S16C_S16C_Sat
- mlib_VectorConjRev_S32C_S32C_Sat
- mlib_VectorConjRev_S8C_S8C_Sat
- mlib_VectorConj_S16C_S16C_Sat
- mlib_VectorConj_S16C_Sat
- mlib_VectorConj_S32C_S32C_Sat
- mlib_VectorConj_S32C_Sat
- mlib_VectorConj_S8C_S8C_Sat
- mlib_VectorConj_S8C_Sat
- mlib_VectorConjSymExt_S16C_S16C_Sat
- mlib_VectorConjSymExt_S32C_S32C_Sat
- mlib_VectorConjSymExt_S8C_S8C_Sat
- mlib_VectorConvert_S16C_S32C_Mod
- mlib_VectorConvert_S16C_S32C_Sat
- mlib_VectorConvert_S16C_S8C_Mod
- mlib_VectorConvert_S16C_S8C_Sat
- mlib_VectorConvert_S16C_U8C_Mod
- mlib_VectorConvert_S16C_U8C_Sat
- mlib_VectorConvert_S16_S32_Mod
- mlib_VectorConvert_S16_S32_Sat
- mlib_VectorConvert_S16_S8_Mod
- mlib_VectorConvert_S16_S8_Sat
- mlib_VectorConvert_S16_U8_Mod
- mlib_VectorConvert_S16_U8_Sat
- mlib_VectorConvert_S32C_S16C_Mod
- mlib_VectorConvert_S32C_S16C_Sat
- mlib_VectorConvert_S32C_S8C_Mod
- mlib_VectorConvert_S32C_S8C_Sat
- mlib_VectorConvert_S32C_U8C_Mod
- mlib_VectorConvert_S32C_U8C_Sat
- mlib_VectorConvert_S32_S16_Mod
- mlib_VectorConvert_S32_S16_Sat
- mlib_VectorConvert_S32_S8_Mod
- mlib_VectorConvert_S32_S8_Sat
- mlib_VectorConvert_S32_U8_Mod
- mlib_VectorConvert_S32_U8_Sat
- mlib_VectorConvert_S8C_S16C_Mod
- mlib_VectorConvert_S8C_S16C_Sat
- mlib_VectorConvert_S8C_S32C_Mod
- mlib_VectorConvert_S8C_S32C_Sat
- mlib_VectorConvert_S8C_U8C_Mod
- mlib_VectorConvert_S8C_U8C_Sat
- mlib_VectorConvert_S8_S16_Mod
- mlib_VectorConvert_S8_S16_Sat
- mlib_VectorConvert_S8_S32_Mod
- mlib_VectorConvert_S8_S32_Sat
- mlib_VectorConvert_S8_U8_Mod
- mlib_VectorConvert_S8_U8_Sat
- mlib_VectorConvert_U8C_S16C_Mod
- mlib_VectorConvert_U8C_S16C_Sat
- mlib_VectorConvert_U8C_S32C_Mod
- mlib_VectorConvert_U8C_S32C_Sat
- mlib_VectorConvert_U8C_S8C_Mod
- mlib_VectorConvert_U8C_S8C_Sat
- mlib_VectorConvert_U8_S16_Mod
- mlib_VectorConvert_U8_S16_Sat
- mlib_VectorConvert_U8_S32_Mod
- mlib_VectorConvert_U8_S32_Sat
- mlib_VectorCopy_S16
- mlib_VectorCopy_S16C
- mlib_VectorCopy_S32
- mlib_VectorCopy_S32C
- mlib_VectorCopy_S8
- mlib_VectorCopy_S8C
- mlib_VectorCopy_U8
- mlib_VectorCopy_U8C
- mlib_VectorDistance_S16_Sat
- mlib_VectorDistance_S32_Sat
- mlib_VectorDistance_S8_Sat
- mlib_VectorDotProd_S16C_Sat
- mlib_VectorDotProd_S16_Sat
- mlib_VectorDotProd_S32C_Sat
- mlib_VectorDotProd_S32_Sat
- mlib_VectorDotProd_S8_Sat
- mlib_VectorDotProd_U8C_Sat
- mlib_VectorDotProd_U8_Sat
- mlib_VectorMag_S16C
- mlib_VectorMag_S32C
- mlib_VectorMag_S8C
- mlib_VectorMag_U8C
- mlib_VectorMaximum_D64
- mlib_VectorMaximum_F32
- mlib_VectorMaximumMag_D64C
- mlib_VectorMaximumMag_F32C
- mlib_VectorMaximumMag_S16C
- mlib_VectorMaximumMag_S32C
- mlib_VectorMaximumMag_S8C
- mlib_VectorMaximumMag_U8C
- mlib_VectorMaximum_S16
- mlib_VectorMaximum_S32
- mlib_VectorMaximum_S8
- mlib_VectorMaximum_U8
- mlib_VectorMerge_S16C_S16
- mlib_VectorMerge_S16C_S16
- mlib_VectorMerge_S16C_S8C
- mlib_VectorMerge_S8C_S8
- mlib_VectorMerge_U8C_U8
- mlib_VectorMinimum_D64
- mlib_VectorMinimum_F32
- mlib_VectorMinimumMag_D64C
- mlib_VectorMinimumMag_F32C
- mlib_VectorMinimumMag_S16C
- mlib_VectorMinimumMag_S32C
- mlib_VectorMinimumMag_S8C
- mlib_VectorMinimumMag_U8C
- mlib_VectorMinimum_S16
- mlib_VectorMinimum_S32
- mlib_VectorMinimum_S8
- mlib_VectorMinimum_U8
- mlib_VectorMulM_S16C_S16C_Mod
- mlib_VectorMulM_S16C_S16C_Sat
- mlib_VectorMulM_S16C_S8C_Mod
- mlib_VectorMulM_S16C_S8C_Sat
- mlib_VectorMulM_S16C_U8C_Mod
- mlib_VectorMulM_S16C_U8C_Sat
- mlib_VectorMulM_S16_S16_Mod
- mlib_VectorMulM_S16_S16_Sat
- mlib_VectorMulM_S16_S8_Mod
- mlib_VectorMulM_S16_S8_Sat
- mlib_VectorMulM_S16_U8_Mod
- mlib_VectorMulM_S16_U8_Sat
- mlib_VectorMulM_S32C_S16C_Mod
- mlib_VectorMulM_S32C_S16C_Sat
- mlib_VectorMulM_S32C_S32C_Mod
- mlib_VectorMulM_S32C_S32C_Sat
- mlib_VectorMulM_S32C_S16_Mod
- mlib_VectorMulM_S32C_S16_Sat
- `mlib_VectorMulM_S32_S32_Mod`
- `mlib_VectorMulM_S32_S32_Sat`
- `mlib_VectorMulM_S8C_S8C_Mod`
- `mlib_VectorMulM_S8C_S8C_Sat`
- `mlib_VectorMulM_S8_S8_Mod`
- `mlib_VectorMulM_S8_S8_Sat`
- `mlib_VectorMulMShift_S16C_S16C_Mod`
- `mlib_VectorMulMShift_S16C_S16C_Sat`
- `mlib_VectorMulMShift_S16_S16_Mod`
- `mlib_VectorMulMShift_S16_S16_Sat`
- `mlib_VectorMulM_U8C_U8C_Mod`
- `mlib_VectorMulM_U8C_U8C_Sat`
- `mlib_VectorMulM_U8_U8_Mod`
- `mlib_VectorMulM_U8_U8_Sat`
- `mlib_VectorMul_S16C_Mod`
- `mlib_VectorMul_S16C_S16C_Mod`
- `mlib_VectorMul_S16C_S16C_Sat`
- `mlib_VectorMul_S16C_S8C_Mod`
- `mlib_VectorMul_S16C_S8C_Sat`
- `mlib_VectorMul_S16C_Sat`
- `mlib_VectorMul_S16_U8_Mod`
- `mlib_VectorMul_S16_U8_Sat`
- `mlib_VectorMul_S32C_Mod`
- `mlib_VectorMul_S32C_S16C_Mod`
- `mlib_VectorMul_S32C_S16C_Sat`
- `mlib_VectorMul_S32C_S32C_Mod`
- `mlib_VectorMul_S32C_S32C_Sat`
- `mlib_VectorMul_S32C_Sat`
- `mlib_VectorMul_S32_Mod`
- `mlib_VectorMul_S32_S16_Mod`
- `mlib_VectorMul_S32_S16_Sat`
- `mlib_VectorMul_S32_S32_Mod`
- `mlib_VectorMul_S32_S32_Sat`
- mlib_VectorMul_S8C_S8C_Sat
- mlib_VectorMul_S8C_Sat
- mlib_VectorMul_S8_Mod
- mlib_VectorMul_S8_S8_Mod
- mlib_VectorMul_S8_S8_Sat
- mlib_VectorMul_S8_Sat
- mlib_VectorMulSAdd_S16C_Mod
- mlib_VectorMulSAdd_S16C_S16C_Mod
- mlib_VectorMulSAdd_S16C_S16C_Sat
- mlib_VectorMulSAdd_S16C_S8C_Mod
- mlib_VectorMulSAdd_S16C_S8C_Sat
- mlib_VectorMulSAdd_S16C_Sat
- mlib_VectorMulSAdd_S16C_U8C_Mod
- mlib_VectorMulSAdd_S16C_U8C_Sat
- mlib_VectorMulSAdd_S16_Mod
- mlib_VectorMulSAdd_S16_S16_Mod
- mlib_VectorMulSAdd_S16_S16_Sat
- mlib_VectorMulSAdd_S16_S8_Mod
- mlib_VectorMulSAdd_S16_S8_Sat
- mlib_VectorMulSAdd_S16_Sat
- mlib_VectorMulSAdd_S16_U8_Mod
- mlib_VectorMulSAdd_S16_U8_Sat
- mlib_VectorMulSAdd_S32C_Mod
- mlib_VectorMulSAdd_S32C_S16C_Mod
- mlib_VectorMulSAdd_S32C_S16C_Sat
- mlib_VectorMulSAdd_S32C_S32C_Mod
- mlib_VectorMulSAdd_S32C_S32C_Sat
- mlib_VectorMulSAdd_S32C_Sat
- mlib_VectorMulSAdd_S32_Mod
- mlib_VectorMulSAdd_S32_S16_Mod
- mlib_VectorMulSAdd_S32_S16_Sat
- mlib_VectorMulSAdd_S32_S32_Mod
- mlib_VectorMulSAdd_S32_S32_Sat
- mlib_VectorMulSAdd_S32_Sat
- mlib_VectorMulSAdd_S8C_Mod
- mlib_VectorMulSAdd_S8C_S8C_Mod
- mlib_VectorMulSAdd_S8C_S8C_Sat
- mlib_VectorMulSAdd_S8C_Sat
- mlib_VectorMulSAdd_S8_Mod
- mlib_VectorMulSAdd_S8_S8_Mod
- mlib_VectorMulSAdd_S8_S8_Sat
- mlib_VectorMulSAdd_S8_Sat
- mlib_VectorMulSAdd_U8C_Mod
- mlib_VectorMulSAdd_U8C_Sat
- mlib_VectorMulS_S16C_U8C_Mod
- mlib_VectorMulS_S16C_U8C_Sat
- mlib_VectorMulS_S16_Mod
- mlib_VectorMulS_S16_S16_Mod
- mlib_VectorMulS_S16_S16_Sat
- mlib_VectorMulS_S16_S8_Mod
- mlib_VectorMulS_S16_S8_Sat
- mlib_VectorMulS_S16_Sat
- mlib_VectorMulS_S16_U8_Mod
- mlib_VectorMulS_S16_U8_Sat
- mlib_VectorMulS_S32C_Mod
- mlib_VectorMulS_S32C_S16C_Mod
- mlib_VectorMulS_S32C_S16C_Sat
- mlib_VectorMulS_S32C_S32C_Mod
- mlib_VectorMulS_S32C_S32C_Sat
- mlib_VectorMulS_S32C_Sat
- mlib_VectorMulSShift_S16C_Mod
- mlib_VectorMulSShift_S16C_S16C_Mod
- mlib_VectorMulSShift_S16C_S16C_Sat
- mlib_VectorMulSShift_S16C_Sat
- mlib_VectorMulSShift_S16_Mod
- mlib_VectorMulSShift_S16_S16_Mod
- mlib_VectorMulSShift_S16_S16_Sat
- mlib_VectorMulSShift_S16_Sat
- mlib_VectorMulSShift_S32C_Mod
- mlib_VectorMulSShift_S32C_S32C_Mod
- mlib_VectorMulSShift_S32C_S32C_Sat
- mlib_VectorMulSShift_S32C_Sat
- mlib_VectorMulSShift_S32_Mod
- mlib_VectorMulSShift_S32_S32_Mod
- mlib_VectorMulSShift_S32_S32_Sat
- mlib_VectorMulSShift_S32_Sat
- mlib_VectorMulSShift_S32_S32_Sat
- mlib_VectorMulSShift_S32_Sat
- mlib_VectorMulSShift_S8C_Mod
- mlib_VectorMulSShift_S8C_S8C_Mod
- mlib_VectorMulSShift_S8C_S8C_Sat
- mlib_VectorMulSShift_S8C_Sat
- mlib_VectorMulSShift_S8_S8_Mod
- mlib_VectorMulSShift_S8_S8_Sat
- mlib_VectorMulSShift_S8_Sat
- mlib_VectorMulSShift_U8C_Mod
- mlib_VectorMulSShift_U8C_Sat
- mlib_VectorMulSShift_U8C_U8C_Mod
- mlib_VectorMulSShift_U8C_U8C_Sat
- mlib_VectorMulSShift_U8C_U8C_Sat
- mlib_VectorMulSShift_U8_C
- mlib_VectorMulSShift_U8_C_Sat
- mlib_VectorMulSShift_U8_U8_Mod
- mlib_VectorMulSShift_U8_U8_Sat
- mlib_VectorMulS_U8C_Mod
- mlib_VectorMulS_U8C_Sat
- mlib_VectorMulS_U8C_U8C_Mod
- mlib_VectorMulS_U8C_U8C_Sat
- mlib_VectorMulS_U8_Sat
- mlib_VectorMulS_U8_U8_Mod
- mlib_VectorMulS_U8_U8_Sat
- mlib_VectorMulS_U8C_Mod
- mlib_VectorMulS_U8C_Sat
- mlib_VectorMulS_U8C_U8C_Mod
- mlib_VectorMulS_U8C_U8C_Sat
- mlib_VectorMulS_U8C_U8C_Sat
- mlib_VectorMulS_U8_C
- mlib_VectorMulS_U8_C_Sat
- mlib_VectorMulS_U8_U8_Mod
- mlib_VectorMulS_U8_U8_Sat
- mlib_VectorNorm_S16_Sat
- mlib_VectorNorm_S32_Sat
- mlib_VectorNorm_S8_Sat
- mlib_VectorNorm_S_Sat
- mlib_VectorNorm_S8_Sat
- mlib_VectorNorm_S8_Sat
- mlib_VectorReverseByteOrder
- mlib_VectorReverseByteOrder_D64
- mlib_VectorReverseByteOrder_D64_D64
- mlib_VectorReverseByteOrder_F32
- mlib_VectorReverseByteOrder_F32_F32
- mlib_VectorReverseByteOrder_Inp

libmlib(3LIB)
- `mlib_VectorReverseByteOrder_S16`
- `mlib_VectorReverseByteOrder_S16_S16`
- `mlib_VectorReverseByteOrder_S32`
- `mlib_VectorReverseByteOrder_S32_S32`
- `mlib_VectorReverseByteOrder_S64`
- `mlib_VectorReverseByteOrder_S64_S64`
- `mlib_VectorReverseByteOrder_U16`
- `mlib_VectorReverseByteOrder_U16_U16`
- `mlib_VectorReverseByteOrder_U32`
- `mlib_VectorReverseByteOrder_U32_U32`
- `mlib_VectorReverseByteOrder_U64`
- `mlib_VectorReverseByteOrder_U64_U64`
- `mlib_VectorScale_S16C_Mod`
- `mlib_VectorScale_S16C_S16C_Mod`
- `mlib_VectorScale_S16C_S16C_Sat`
- `mlib_VectorScale_S16C_S8C_Mod`
- `mlib_VectorScale_S16C_S8C_Sat`
- `mlib_VectorScale_S16C_Sat`
- `mlib_VectorScale_S16C_U8C_Mod`
- `mlib_VectorScale_S16C_U8C_Sat`
- `mlib_VectorScale_S16_Mod`
- `mlib_VectorScale_S16_S16_Mod`
- `mlib_VectorScale_S16_S16_Sat`
- `mlib_VectorScale_S16_S8_Mod`
- `mlib_VectorScale_S16_S8_Sat`
- `mlib_VectorScale_S16_Sat`
- `mlib_VectorScale_S16_U8_Mod`
- `mlib_VectorScale_S16_U8_Sat`
- `mlib_VectorScale_S32C_Mod`
- `mlib_VectorScale_S32C_S16C_Mod`
- `mlib_VectorScale_S32C_S16C_Sat`
- `mlib_VectorScale_S32C_S32C_Mod`
- `mlib_VectorScale_S32C_S32C_Sat`
- `mlib_VectorScale_S32C_Sat`
- `mlib_VectorScale_S32C_S8C_Mod`
- `mlib_VectorScale_S32C_S8C_Sat`
- `mlib_VectorScale_S32_Sat`
- `mlib_VectorScale_S8C_Mod`
- `mlib_VectorScale_S8C_S8C_Mod`
- `mlib_VectorScale_S8C_S8C_Sat`
- `mlib_VectorScale_S8C_Sat`
- mlib_VectorScale_S8_Mod
- mlib_VectorScale_S8_S8_Mod
- mlib_VectorScale_S8_S8_Sat
- mlib_VectorScale_S8_Sat
- mlib_VectorScale_U8C_Mod
- mlib_VectorScale_U8C_Sat
- mlib_VectorScale_U8C_U8C_Mod
- mlib_VectorScale_U8C_U8C_Sat
- mlib_VectorScale_U8_Mod
- mlib_VectorScale_U8_Sat
- mlib_VectorScale_U8_U8_Mod
- mlib_VectorScale_U8_U8_Sat
- mlib_VectorSet_S16
- mlib_VectorSet_S16C
- mlib_VectorSet_S32
- mlib_VectorSet_S32C
- mlib_VectorSet_S8
- mlib_VectorSet_S8C
- mlib_VectorSet_U8
- mlib_VectorSet_U8C
- mlib_VectorSplit_S16_S16C
- mlib_VectorSplit_S32_S32C
- mlib_VectorSplit_S8_S8C
- mlib_VectorSplit_U8_U8C
- mlib_VectorSub_S16C_Mod
- mlib_VectorSub_S16C_S16C_Mod
- mlib_VectorSub_S16C_S16C_Sat
- mlib_VectorSub_S16C_S8C_Mod
- mlib_VectorSub_S16C_S8C_Sat
- mlib_VectorSub_S16C_Sat
- mlib_VectorSub_S16C_U8C_Mod
- mlib_VectorSub_S16C_U8C_Sat
- mlib_VectorSub_S16_S16_Mod
- mlib_VectorSub_S16_S16_Sat
- mlib_VectorSub_S16_S8_Mod
- mlib_VectorSub_S16_S8_Sat
- mlib_VectorSub_S16_Sat
- mlib_VectorSub_S16_U8_Mod
- mlib_VectorSub_S16_U8_Sat
- mlib_VectorSub_S16_S16C_Mod
- mlib_VectorSub_S32C_S16C_Mod
- mlib_VectorSub_S32C_S16C_Sat
- mlib_VectorSub_S32C_S32C_Mod
- mlib_VectorSub_S32C_S32C_Sat
- `mlib_VectorSub_S32C_S32C_Sat`
- `mlib_VectorSub_S32C_Sat`
- `mlib_VectorSub_S32_Mod`
- `mlib_VectorSub_S32_S16_Mod`
- `mlib_VectorSub_S32_S16_Sat`
- `mlib_VectorSub_S32_S32_Mod`
- `mlib_VectorSub_S32_S32_Sat`
- `mlib_VectorSub_S32_Sat`
- `mlib_VectorSub_S8C_Mod`
- `mlib_VectorSub_S8C_S8C_Mod`
- `mlib_VectorSub_S8C_S8C_Sat`
- `mlib_VectorSub_S8C_Sat`
- `mlib_VectorSub_S8_Mod`
- `mlib_VectorSub_S8_S8_Mod`
- `mlib_VectorSub_S8_S8_Sat`
- `mlib_VectorSub_S8_Sat`
- `mlib_VectorSubS_S16C_Mod`
- `mlib_VectorSubS_S16C_S16C_Mod`
- `mlib_VectorSubS_S16C_S16C_Sat`
- `mlib_VectorSubS_S16C_S8C_Mod`
- `mlib_VectorSubS_S16C_S8C_Sat`
- `mlib_VectorSubS_S16C_Sat`
- `mlib_VectorSubS_S16C_U8C_Mod`
- `mlib_VectorSubS_S16C_U8C_Sat`
- `mlib_VectorSubS_S16_Mod`
- `mlib_VectorSubS_S16_S16_Mod`
- `mlib_VectorSubS_S16_S16_Sat`
- `mlib_VectorSubS_S16_S8_Mod`
- `mlib_VectorSubS_S16_S8_Sat`
- `mlib_VectorSubS_S16_Sat`
- `mlib_VectorSubS_S16_U8_Mod`
- `mlib_VectorSubS_S16_U8_Sat`
- `mlib_VectorSubS_S32C_Mod`
- `mlib_VectorSubS_S32C_S16C_Mod`
- `mlib_VectorSubS_S32C_S16C_Sat`
- `mlib_VectorSubS_S32C_S32C_Mod`
- `mlib_VectorSubS_S32C_S32C_Sat`
- `mlib_VectorSubS_S32C_Sat`
- `mlib_VectorSubS_S8C_Mod`
- `mlib_VectorSubS_S8C_S8C_Mod`
- `mlib_VectorSubS_S8C_S8C_Sat`
- `mlib_VectorSubS_S8C_S8C_Sat`
- `mlib_VectorSubS_S8_Mod`
- `mlib_VectorSubS_S8_S8_Mod`
- `mlib_VectorSubS_S8_S8_Sat`
- `mlib_VectorSubS_S8_Sat`
- `mlib_VectorSubS_U8C_Mod`
- `mlib_VectorSubS_U8C_Sat`
- `mlib_VectorSubS_U8C_U8C_Mod`
- `mlib_VectorSubS_U8C_U8C_Sat`
- `mlib_VectorSubS_U8_Mod`
- `mlib_VectorSubS_U8_Sat`
- `mlib_VectorSubS_U8_U8_Mod`
- `mlib_VectorSubS_U8_U8_Sat`
- `mlib_VectorSub_U8C_Mod`
- `mlib_VectorSub_U8C_Sat`
- `mlib_VectorSub_U8C_U8C_Mod`
- `mlib_VectorSub_U8C_U8C_Sat`
- `mlib_VectorSub_U8_Mod`
- `mlib_VectorSub_U8_Sat`
- `mlib_VectorSub_U8_U8_Mod`
- `mlib_VectorSub_U8_U8_Sat`
- `mlib_VectorSumAbsDiff_S16_Sat`
- `mlib_VectorSumAbsDiff_S32_Sat`
- `mlib_VectorSumAbsDiff_S8_Sat`
- `mlib_VectorSumAbsDiff_U8_Sat`
- `mlib_VectorSumAbs_S16_Sat`
- `mlib_VectorSumAbs_S32_Sat`
- `mlib_VectorSumAbs_S8_Sat`
- `mlib_VectorSumAbs_U8_Sat`
- `mlib_VectorZero_S16`
- `mlib_VectorZero_S16C`
- `mlib_VectorZero_S32`
- `mlib_VectorZero_S32C`
- `mlib_VectorZero_S8`
- `mlib_VectorZero_S8C`
- `mlib_VectorZero_U8`
- `mlib_VectorZero_U8C`

**Graphics Functions**
- `mlib_GraphicsBoundaryFill_32`
- `mlib_GraphicsBoundaryFill_8`
- `mlib_GraphicsDrawArc_32`
- `mlib_GraphicsDrawArc_8`
- mlib_GraphicsDrawArc_A_32
- mlib_GraphicsDrawArc_A_8
- mlib_GraphicsDrawArc_AB_32
- mlib_GraphicsDrawArc_AB_8
- mlib_GraphicsDrawArc_B_32
- mlib_GraphicsDrawArc_B_8
- mlib_GraphicsDrawArc_X_32
- mlib_GraphicsDrawArc_X_8
- mlib_GraphicsDrawCircle_32
- mlib_GraphicsDrawCircle_8
- mlib_GraphicsDrawCircle_A_32
- mlib_GraphicsDrawCircle_A_8
- mlib_GraphicsDrawCircle_AB_32
- mlib_GraphicsDrawCircle_AB_8
- mlib_GraphicsDrawCircle_B_32
- mlib_GraphicsDrawCircle_B_8
- mlib_GraphicsDrawCircle_X_32
- mlib_GraphicsDrawCircle_X_8
- mlib_GraphicsDrawEllipse_32
- mlib_GraphicsDrawEllipse_8
- mlib_GraphicsDrawEllipse_A_32
- mlib_GraphicsDrawEllipse_A_8
- mlib_GraphicsDrawEllipse_AB_32
- mlib_GraphicsDrawEllipse_AB_8
- mlib_GraphicsDrawEllipse_B_32
- mlib_GraphicsDrawEllipse_B_8
- mlib_GraphicsDrawEllipse_X_32
- mlib_GraphicsDrawEllipse_X_8
- mlib_GraphicsDrawLine_32
- mlib_GraphicsDrawLine_8
- mlib_GraphicsDrawLine_A_32
- mlib_GraphicsDrawLine_A_8
- mlib_GraphicsDrawLine_AB_32
- mlib_GraphicsDrawLine_AB_8
- mlib_GraphicsDrawLine_ABG_32
- mlib_GraphicsDrawLine_ABG_8
- mlib_GraphicsDrawLine_ABGZ_32
- mlib_GraphicsDrawLine_ABGZ_8
- mlib_GraphicsDrawLine_ABZ_32
- mlib_GraphicsDrawLine_ABZ_8
- mlib_GraphicsDrawLine_ABG_32
- mlib_GraphicsDrawLine_ABG_8
- mlib_GraphicsDrawLine_ABGZ_32
- mlib_GraphicsDrawLine_ABGZ_8
- mlib_GraphicsDrawLine_AZ_32
- mlib_GraphicsDrawLine_AZ_8
- mlib_GraphicsDrawLine_B_32
- mlib_GraphicsDrawLine_B_8
- mlib_GraphicsDrawLine_BG_32
- mlib_GraphicsDrawLine_BG_8
- mlib_GraphicsDrawLine_BGZ_32
- mlib_GraphicsDrawLine_BGZ_8
- mlib_GraphicsDrawLine_BZ_32
- mlib_GraphicsDrawLine_BZ_8
- mlib_GraphicsDrawLineFanSet_32
- mlib_GraphicsDrawLineFanSet_8
- mlib_GraphicsDrawLineFanSet_A_32
- mlib_GraphicsDrawLineFanSet_A_8
- mlib_GraphicsDrawLineFanSet_AB_32
- mlib_GraphicsDrawLineFanSet_AB_8
- mlib_GraphicsDrawLineFanSet_ABG_32
- mlib_GraphicsDrawLineFanSet_ABG_8
- mlib_GraphicsDrawLineFanSet_ABGZ_32
- mlib_GraphicsDrawLineFanSet_ABGZ_8
- mlib_GraphicsDrawLineFanSet_ABZ_32
- mlib_GraphicsDrawLineFanSet_ABZ_8
- mlib_GraphicsDrawLineFanSet_AZ_32
- mlib_GraphicsDrawLineFanSet_AZ_8
- mlib_GraphicsDrawLineFanSet_B_32
- mlib_GraphicsDrawLineFanSet_B_8
- mlib_GraphicsDrawLineFanSet_BG_32
- mlib_GraphicsDrawLineFanSet_BG_8
- mlib_GraphicsDrawLineFanSet_BGZ_32
- mlib_GraphicsDrawLineFanSet_BGZ_8
- mlib_GraphicsDrawLineFanSet_BZ_32
- mlib_GraphicsDrawLineFanSet_BZ_8
- mlib_GraphicsDrawLineFanSet_G_32
- mlib_GraphicsDrawLineFanSet_G_8
- mlib_GraphicsDrawLineFanSet_GZ_32
- mlib_GraphicsDrawLineFanSet_GZ_8
- mlib_GraphicsDrawLineFanSet_X_32
- mlib_GraphicsDrawLineFanSet_X_8
- mlib_GraphicsDrawLineFanSet_Z_32
- mlib_GraphicsDrawLineFanSet_Z_8
- mlib_GraphicsDrawLine_G_32
- mlib_GraphicsDrawLine_G_8
- mlib_GraphicsDrawLine_GZ_32
- mlib_GraphicsDrawLine_GZ_8
- mlib_GraphicsDrawLineSet_32
- mlib_GraphicsDrawLineSet_8
- mlib_GraphicsDrawLineSet_A_32
- mlib_GraphicsDrawLineSet_A_8
- mlib_GraphicsDrawLineSet_AB_32
- mlib_GraphicsDrawLineSet_AB_8
- mlib_GraphicsDrawLineSet_ABG_32
- mlib_GraphicsDrawLineSet_ABG_8
- mlib_GraphicsDrawLineSet_ABGZ_32
- mlib_GraphicsDrawLineSet_ABGZ_8
- mlib_GraphicsDrawLineSet_ABZ_32
- mlib_GraphicsDrawLineSet_ABZ_8
- mlib_GraphicsDrawLineSet_AG_32
- mlib_GraphicsDrawLineSet_AG_8
- mlib_GraphicsDrawLineSet_AGZ_32
- mlib_GraphicsDrawLineSet_AGZ_8
- mlib_GraphicsDrawLineSet_AZ_32
- mlib_GraphicsDrawLineSet_AZ_8
- mlib_GraphicsDrawLineSet_B_32
- mlib_GraphicsDrawLineSet_B_8
- mlib_GraphicsDrawLineSet_BG_32
- mlib_GraphicsDrawLineSet_BG_8
- mlib_GraphicsDrawLineSet_BGZ_32
- mlib_GraphicsDrawLineSet_BGZ_8
- mlib_GraphicsDrawLineSet_BZ_32
- mlib_GraphicsDrawLineSet_BZ_8
- mlib_GraphicsDrawLineSet_G_32
- mlib_GraphicsDrawLineSet_G_8
- mlib_GraphicsDrawLineSet_GZ_32
- mlib_GraphicsDrawLineSet_GZ_8
- mlib_GraphicsDrawLineSet_X_32
- mlib_GraphicsDrawLineSet_X_8
- mlib_GraphicsDrawLineSet_Z_32
- mlib_GraphicsDrawLineSet_Z_8
- mlib_GraphicsDrawLineStripSet_32
- mlib_GraphicsDrawLineStripSet_8
- mlib_GraphicsDrawLineStripSet_A_32
- mlib_GraphicsDrawLineStripSet_A_8
- mlib_GraphicsDrawLineStripSet_AB_32
- mlib_GraphicsDrawLineStripSet_AB_8
- mlib_GraphicsDrawPolygon_32
- mlib_GraphicsDrawPolygon_8
- mlib_GraphicsDrawPolygon_A_32
- mlib_GraphicsDrawPolygon_A_8
- mlib_GraphicsDrawPolygon_AB_32
- mlib_GraphicsDrawPolygon_AB_8
- mlib_GraphicsDrawPolygon_ABG_32
- mlib_GraphicsDrawPolygon_ABG_8
- mlib_GraphicsDrawPolygon_ABGZ_32
- mlib_GraphicsDrawPolygon_ABGZ_8
- mlib_GraphicsDrawPolygon_ABZ_32
- mlib_GraphicsDrawPolygon_ABZ_8
- mlib_GraphicsDrawPolygon_AG_32
- mlib_GraphicsDrawPolygon_AG_8
- mlib_GraphicsDrawPolygon_AGZ_32
- mlib_GraphicsDrawPolygon_AGZ_8
- mlib_GraphicsDrawPolygon_AZ_32
- mlib_GraphicsDrawPolygon_AZ_8
- mlib_GraphicsDrawPolygon_B_32
- mlib_GraphicsDrawPolygon_B_8
- mlib_GraphicsDrawPolygon_BG_32
- mlib_GraphicsDrawPolygon_BG_8
- mlib_GraphicsDrawPolygon_BGZ_32
- mlib_GraphicsDrawPolygon_BGZ_8
- mlib_GraphicsDrawPolygon_BZ_32
- mlib_GraphicsDrawPolygon_BZ_8
- mlib_GraphicsDrawPolygon_G_32
- mlib_GraphicsDrawPolygon_G_8
- mlib_GraphicsDrawPolygon_GZ_32
- mlib_GraphicsDrawPolygon_GZ_8
- mlib_GraphicsDrawPolygon_X_32
- mlib_GraphicsDrawPolygon_X_8
- mlib_GraphicsDrawPolygon_Z_32
- mlib_GraphicsDrawPolygon_Z_8
- mlib_GraphicsDrawPolyline_32
- mlib_GraphicsDrawPolyline_8
- mlib_GraphicsDrawPolyline_A_32
- mlib_GraphicsDrawPolyline_A_8
- mlib_GraphicsDrawPolyline_AB_32
- mlib_GraphicsDrawPolyline_AB_8
- mlib_GraphicsDrawPolyline_ABG_32
- mlib_GraphicsDrawPolyline_ABG_8
- mlib_GraphicsDrawPolyline_ABGZ_32
- mlib_GraphicsDrawPolyline_ABGZ_8
- mlib_GraphicsDrawPolyline_ABZ_32
- mlib_GraphicsDrawPolyline_ABZ_8
- mlib_GraphicsDrawPolyline_AG_32
- mlib_GraphicsDrawPolyline_AG_8
- mlib_GraphicsDrawPolyline_AGZ_32
- mlib_GraphicsDrawPolyline_AGZ_8
- mlib_GraphicsDrawPolyline_AZ_32
- mlib_GraphicsDrawPolyline_AZ_8
- mlib_GraphicsDrawPolyline_B_32
- mlib_GraphicsDrawPolyline_B_8
- mlib_GraphicsDrawPolyline_BG_32
- mlib_GraphicsDrawPolyline_BG_8
- mlib_GraphicsDrawPolyline_BGZ_32
- mlib_GraphicsDrawPolyline_BGZ_8
- mlib_GraphicsDrawPolyline_BZ_32
- mlib_GraphicsDrawPolyline_BZ_8
- mlib_GraphicsDrawPolyline_G_32
- mlib_GraphicsDrawPolyline_G_8
- mlib_GraphicsDrawPolyline_GZ_32
- mlib_GraphicsDrawPolyline_GZ_8
- mlib_GraphicsDrawPolyline_X_32
- mlib_GraphicsDrawPolyline_X_8
- mlib_GraphicsDrawPolypoint_32
- mlib_GraphicsDrawPolypoint_8
- mlib_GraphicsDrawPolypoint_B_32
- mlib_GraphicsDrawPolypoint_B_8
- mlib_GraphicsDrawPolypoint_X_32
- mlib_GraphicsDrawPolypoint_X_8
- mlib_GraphicsDrawRectangle_32
- mlib_GraphicsDrawRectangle_8
- mlib_GraphicsDrawRectangle_B_32
- mlib_GraphicsDrawRectangle_B_8
- mlib_GraphicsDrawRectangle_X_32
- mlib_GraphicsDrawRectangle_X_8
- mlib_GraphicsDrawTriangle_32
- mlib_GraphicsDrawTriangle_8
- mlib_GraphicsDrawTriangle_A_32
- mlib_GraphicsDrawTriangle_A_8
- mlib_GraphicsDrawTriangle_AB_32
- mlib_GraphicsDrawTriangle_AB_8
- mlib_GraphicsDrawTriangle_ABG_32
- mlib_GraphicsDrawTriangle_ABG_8
- mlib_GraphicsDrawTriangle_ABGZ_32
- mlib_GraphicsDrawTriangle_ABGZ_8
- mlib_GraphicsDrawTriangle_ABZ_32
- mlib_GraphicsDrawTriangle_ABZ_8
- mlib_GraphicsDrawTriangle_AG_32
- mlib_GraphicsDrawTriangle_AG_8
- mlib_GraphicsDrawTriangle_AGZ_32
- mlib_GraphicsDrawTriangle_AGZ_8
- mlib_GraphicsDrawTriangle_AZ_32
- mlib_GraphicsDrawTriangle_AZ_8
- mlib_GraphicsDrawTriangleFanSet_32
- mlib_GraphicsDrawTriangleFanSet_8
- mlib_GraphicsDrawTriangleFanSet_A_32
- mlib_GraphicsDrawTriangleFanSet_A_8
- mlib_GraphicsDrawTriangleFanSet_AB_32
- mlib_GraphicsDrawTriangleFanSet_AB_8
- mlib_GraphicsDrawTriangleFanSet_ABG_32
- mlib_GraphicsDrawTriangleFanSet_ABG_8
- mlib_GraphicsDrawTriangleFanSet_ABGZ_32
- mlib_GraphicsDrawTriangleFanSet_ABGZ_8
- mlib_GraphicsDrawTriangleFanSet_ABZ_32
- mlib_GraphicsDrawTriangleFanSet_ABZ_8
- mlib_GraphicsDrawTriangleFanSet_B_32
- mlib_GraphicsDrawTriangleFanSet_B_8
- mlib_GraphicsDrawTriangleFanSet_BG_32
- mlib_GraphicsDrawTriangleFanSet_BG_8
- mlib_GraphicsDrawTriangleFanSet_BGZ_32
- mlib_GraphicsDrawTriangleFanSet_BGZ_8
- mlib_GraphicsDrawTriangleFanSet_BZ_32
- mlib_GraphicsDrawTriangleFanSet_BZ_8
- mlib_GraphicsDrawTriangleFanSet_G_32
- mlib_GraphicsDrawTriangleFanSet_G_8
- mlib_GraphicsDrawTriangleFanSet_GZ_32
- mlib_GraphicsDrawTriangleFanSet_GZ_8
- mlib_GraphicsDrawTriangleFanSet_X_32
- mlib_GraphicsDrawTriangleFanSet_X_8
- mlib_GraphicsDrawTriangleFanSet_Z_32
- mlib_GraphicsDrawTriangleFanSet_Z_8
- mlib_GraphicsDrawTriangle_G_32
- mlib_GraphicsDrawTriangle_G_8
- mlib_GraphicsDrawTriangle_GZ_32
- mlib_GraphicsDrawTriangle_GZ_8
- mlib_GraphicsDrawTriangleSet_32
- mlib_GraphicsDrawTriangleSet_8
- mlib_GraphicsDrawTriangleSet_A_32
- mlib_GraphicsDrawTriangleSet_A_8
- mlib_GraphicsDrawTriangleSet_AB_32
- mlib_GraphicsDrawTriangleSet_AB_8
- mlib_GraphicsDrawTriangleSet_ABG_32
- mlib_GraphicsDrawTriangleSet_ABG_8
- mlib_GraphicsDrawTriangleSet_ABGZ_32
- mlib_GraphicsDrawTriangleSet_ABGZ_8
- mlib_GraphicsDrawTriangleSet_ABZ_32
- mlib_GraphicsDrawTriangleSet_ABZ_8
- mlib_GraphicsDrawTriangleSet_G_32
- mlib_GraphicsDrawTriangleSet_G_8
- mlib_GraphicsDrawTriangleSet_GZ_32
- mlib_GraphicsDrawTriangleSet_GZ_8
- mlib_GraphicsDrawTriangleSet_X_32
- mlib_GraphicsDrawTriangleSet_X_8
- mlib_GraphicsDrawTriangleSet_Z_32
- mlib_GraphicsDrawTriangleSet_Z_8
- mlib_GraphicsDrawTriangleStripSet_32
- mlib_GraphicsDrawTriangleStripSet_8
- mlib_GraphicsDrawTriangleStripSet_A_32
- mlib_GraphicsDrawTriangleStripSet_A_8
- mlib_GraphicsDrawTriangleStripSet_AB_32
- mlib_GraphicsDrawTriangleStripSet_AB_8
- mlib_GraphicsDrawTriangleStripSet_ABG_32
- mlib_GraphicsDrawTriangleStripSet_ABG_8
- mlib_GraphicsDrawTriangleStripSet_ABGZ_32
- mlib_GraphicsDrawTriangleStripSet_ABGZ_8
- mlib_GraphicsDrawTriangleStripSet_ABZ_32
- mlib_GraphicsDrawTriangleStripSet_ABZ_8
- mlib_GraphicsDrawTriangleStripSet_AG_32
- mlib_GraphicsDrawTriangleStripSet_AG_8
- mlib_GraphicsDrawTriangleStripSet_AGZ_32
- mlib_GraphicsDrawTriangleStripSet_AGZ_8
- mlib_GraphicsDrawTriangleStripSet_AZ_32
- mlib_GraphicsDrawTriangleStripSet_AZ_8
- mlib_GraphicsDrawTriangleStripSet_B_32
- mlib_GraphicsDrawTriangleStripSet_B_8
- mlib_GraphicsDrawTriangleStripSet_BG_32
- mlib_GraphicsDrawTriangleStripSet_BG_8
- mlib_GraphicsDrawTriangleStripSet_BGZ_32
- mlib_GraphicsDrawTriangleStripSet_BGZ_8
- mlib_GraphicsDrawTriangleStripSet_BZ_32
- mlib_GraphicsDrawTriangleStripSet_BZ_8
- mlib_GraphicsDrawTriangleStripSet_G_32
- mlib_GraphicsDrawTriangleStripSet_G_8
- mlib_GraphicsDrawTriangleStripSet_GZ_32
- mlib_GraphicsDrawTriangleStripSet_GZ_8
- mlib_GraphicsDrawTriangleStripSet_X_32
- mlib_GraphicsDrawTriangleStripSet_X_8
- mlib_GraphicsDrawTriangleStripSet_Z_32
- mlib_GraphicsDrawTriangleStripSet_Z_8
- mlib_GraphicsDrawTriangle_X_32
- mlib_GraphicsDrawTriangle_X_8
- mlib_GraphicsDrawTriangle_Z_32
- mlib_GraphicsFillArc_32
- mlib_GraphicsFillArc_8
- mlib_GraphicsFillArc_A_32
- mlib_GraphicsFillArc_A_8
libmlib(3LIB)

- mlib_GraphicsFillArc_AB_32
- mlib_GraphicsFillArc_AB_8
- mlib_GraphicsFillArc_B_32
- mlib_GraphicsFillArc_B_8
- mlib_GraphicsFillArc_X_32
- mlib_GraphicsFillArc_X_8
- mlib_GraphicsFillCircle_32
- mlib_GraphicsFillCircle_8
- mlib_GraphicsFillCircle_A_32
- mlib_GraphicsFillCircle_A_8
- mlib_GraphicsFillCircle_AB_32
- mlib_GraphicsFillCircle_AB_8
- mlib_GraphicsFillCircle_B_32
- mlib_GraphicsFillCircle_B_8
- mlib_GraphicsFillCircle_X_32
- mlib_GraphicsFillCircle_X_8
- mlib_GraphicsFillEllipse_32
- mlib_GraphicsFillEllipse_8
- mlib_GraphicsFillEllipse_A_32
- mlib_GraphicsFillEllipse_A_8
- mlib_GraphicsFillEllipse_AB_32
- mlib_GraphicsFillEllipse_AB_8
- mlib_GraphicsFillEllipse_B_32
- mlib_GraphicsFillEllipse_B_8
- mlib_GraphicsFillEllipse_X_32
- mlib_GraphicsFillEllipse_X_8
- mlib_GraphicsFillPolygon_32
- mlib_GraphicsFillPolygon_8
- mlib_GraphicsFillPolygon_A_32
- mlib_GraphicsFillPolygon_A_8
- mlib_GraphicsFillPolygon_AB_32
- mlib_GraphicsFillPolygon_AB_8
- mlib_GraphicsFillPolygon_ABG_32
- mlib_GraphicsFillPolygon_ABG_8
- mlib_GraphicsFillPolygon_ABGZ_32
- mlib_GraphicsFillPolygon_ABGZ_8
- mlib_GraphicsFillPolygon_ABZ_32
- mlib_GraphicsFillPolygon_ABZ_8
- mlib_GraphicsFillPolygon_AG_32
- mlib_GraphicsFillPolygon_AG_8
- mlib_GraphicsFillPolygon_AGZ_32
- mlib_GraphicsFillPolygon_AGZ_8
- mlib_GraphicsFillPolygon_AZ_32
- mlib_GraphicsFillPolygon_AZ_8
- mlib_GraphicsFillPolygon_B_32
- mlib_GraphicsFillPolygon_B_8
- mlib_GraphicsFillPolygon_BG_32
- mlib_GraphicsFillPolygon_BG_8
- mlib_GraphicsFillPolygon_BGZ_32
- mlib_GraphicsFillPolygon_BGZ_8
- mlib_GraphicsFillPolygon_BZ_32
- mlib_GraphicsFillPolygon_BZ_8
- mlib_GraphicsFillPolygon_G_32
- mlib_GraphicsFillPolygon_G_8
- mlib_GraphicsFillPolygon_GZ_32
- mlib_GraphicsFillPolygon_GZ_8
- mlib_GraphicsFillPolygon_X_32
- mlib_GraphicsFillPolygon_X_8
- mlib_GraphicsFillPolygon_X_32
- mlib_GraphicsFillPolygon_X_8
- mlib_GraphicsFillRectangle_32
- mlib_GraphicsFillRectangle_8
- mlib_GraphicsFillRectangle_B_32
- mlib_GraphicsFillRectangle_B_8
- mlib_GraphicsFillRectangle_X_32
- mlib_GraphicsFillRectangle_X_8
- mlib_GraphicsFillTriangle_32
- mlib_GraphicsFillTriangle_8
- mlib_GraphicsFillTriangle_A_32
- mlib_GraphicsFillTriangle_A_8
- mlib_GraphicsFillTriangle_AB_32
- mlib_GraphicsFillTriangle_AB_8
- mlib_GraphicsFillTriangle_ABG_32
- mlib_GraphicsFillTriangle_ABG_8
- mlib_GraphicsFillTriangle_ABGZ_32
- mlib_GraphicsFillTriangle_ABGZ_8
- mlib_GraphicsFillTriangle_ABZ_32
- mlib_GraphicsFillTriangle_ABZ_8
- mlib_GraphicsFillTriangle_AG_32
- mlib_GraphicsFillTriangle_AG_8
- mlib_GraphicsFillTriangle_AGZ_32
- mlib_GraphicsFillTriangle_AGZ_8
- mlib_GraphicsFillTriangle_AZ_32
- mlib_GraphicsFillTriangle_AZ_8
- mlib_GraphicsFillTriangle_B_32
- mlib_GraphicsFillTriangle_B_8
- mlib_GraphicsFillTriangle_BG_32
- mlib_GraphicsFillTriangle_BG_8
- mlib_GraphicsFillTriangle_BGZ_32
- mlib_GraphicsFillTriangle_BGZ_8
- mlib_GraphicsFillTriangle_BZ_32
- mlib_GraphicsFillTriangle_BZ_8
- mlib_GraphicsFillTriangleFanSet_32
- mlib_GraphicsFillTriangleFanSet_8
- mlib_GraphicsFillTriangleFanSet_A_32
- mlib_GraphicsFillTriangleFanSet_A_8
- mlib_GraphicsFillTriangleFanSet_AB_32
- mlib_GraphicsFillTriangleFanSet_AB_8
- mlib_GraphicsFillTriangleFanSet_ABG_32
- mlib_GraphicsFillTriangleFanSet_ABG_8
- mlib_GraphicsFillTriangleFanSet_ABGZ_32
- mlib_GraphicsFillTriangleFanSet_ABGZ_8
- mlib_GraphicsFillTriangleFanSet_ABZ_32
- mlib_GraphicsFillTriangleFanSet_ABZ_8
- mlib_GraphicsFillTriangleFanSet_AG_32
- mlib_GraphicsFillTriangleFanSet_AG_8
- mlib_GraphicsFillTriangleFanSet_AGZ_32
- mlib_GraphicsFillTriangleFanSet_AGZ_8
- mlib_GraphicsFillTriangleFanSet_AZ_32
- mlib_GraphicsFillTriangleFanSet_AZ_8
- mlib_GraphicsFillTriangleFanSet_B_32
- mlib_GraphicsFillTriangleFanSet_B_8
- mlib_GraphicsFillTriangleFanSet_BG_32
- mlib_GraphicsFillTriangleFanSet_BG_8
- mlib_GraphicsFillTriangleFanSet_BGZ_32
- mlib_GraphicsFillTriangleFanSet_BGZ_8
- mlib_GraphicsFillTriangleFanSet_BZ_32
- mlib_GraphicsFillTriangleFanSet_BZ_8
- mlib_GraphicsFillTriangleFanSet_G_32
- mlib_GraphicsFillTriangleFanSet_G_8
- mlib_GraphicsFillTriangleFanSet_GZ_32
- mlib_GraphicsFillTriangleFanSet_GZ_8
- mlib_GraphicsFillTriangleFanSet_X_32
- mlib_GraphicsFillTriangleFanSet_X_8
- mlib_GraphicsFillTriangleFanSet_Z_32
- mlib_GraphicsFillTriangleFanSet_Z_8
- mlib_GraphicsFillTriangle_G_32
- mlib_GraphicsFillTriangle_GZ_32
- mlib_GraphicsFillTriangle_GZ_8
- mlib_GraphicsFillTriangleSet_32
- mlib_GraphicsFillTriangleSet_8
- mlib_GraphicsFillTriangleSet_A_32
- mlib_GraphicsFillTriangleSet_A_8
- mlib_GraphicsFillTriangleSet_AB_32
- mlib_GraphicsFillTriangleSet_AB_8
- mlib_GraphicsFillTriangleSet_ABG_32
- mlib_GraphicsFillTriangleSet_ABG_8
- mlib_GraphicsFillTriangleSet_ABGZ_32
- mlib_GraphicsFillTriangleSet_ABGZ_8
- mlib_GraphicsFillTriangleSet_ABZ_32
- mlib_GraphicsFillTriangleSet_ABZ_8
- mlib_GraphicsFillTriangleSet_AG_32
- mlib_GraphicsFillTriangleSet_AG_8
- mlib_GraphicsFillTriangleSet_AGZ_32
- mlib_GraphicsFillTriangleSet_AGZ_8
- mlib_GraphicsFillTriangleSet_AZ_32
- mlib_GraphicsFillTriangleSet_AZ_8
- mlib_GraphicsFillTriangleSet_B_32
- mlib_GraphicsFillTriangleSet_B_8
- mlib_GraphicsFillTriangleSet_BG_32
- mlib_GraphicsFillTriangleSet_BG_8
- mlib_GraphicsFillTriangleSet_BGZ_32
- mlib_GraphicsFillTriangleSet_BGZ_8
- mlib_GraphicsFillTriangleSet_BZ_32
- mlib_GraphicsFillTriangleSet_BZ_8
- mlib_GraphicsFillTriangleSet_G_32
- mlib_GraphicsFillTriangleSet_G_8
- mlib_GraphicsFillTriangleSet_GZ_32
- mlib_GraphicsFillTriangleSet_GZ_8
- mlib_GraphicsFillTriangleSet_X_32
- mlib_GraphicsFillTriangleSet_X_8
- mlib_GraphicsFillTriangleSet_Z_32
- mlib_GraphicsFillTriangleSet_Z_8
- mlib_GraphicsFillTriangleStripSet_32
- mlib_GraphicsFillTriangleStripSet_8
- mlib_GraphicsFillTriangleStripSet_A_32
- mlib_GraphicsFillTriangleStripSet_A_8
- mlib_GraphicsFillTriangleStripSet_AB_32
- mlib_GraphicsFillTriangleStripSet_AB_8
- mlib_GraphicsFillTriangleStripSet_ABG_32
- mlib_GraphicsFillTriangleStripSet_ABG_8
- mlib_GraphicsFillTriangleStripSet_ABGZ_32
- mlib_GraphicsFillTriangleStripSet_ABGZ_8
- mlib_GraphicsFillTriangleStripSet_ABZ_32
- mlib_GraphicsFillTriangleStripSet_ABZ_8
- mlib_GraphicsFillTriangleStripSet_AG_32
- mlib_GraphicsFillTriangleStripSet_AG_8
- mlib_GraphicsFillTriangleStripSet_AGZ_32
- mlib_GraphicsFillTriangleStripSet_AGZ_8
- mlib_GraphicsFillTriangleStripSet_AZ_32
- mlib_GraphicsFillTriangleStripSet_AZ_8
- mlib_GraphicsFillTriangleStripSet_B_32
- mlib_GraphicsFillTriangleStripSet_B_8
- mlib_GraphicsFillTriangleStripSet_BG_32
- mlib_GraphicsFillTriangleStripSet_BG_8
- mlib_GraphicsFillTriangleStripSet_BGZ_32
- mlib_GraphicsFillTriangleStripSet_BGZ_8
- mlib_GraphicsFillTriangleStripSet_BZ_32
- mlib_GraphicsFillTriangleStripSet_BZ_8
- mlib_GraphicsFillTriangleStripSet_G_32
- mlib_GraphicsFillTriangleStripSet_G_8
- mlib_GraphicsFillTriangleStripSet_GZ_32
- mlib_GraphicsFillTriangleStripSet_GZ_8
- mlib_GraphicsFillTriangleStripSet_X_32
- mlib_GraphicsFillTriangleStripSet_X_8
- mlib_GraphicsFillTriangleStripSet_Z_32
- mlib_GraphicsFillTriangleStripSet_Z_8
libmlib(3LIB)

- mlib_GraphicsFillTriangleStripSet_AG_32
- mlib_GraphicsFillTriangleStripSet_AG_8
- mlib_GraphicsFillTriangleStripSet_AGZ_32
- mlib_GraphicsFillTriangleStripSet_AGZ_8
- mlib_GraphicsFillTriangleStripSet_AZ_32
- mlib_GraphicsFillTriangleStripSet_AZ_8
- mlib_GraphicsFillTriangleStripSet_B_32
- mlib_GraphicsFillTriangleStripSet_B_8
- mlib_GraphicsFillTriangleStripSet_BG_32
- mlib_GraphicsFillTriangleStripSet_BG_8
- mlib_GraphicsFillTriangleStripSet_BGZ_32
- mlib_GraphicsFillTriangleStripSet_BGZ_8
- mlib_GraphicsFillTriangleStripSet_BZ_32
- mlib_GraphicsFillTriangleStripSet_BZ_8
- mlib_GraphicsFillTriangleStripSet_G_32
- mlib_GraphicsFillTriangleStripSet_G_8
- mlib_GraphicsFillTriangleStripSet_GZ_32
- mlib_GraphicsFillTriangleStripSet_GZ_8
- mlib_GraphicsFillTriangle_X_32
- mlib_GraphicsFillTriangle_X_8
- mlib_GraphicsFillTriangle_Z_32
- mlib_GraphicsFillTriangle_Z_8
- mlib_GraphicsFloodFill_32
- mlib_GraphicsFloodFill_8

Imaging Functions
- mlib_ImageAbs
- mlib_ImageAbs_Fp
- mlib_ImageAbs_Fp_Inp
- mlib_ImageAbs_Inp
- mlib_ImageAdd
- mlib_ImageAdd_Fp
- mlib_ImageAdd_Fp_Inp
- mlib_ImageAdd_Inp
- mlib_ImageAffine
- mlib_ImageAffine_Fp
- mlib_ImageAffineIndex
- mlib_ImageAffineTable
- mlib_ImageAffineTable_Fp
- mlib_ImageAffineTransform
- mlib_ImageAffineTransform_Fp
- mlib_ImageAffineTransformIndex
- mlib_ImageAnd
- mlib_ImageAnd_Inp
- mlib_ImageAndNot
- mlib_ImageAndNot1_Inp
- mlib_ImageAndNot2_Inp
- mlib_ImageAutoCorrel
- mlib_ImageAutoCorrel_Fp
- mlib_ImageAve
- mlib_ImageAve_Fp
- mlib_ImageAve_Fp_Inp
- mlib_ImageAve_Inp
- mlib_ImageBlend
- mlib_ImageBlend1_Fp_Inp
- mlib_ImageBlend1_Inp
- mlib_ImageBlend2_Fp_Inp
- mlib_ImageBlend2_Inp
- mlib_ImageBlendColor
- mlib_ImageBlendColor_Fp
- mlib_ImageBlendColor_Fp_Inp
- mlib_ImageBlendColor_Inp
- mlib_ImageBlend_DA_DA
- mlib_ImageBlend_DA_DA_Inp
- mlib_ImageBlend_DA_DC
- mlib_ImageBlend_DA_DC_Inp
- mlib_ImageBlend_DA_OMDA
- mlib_ImageBlend_DA_OMDA_Inp
- mlib_ImageBlend_DA_OMDC
- mlib_ImageBlend_DA_OMDC_Inp
- mlib_ImageBlend_DA_OMSA
- mlib_ImageBlend_DA_OMSA_Inp
- mlib_ImageBlend_DA_ONE
- mlib_ImageBlend_DA_ONE_Inp
- mlib_ImageBlend_DA_SA
- mlib_ImageBlend_DA_SA_Inp
- mlib_ImageBlend_DA_SAS
- mlib_ImageBlend_DA_SAS_Inp
- mlib_ImageBlend_DA_ZERO
- mlib_ImageBlend_DA_ZERO_Inp
- mlib_ImageBlend_Fp
- mlib_ImageBlendMulti
- mlib_ImageBlendMulti_Fp
- mlib_ImageBlend_OMDA_DA
- mlib_ImageBlend_OMDA_DA_Inp
- mlib_ImageBlend_OMDA_DC
- mlib_ImageBlend_OMDA_DC_Inp
- mlib_ImageBlend_OMDA_OMDA
- mlib_ImageBlend_OMDA_OMDA_Inp
- mlib_ImageBlend_OMDA_OMDC
- mlib_ImageBlend_OMDA_OMDC_Inp
- mlib_ImageBlend_OMDA_OMSA
- mlib_ImageBlend_OMDA_OMSA_Inp
- mlib_ImageBlend_OMDA_ONE
- mlib_ImageBlend_OMDA_ONE_Inp
- mlib_ImageBlend_OMDA_SA
- mlib_ImageBlend_OMDA_SA_Inp
- mlib_ImageBlend_OMDA_SAS
- mlib_ImageBlend_OMDA_SAS_Inp
- mlib_ImageBlend_OMDA_ZERO
- mlib_ImageBlend_OMDA_ZERO_Inp
- mlib_ImageBlend_OMSA_DA
- mlib_ImageBlend_OMSA_DA_Inp
- mlib_ImageBlend_OMSA_DC
- mlib_ImageBlend_OMSA_DC_Inp
- mlib_ImageBlend_OMSA_OMDA
- mlib_ImageBlend_OMSA_OMDA_Inp
- mlib_ImageBlend_OMSA_OMDC
- mlib_ImageBlend_OMSA_OMDC_Inp
- mlib_ImageBlend_OMSA_OMSA
- mlib_ImageBlend_OMSA_OMSA_Inp
- mlib_ImageBlend_OMSA_ONE
- mlib_ImageBlend_OMSA_ONE_Inp
- mlib_ImageBlend_OMSA_SA
- mlib_ImageBlend_OMSA_SA_Inp
- mlib_ImageBlend_OMSA_SAS
- mlib_ImageBlend_OMSA_SAS_Inp
- mlib_ImageBlend_OMSA_ZERO
- mlib_ImageBlend_OMSA_ZERO_Inp
- mlib_ImageBlend_OMSC_DA
- mlib_ImageBlend_OMSC_DA_Inp
- mlib_ImageBlend_OMSC_DC
- mlib_ImageBlend_OMSC_DC_Inp
- mlib_ImageBlend_OMSC_OMDA
- mlib_ImageBlend_OMSC_OMDA_Inp
- mlib_ImageBlend_OMSC_OMDC
- mlib_ImageBlend_OMSC_OMDC_Inp
- mlib_ImageBlend_OMSC_OMSA
- mlib_ImageBlend_OMSC_OMSA_Inp
- mlib_ImageBlend_OMSC_ONE
- `mlib_ImageBlend_OMSC_ONE_Inp`
- `mlib_ImageBlend_OMSC_SA`
- `mlib_ImageBlend_OMSC_SA_Inp`
- `mlib_ImageBlend_OMSC_SAS`
- `mlib_ImageBlend_OMSC_SAS_Inp`
- `mlib_ImageBlend_OMSC_ZERO`
- `mlib_ImageBlend_OMSC_ZERO_Inp`
- `mlib_ImageBlend_ONE_DA`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_DA_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_DC`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_DC_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_OMDA`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_OMDA_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_OMDC`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_OMDC_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_OMSA`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_OMSA_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_ONE`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_ONE_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_SA`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_SA_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_SAS`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_SAS_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_ZERO`  # Syntaxhighlighted
- `mlib_ImageBlend_ONE_ZERO_Inp`  # Syntaxhighlighted
- `mlib_ImageBlendRGBA2ARGB`
- `mlib_ImageBlendRGBA2BGRA`
- `mlib_ImageBlend_SA_DA`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_DA_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_DC`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_DC_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_OMDA`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_OMDA_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_OMDC`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_OMDC_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_OMSA`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_OMSA_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_ONE`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_ONE_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_SA`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_SA_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_SAS`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_SAS_Inp`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_ZERO`  # Syntaxhighlighted
- `mlib_ImageBlend_SA_ZERO_Inp`  # Syntaxhighlighted
libmlib(3LIB)

- mlib_ImageBlend_SA_ZERO_Inp
- mlib_ImageBlend_SC_DA
- mlib_ImageBlend_SC_DA_Inp
- mlib_ImageBlend_SC_DC
- mlib_ImageBlend_SC_DC_Inp
- mlib_ImageBlend_SC_OMDA
- mlib_ImageBlend_SC_OMDA_Inp
- mlib_ImageBlend_SC_OMDC
- mlib_ImageBlend_SC_OMDC_Inp
- mlib_ImageBlend_SC_OMSA
- mlib_ImageBlend_SC_OMSA_Inp
- mlib_ImageBlend_SC_ONE
- mlib_ImageBlend_SC_ONE_Inp
- mlib_ImageBlend_SC_SA
- mlib_ImageBlend_SC_SA_Inp
- mlib_ImageBlend_SC_SAS
- mlib_ImageBlend_SC_SAS_Inp
- mlib_ImageBlend_SC_ZERO
- mlib_ImageBlend_SC_ZERO_Inp
- mlib_ImageBlend_ZERO_DA
- mlib_ImageBlend_ZERO_DA_Inp
- mlib_ImageBlend_ZERO_DC
- mlib_ImageBlend_ZERO_DC_Inp
- mlib_ImageBlend_ZERO_OMDA
- mlib_ImageBlend_ZERO_OMDA_Inp
- mlib_ImageBlend_ZERO_OMDC
- mlib_ImageBlend_ZERO_OMDC_Inp
- mlib_ImageBlend_ZERO_OMSA
- mlib_ImageBlend_ZERO_OMSA_Inp
- mlib_ImageBlend_ZERO_ONE
- mlib_ImageBlend_ZERO_ONE_Inp
- mlib_ImageBlend_ZERO_SA
- mlib_ImageBlend_ZERO_SA_Inp
- mlib_ImageBlend_ZERO_SAS
- mlib_ImageBlend_ZERO_SAS_Inp
- mlib_ImageBlend_ZERO_ZERO
- mlib_ImageBlend_ZERO_ZERO_Inp
- mlib_ImageChannelCopy
- mlib_ImageChannelExtract
- mlib_ImageChannelInsert
- mlib_ImageChannelMerge
- mlib_ImageChannelSplit
- mlib_ImageClear
- mlib_ImageClearEdge
- mlib_ImageClearEdge_Fp
- mlib_ImageClear_Fp
- mlib_ImageColorConvert1
- mlib_ImageColorConvert1_Fp
- mlib_ImageColorConvert2
- mlib_ImageColorConvert2_Fp
- mlib_ImageColorDitherFree
- mlib_ImageColorDitherInit
- mlib_ImageColorErrorDiffusion3x3
- mlib_ImageColorErrorDiffusionMxN
- mlib_ImageColorHSL2RGB
- mlib_ImageColorHSL2RGB_Fp
- mlib_ImageColorHSV2RGB
- mlib_ImageColorHSV2RGB_Fp
- mlib_ImageColorOrderedDither8x8
- mlib_ImageColorOrderedDitherMxN
- mlib_ImageColorRGB2CIEMono
- mlib_ImageColorRGB2CIEMono_Fp
- mlib_ImageColorRGB2HSL
- mlib_ImageColorRGB2HSL_Fp
- mlib_ImageColorRGB2HSV
- mlib_ImageColorRGB2HSV_Fp
- mlib_ImageColorRGB2Mono
- mlib_ImageColorRGB2Mono_Fp
- mlib_ImageColorRGB2XYZ
- mlib_ImageColorRGB2XYZ_Fp
- mlib_ImageColorRGB2YCC
- mlib_ImageColorRGB2YCC_Fp
- mlib_ImageColorTrue2Index
- mlib_ImageColorTrue2IndexFree
- mlib_ImageColorTrue2IndexInit
- mlib_ImageColorXYZ2RGB
- mlib_ImageColorXYZ2RGB_Fp
- mlib_ImageColorYCC2RGB
- mlib_ImageColorYCC2RGB_Fp
- mlib_ImageComposite
- mlib_ImageComposite_Inp
- mlib_ImageConstAdd
- mlib_ImageConstAdd_Fp
- mlib_ImageConstAdd_Fp_Inp
- mlib_ImageConstAdd_Inp
- mlib_ImageConstAnd
- mlib_ImageConstAnd_Inp
- mlib_ImageConstAndNot
- mlib_ImageConstAndNot_Inp
- mlib_ImageConstDiv
- mlib_ImageConstDiv_Fp
- mlib_ImageConstDiv_Fp_Inp
- mlib_ImageConstDiv_Inp
- mlib_ImageConstDivShift
- mlib_ImageConstDivShift_Inp
- mlib_ImageConstMul
- mlib_ImageConstMul_Fp
- mlib_ImageConstMul_Fp_Inp
- mlib_ImageConstMul_Inp
- mlib_ImageConstMulShift
- mlib_ImageConstMulShift_Inp
- mlib_ImageConstNotAnd
- mlib_ImageConstNotAnd_Inp
- mlib_ImageConstNotOr
- mlib_ImageConstNotOr_Inp
- mlib_ImageConstNotXor
- mlib_ImageConstNotXor_Inp
- mlib_ImageConstOr
- mlib_ImageConstOr_Inp
- mlib_ImageConstOrNot
- mlib_ImageConstOrNot_Inp
- mlib_ImageConstSub
- mlib_ImageConstSub_Fp
- mlib_ImageConstSub_Fp_Inp
- mlib_ImageConstSub_Inp
- mlib_ImageConstXor
- mlib_ImageConstXor_Inp
- mlib_ImageConv2x2
- mlib_ImageConv2x2_Fp
- mlib_ImageConv2x2Index
- mlib_ImageConv3x3
- mlib_ImageConv3x3_Fp
- mlib_ImageConv3x3Index
- mlib_ImageConv4x4
- mlib_ImageConv4x4_Fp
- mlib_ImageConv4x4Index
- mlib_ImageConv5x5
- mlib_ImageConv5x5_Fp
- mlib_ImageConv5x5Index
- mlib_ImageConv7x7
- mlib_ImageConv7x7_Fp
- mlib_ImageConv7x7Index
- mlib_ImageConvKernelConvert
- mlib_ImageConvMxN
- mlib_ImageConvMxN_Fp
- mlib_ImageConvMxNIndex
- mlib_ImageConvolveMxN
- mlib_ImageConvolveMxN_Fp
- mlib_ImageCopy
- mlib_ImageCopyArea
- mlib_ImageCopyMask
- mlib_ImageCopyMask_Fp
- mlib_ImageCopySubimage
- mlib_ImageCreate
- mlib_ImageCreateStruct
- mlib_ImageCreateSubimage
- mlib_ImageCrossCorrel
- mlib_ImageCrossCorrel_Fp
- mlib_ImageDataTypeConvert
- mlib_ImageDelete
- mlib_ImageDilate4
- mlib_ImageDilate4_Fp
- mlib_ImageDilate8
- mlib_ImageDilate8_Fp
- mlib_ImageDiv1_Fp_Inp
- mlib_ImageDiv2_Fp_Inp
- mlib_ImageDivAlpha
- mlib_ImageDivAlpha_Fp
- mlib_ImageDivAlpha_Fp_Inp
- mlib_ImageDivAlpha_Inp
- mlib_ImageDivConstShift
- mlib_ImageDivConstShift_Inp
- mlib_ImageDiv_Fp
- mlib_ImageDivShift
- mlib_ImageDivShift1_Inp
- mlib_ImageDivShift2_Inp
- mlib_ImageErode4
- mlib_ImageErode4_Fp
- mlib_ImageErode8
- mlib_ImageErode8_Fp
- mlib_ImageExp
- mlib_ImageExp_Fp
- mlib_ImageExp_Fp_Inp
- mlib_ImageExp_Inp
- mlib_ImageExtrema2
- mlib_ImageExtrema2_Fp
- mlib_ImageExtremaLocations
- mlib_ImageExtremaLocations_Fp
- mlib_ImageFilteredSubsample
- mlib_ImageFilteredSubsample_Fp
- mlib_ImageFlipAntiDiag
- mlib_ImageFlipAntiDiag_Fp
- mlib_ImageFlipMainDiag
- mlib_ImageFlipMainDiag_Fp
- mlib_ImageFlipX
- mlib_ImageFlipX_Fp
- mlib_ImageFlipY
- mlib_ImageFlipY_Fp
- mlib_ImageFourierTransform
- mlib_ImageGetBitOffset
- mlib_ImageGetChannels
- mlib_ImageGetData
- mlib_ImageGetFlags
- mlib_ImageGetFormat
- mlib_ImageGetHeight
- mlib_ImageGetPaddings
- mlib_ImageGetStride
- mlib_ImageGetType
- mlib_ImageGetWidth
- mlib_ImageGradient3x3
- mlib_ImageGradient3x3_Fp
- mlib_ImageGradientMxN
- mlib_ImageGradientMxN_Fp
- mlib_ImageGridWarp
- mlib_ImageGridWarp_Fp
- mlib_ImageGridWarpTable
- mlib_ImageGridWarpTable_Fp
- mlib_ImageHistogram
- mlib_ImageHistogram2
- mlib_ImageInterpTableCreate
- mlib_ImageInterpTableDelete
- mlib_ImageInvert
- mlib_ImageInvert_Fp
- mlib_ImageInvert_Fp_Inp
- mlib_ImageInvert_Inp
- mlib_ImagesNotAligned2
- mlib_ImagesNotAligned4
- mlib_ImagesNotAligned64
- mlib_ImagesNotAligned8
- mlib_ImagesNotHeight2X
- mlib_ImageIsNotHeight4X
- mlib_ImageIsNotHeight8X
- mlib_ImageIsNotOneDvector
- mlib_ImageIsNotStride8X
- mlib_ImageIsNotWidth2X
- mlib_ImageIsNotWidth4X
- mlib_ImageIsNotWidth8X
- mlib_ImageIsUserAllocated
- mlib_ImageLog
- mlib_ImageLog_Fp
- mlib_ImageLog_Fp_Inp
- mlib_ImageLog_Inp
- mlib_ImageLookUp
- mlib_ImageLookUp2
- mlib_ImageLookUp_Inp
- mlib_ImageLookUpMask
- mlib_ImageMax
- mlib_ImageMaxFilter3x3
- mlib_ImageMaxFilter3x3_Fp
- mlib_ImageMaxFilter5x5
- mlib_ImageMaxFilter5x5_Fp
- mlib_ImageMaxFilter7x7
- mlib_ImageMaxFilter7x7_Fp
- mlib_ImageMaxFilter7x7_US
- mlib_ImageMedianFilter3x3
- mlib_ImageMedianFilter3x3_Fp
- mlib_ImageMedianFilter3x3_US
- mlib_ImageMedianFilter5x5
- mlib_ImageMedianFilter5x5_Fp
- mlib_ImageMedianFilter5x5_US
- mlib_ImageMedianFilter7x7
- mlib_ImageMedianFilter7x7_Fp
- mlib_ImageMedianFilter7x7_US
- mlib_ImageMedianFilterMxN
- mlib_ImageMedianFilterMxN_Fp
- mlib_ImageMedianFilterMxN_US
- mlib_ImageMin
- mlib_ImageMinFilter3x3
- mlib_ImageMinFilter3x3_Fp
- mlib_ImageMinFilter5x5
- mlib_ImageMinFilter5x5_Fp
- mlib_ImageMinFilter7x7
- mlib_ImageMinFilter7x7_Fp
- mlib_ImageMin_Fp
- mlib_ImageMin_Fp_Inp
- mlib_ImageMinimum
- mlib_ImageMinimum_Fp
- mlib_ImageMin_Inp
- mlib_ImageMoment2
- mlib_ImageMoment2_Fp
- mlib_ImageMulAlpha
- mlib_ImageMulAlpha_Fp
- mlib_ImageMulAlpha_Fp_Inp
- mlib_ImageMulAlpha_Inp
- mlib_ImageMul_Fp
- mlib_ImageMul_Fp_Inp
- mlib_ImageMulShift
- mlib_ImageMulShift_Inp
- mlib_ImageNormCrossCorrel
- mlib_ImageNormCrossCorrel_Fp
- mlib_ImageNot
- mlib_ImageNotAnd
- mlib_ImageNotAnd_Inp
- mlib_ImageNot_Inp
- mlib_ImageNotOr
- mlib_ImageNotOr_Inp
- mlib_ImageNotXor
- mlib_ImageNotXor_Inp
- mlib_ImageOr
- mlib_ImageOr_Inp
- mlib_ImageOrNot
- mlib_ImageOrNot1_Inp
- mlib_ImageOrNot2_Inp
- mlib_ImagePolynomialWarp
- mlib_ImagePolynomialWarp_Fp
- mlib_ImagePolynomialWarpTable
- mlib_ImagePolynomialWarpTable_Fp
- mlib_ImageRankFilter3x3
- mlib_ImageRankFilter3x3_Fp
- mlib_ImageRankFilter3x3_US
- mlib_ImageRankFilter5x5
- mlib_ImageRankFilter5x5_Fp
- mlib_ImageRankFilter5x5_US
- mlib_ImageRankFilter7x7
- mlib_ImageRankFilter7x7_Fp
- mlib_ImageRankFilter7x7_US
- mlib_ImageRankFilterMxN
- mlib_ImageRankFilterMxN_Fp
- mlib_ImageRankFilterMxN_US
- mlib_ImageReformat
- mlib_ImageReplaceColor
- mlib_ImageReplaceColor_Fp
- mlib_ImageReplaceColor_Fp_Inp
- mlib_ImageReplaceColor_Inp
- mlib_ImageResetStruct
- mlib_ImageResetSubimageStruct
- mlib_ImageRotate
- mlib_ImageRotate180
- mlib_ImageRotate180_Fp
- mlib_ImageRotate270
- mlib_ImageRotate270_Fp
- mlib_ImageRotate90
- mlib_ImageRotate90_Fp
- mlib_ImageRotate_Fp
- mlib_ImageRotateIndex
- mlib_ImageScalarBlend
- mlib_ImageScalarBlend_Fp
- mlib_ImageScalarBlend_Fp_Inp
- mlib_ImageScalarBlend_Inp
- mlib_ImageScale
- mlib_ImageScale2
- mlib_ImageScale2_Inp
- mlib_ImageScale_Fp
- mlib_ImageScale_Fp_Inp
- mlib_ImageScale_Inp
- mlib_ImageSConv3x3
- mlib_ImageSConv3x3_Fp
- mlib_ImageSConv5x5
- mlib_ImageSConv5x5_Fp
- mlib_ImageSConv7x7
- mlib_ImageSConv7x7_Fp
- mlib_ImageSConvKernelConvert
- mlib_ImageSetFormat
- mlib_ImageSetPaddings
- mlib_ImageSetStruct
- mlib_ImageSetSubimageStruct
- mlib_ImageSobel
- mlib_ImageSobel_Fp
- mlib_ImageSqr_Fp
- mlib_ImageSqr_Fp_Inp
- mlib_ImageSqrShift
- mlib_ImageSqrShift_Inp
- mlib_ImageStdDev
- mlib_ImageStdDev_Fp
- mlib_ImageSub
- mlib_ImageSub1_Fp_Inp
- mlib_ImageSub1_Inp
- mlib_ImageSub2_Fp_Inp
- mlib_ImageSub2_Inp
- mlib_ImageSub_Fp
- mlib_ImageSubsampleAverage
- mlib_ImageSubsampleAverage_Fp
- mlib_ImageSubsampleBinaryToGray
- mlib_ImageTestFlags
- mlib_ImageThresh1
- mlib_ImageThresh1_Fp
- mlib_ImageThresh1_Fp_Inp
- mlib_ImageThresh1_Inp
- mlib_ImageThresh2
- mlib_ImageThresh2_Fp
- mlib_ImageThresh2_Fp_Inp
- mlib_ImageThresh2_Inp
- mlib_ImageThresh3
- mlib_ImageThresh3_Fp
- mlib_ImageThresh3_Fp_Inp
- mlib_ImageThresh3_Inp
- mlib_ImageThresh4
- mlib_ImageThresh4_Fp
- mlib_ImageThresh4_Fp_Inp
- mlib_ImageThresh4_Inp
- mlib_ImageThresh5
- mlib_ImageThresh5_Fp
- mlib_ImageThresh5_Fp_Inp
- mlib_ImageThresh5_Inp
- mlib_ImageXor
- mlib_ImageXor_Inp
- mlib_ImageXProj
- mlib_ImageXProj_Fp
- mlib_ImageYProj
- mlib_ImageYProj_Fp
- mlib_ImageZoom
- mlib_ImageZoomBlend
- mlib_ImageZoom_Fp
- mlib_ImageZoomIn2X
- mlib_ImageZoomIn2X_Fp
- mlib_ImageZoomIn2XIndex
- mlib_ImageZoomIndex
- mlib_ImageZoomOut2X
- mlib_ImageZoomOut2X_Fp
- mlib_ImageZoomOut2XIndex
- mlib_ImageZoomTranslate
- mlib_ImageZoomTranslateBlend
- mlib_ImageZoomTranslate_Fp
- mlib_ImageZoomTranslateTable
- mlib_ImageZoomTranslateTableBlend
- mlib_ImageZoomTranslateTable_Fp
- mlib_ImageZoomTranslateToGray
- mlib_SignalADPCM2Bits2Linear
- mlib_SignalADPCM3Bits2Linear
- mlib_SignalADPCM4Bits2Linear
- mlib_SignalADPCM5Bits2Linear
- mlib_SignalADPCMFree
- mlib_SignalADPCMInit
- mlib_SignalALaw2Linear
- mlib_SignalALaw2uLaw
- mlib_SignalAutoCorrel_F32
- mlib_SignalAutoCorrel_F32S
- mlib_SignalAutoCorrel_S16
- mlib_SignalAutoCorrel_S16S
- mlib_SignalCepstral_F32
- mlib_SignalCepstralFree_F32
- mlib_SignalCepstralFree_S16
- mlib_SignalCepstralInit_F32
- mlib_SignalCepstralInit_S16
- mlib_SignalCepstral_S16
- mlib_SignalCepstral_S16_Adp
- mlib_SignalConvertShift_F32_F32
- mlib_SignalConvertShift_F32_S16
- mlib_SignalConvertShift_F32_S32
- mlib_SignalConvertShift_F32_S8
- mlib_SignalConvertShift_F32S_F32S
- mlib_SignalConvertShift_F32S_S32S
- mlib_SignalConvertShift_F32S_S8S
- mlib_SignalConvertShift_F32S_U8S
- mlib_SignalConvertShift_F32_U8
- mlib_SignalConvertShift_S16_F32_Sat
- mlib_SignalConvertShift_S16_S32_Sat
- mlib_SignalConvertShift_S16_S8_Sat
- mlib_SignalConvertShift_S16S_F32S_Sat
- mlib_SignalConvertShift_S16S_S32S_Sat
- mlib_SignalConvertShift_S16S_S8S_Sat
- mlib_SignalConvertShift_S16S_U8S_Sat
- mlib_SignalConvertShift_S32_F32_Sat
- mlib_SignalConvertShift_S32_S16_Sat
- mlib_SignalConvertShift_S32_S8_Sat
- mlib_SignalConvertShift_S32S_F32S_Sat
- mlib_SignalConvertShift_S32S_S16S_Sat
- mlib_SignalConvertShift_S32S_S8S_Sat
- mlib_SignalConvertShift_S32S_U8S_Sat
- mlib_SignalConvertShift_S8_F32_Sat
- mlib_SignalConvertShift_S8_S16_Sat
- mlib_SignalConvertShift_S8_S32_Sat
- mlib_SignalConvertShift_S8S_F32S_Sat
- mlib_SignalConvertShift_S8S_S16S_Sat
- mlib_SignalConvertShift_S8S_S8S_Sat
- mlib_SignalConvertShift_S8S_U8S_Sat
- mlib_SignalConv_F32_F32
- mlib_SignalConv_F32S_F32S
- mlib_SignalConv_S16_S16_Sat
- mlib_SignalConv_S16S_S16S_Sat
- mlib_SignalCrossCorrel_F32
- mlib_SignalCrossCorrel_F32S
- mlib_SignalCrossCorrel_S16
- mlib_SignalCrossCorrel_S16S
- mlib_SignalDownSample_F32_F32
- mlib_SignalDownSample_F32S_F32S
- mlib_SignalDownSample_S16_S16
- mlib_SignalDownSample_S16S_S16S
- mlib_SignalDTWKScalar_F32
- mlib_SignalDTWKScalarFree_F32
- mlib_SignalDTWKScalarFree_S16
- mlib_SignalDTWKScalarInit_F32
- mlib_SignalDTWKScalarInit_S16
- mlib_SignalDTWKScalarPath_F32
- mlib_SignalDTWKScalarPath_S16
- mlib_SignalDTWKScalar_S16
- mlib_SignalDTWKVector_F32
- mlib_SignalDTWKVectorFree_F32
- mlib_SignalDTWKVectorFree_S16
- mlib_SignalDTWKVectorInit_F32
- mlib_SignalDTWKVectorInit_S16
- mlib_SignalDTWKVectorPath_F32
- mlib_SignalDTWKVectorPath_S16
- mlib_SignalDTWKVector_S16
- mlib_SignalEmphasize_F32_F32
- mlib_SignalEmphasize_F32S_F32S
- mlib_SignalEmphasizeFree_F32_F32
- mlib_SignalEmphasizeFree_S16_S16
- mlib_SignalEmphasizeFree_S16S_S16S
- mlib_SignalEmphasizeInit_F32_F32
- mlib_SignalEmphasizeInit_F32S_F32S
- mlib_SignalEmphasizeInit_S16_S16
- mlib_SignalEmphasizeInit_S16S_S16S
- mlib_SignalEmphasize_S16_S16_Sat
- mlib_SignalEmphasize_S16S_S16S_Sat
mlib_SignalFFT_1_D64
mlib_SignalFFT_1_D64C
mlib_SignalFFT_1_D64C_D64
mlib_SignalFFT_1_D64C_D64C
mlib_SignalFFT_1_D64_D64
mlib_SignalFFT_1_F32
mlib_SignalFFT_1_F32C
mlib_SignalFFT_1_F32C_F32
mlib_SignalFFT_1_F32C_F32C
mlib_SignalFFT_1_F32_F32
mlib_SignalFFT_1_S16C_Mod
mlib_SignalFFT_1_S16C_S16C_Mod
mlib_SignalFFT_1_S16C_S16_Mod
mlib_SignalFFT_1_S16_Mod
mlib_SignalFFT_1_S16_S16_Mod
mlib_SignalFFT_2_D64
mlib_SignalFFT_2_D64C
mlib_SignalFFT_2_D64C_D64
mlib_SignalFFT_2_D64C_D64C
mlib_SignalFFT_2_D64_D64
mlib_SignalFFT_2_F32
mlib_SignalFFT_2_F32C
mlib_SignalFFT_2_F32C_F32
mlib_SignalFFT_2_F32C_F32C
mlib_SignalFFT_2_F32_F32
mlib_SignalFFT_2_S16
mlib_SignalFFT_2_S16C
mlib_SignalFFT_2_S16C_S16
mlib_SignalFFT_2_S16C_S16C
mlib_SignalFFT_2_S16_S16
mlib_SignalFFT_3_D64
mlib_SignalFFT_3_D64C
mlib_SignalFFT_3_D64C_D64
mlib_SignalFFT_3_D64C_D64C
mlib_SignalFFT_3_D64_D64
mlib_SignalFFT_3_F32
mlib_SignalFFT_3_F32C
mlib_SignalFFT_3_F32C_F32
mlib_SignalFFT_3_F32C_F32C
mlib_SignalFFT_3_F32_F32
mlib_SignalFFT_3_S16C_Mod
mlib_SignalFFT_3_S16C_S16C_Mod
mlib_SignalFFT_3_S16C_S16_Mod
mlib_SignalFFT_3_S16_Mod
- mlib_SignalFFT_3_S16_S16_Mod
- mlib_SignalFFT_4_S16
- mlib_SignalFFT_4_S16C
- mlib_SignalFFT_4_S16C_S16
- mlib_SignalFFT_4_S16C_S16C
- mlib_SignalFFT_4_S16_S16
- mlib_SignalFFTW_1_F32
- mlib_SignalFFTW_1_F32C
- mlib_SignalFFTW_1_F32C_F32
- mlib_SignalFFTW_1_F32C_F32C
- mlib_SignalFFTW_1_F32_F32
- mlib_SignalFFTW_1_S16C_Mod
- mlib_SignalFFTW_1_S16C_S16C_Mod
- mlib_SignalFFTW_1_S16C_S16_Mod
- mlib_SignalFFTW_1_S16_Mod
- mlib_SignalFFTW_1_S16_S16_Mod
- mlib_SignalFFTW_2_F32
- mlib_SignalFFTW_2_F32C
- mlib_SignalFFTW_2_F32C_F32
- mlib_SignalFFTW_2_F32C_F32C
- mlib_SignalFFTW_2_F32_F32
- mlib_SignalFFTW_2_S16
- mlib_SignalFFTW_2_S16C
- mlib_SignalFFTW_2_S16C_S16
- mlib_SignalFFTW_2_S16C_S16C
- mlib_SignalFFTW_2_S16_S16
- mlib_SignalFFTW_3_F32
- mlib_SignalFFTW_3_F32C
- mlib_SignalFFTW_3_F32C_F32
- mlib_SignalFFTW_3_F32C_F32C
- mlib_SignalFFTW_3_F32_F32
- mlib_SignalFFTW_3_S16C_Mod
- mlib_SignalFFTW_3_S16C_S16C_Mod
- mlib_SignalFFTW_3_S16C_S16_Mod
- mlib_SignalFFTW_3_S16_Mod
- mlib_SignalFFTW_3_S16_S16_Mod
- mlib_SignalFFTW_4_S16
- mlib_SignalFFTW_4_S16C
- mlib_SignalFFTW_4_S16C_S16
- mlib_SignalFFTW_4_S16C_S16C
- mlib_SignalFFTW_4_S16_S16
- mlib_SignalFIR_F32_F32
- mlib_SignalFIR_F32S_F32S
- mlib_SignalFIRFree_F32_F32
- mlib_SignalFIRFree_F32S_F32S
- mlib_SignalFIRFree_S16_S16
- mlib_SignalFIRFree_S16S_S16S
- mlib_SignalFIRInit_F32_F32
- mlib_SignalFIRInit_F32S_F32S
- mlib_SignalFIRInit_S16_S16
- mlib_SignalFIRInit_S16S_S16S
- mlib_SignalFIR_S16_S16_Sat
- mlib_SignalFIR_S16S_S16S_Sat
- mlib_SignalGaussNoise_F32
- mlib_SignalGaussNoiseFree_F32
- mlib_SignalGaussNoiseFree_S16
- mlib_SignalGaussNoiseInit_F32
- mlib_SignalGaussNoiseInit_S16
- mlib_SignalGenBartlett_F32
- mlib_SignalGenBartlett_S16
- mlib_SignalGenBlackman_F32
- mlib_SignalGenBlackman_S16
- mlib_SignalGenHamming_F32
- mlib_SignalGenHamming_S16
- mlib_SignalGenHanning_F32
- mlib_SignalGenHanning_S16
- mlib_SignalGenKaiser_F32
- mlib_SignalGenKaiser_S16
- mlib_SignalIFFT_1_D64
- mlib_SignalIFFT_1_D64C
- mlib_SignalIFFT_1_D64C_D64C
- mlib_SignalIFFT_1_D64_D64
- mlib_SignalIFFT_1_D64_D64C
- mlib_SignalIFFT_1_F32
- mlib_SignalIFFT_1_F32C
- mlib_SignalIFFT_1_F32C_F32C
- mlib_SignalIFFT_1_F32_F32
- mlib_SignalIFFT_1_F32C_F32C
- mlib_SignalIFFT_1_S16
- mlib_SignalIFFT_1_S16C
- mlib_SignalIFFT_1_S16C_S16C
- mlib_SignalIFFT_1_S16_S16
- mlib_SignalIFFT_1_S16_S16C
- mlib_SignalIFFT_2_D64
- mlib_SignalIFFT_2_D64C
- mlib_SignalIFFT_2_D64C_D64C
- mlib_SignalIFFT_2_D64_D64
- mlib_SignalIFFT_2_D64_D64C
- mlib_SignalIFFT_2_F32
- mlib_SignalIFFT_2_F32C
- mlib_SignalIFFT_2_F32C_F32C
- mlib_SignalIFFT_2_F32_F32
- mlib_SignalIFFT_2_F32_F32C
- mlib_SignalIFFT_2_S16C_Mod
- mlib_SignalIFFT_2_S16C_S16C_Mod
- mlib_SignalIFFT_2_S16_Mod
- mlib_SignalIFFT_2_S16_S16C_Mod
- mlib_SignalIFFT_2_S16_S16_Mod
- mlib_SignalIFFT_3_D64
- mlib_SignalIFFT_3_D64C
- mlib_SignalIFFT_3_D64C_D64C
- mlib_SignalIFFT_3_D64_D64
- mlib_SignalIFFT_3_D64_D64C
- mlib_SignalIFFT_3_F32
- mlib_SignalIFFT_3_F32C
- mlib_SignalIFFT_3_F32C_F32C
- mlib_SignalIFFT_3_F32_F32
- mlib_SignalIFFT_3_F32_F32C
- mlib_SignalIFFT_3_S16C_Mod
- mlib_SignalIFFT_3_S16C_S16C_Mod
- mlib_SignalIFFT_3_S16_Mod
- mlib_SignalIFFT_3_S16_S16C_Mod
- mlib_SignalIFFT_3_S16_S16_Mod
- mlib_SignalIFFT_4_S16
- mlib_SignalIFFT_4_S16C
- mlib_SignalIFFT_4_S16C_S16C
- mlib_SignalIFFT_4_S16_S16
- mlib_SignalIFFT_4_S16_S16C
- mlib_SignalIFFTW_1_F32
- mlib_SignalIFFTW_1_F32C
- mlib_SignalIFFTW_1_F32C_F32C
- mlib_SignalIFFTW_1_F32_F32
- mlib_SignalIFFTW_1_F32_F32C
- mlib_SignalIFFTW_1_S16
- mlib_SignalIFFTW_1_S16C
- mlib_SignalIFFTW_1_S16C_S16C
- mlib_SignalIFFTW_1_S16_S16
- mlib_SignalIFFTW_1_S16_S16C
- mlib_SignalIFFTW_2_F32
- mlib_SignalIFFTW_2_F32C
- mlib_SignalIFFTW_2_F32C_F32C
libmlib(3LIB)

- mlib_SignalIFFTW_2_F32_F32
- mlib_SignalIFFTW_2_F32_F32C
- mlib_SignalIFFTW_2_S16C_Mod
- mlib_SignalIFFTW_2_S16C_S16C_Mod
- mlib_SignalIFFTW_2_S16_Mod
- mlib_SignalIFFTW_2_S16_S16C_Mod
- mlib_SignalIFFTW_2_S16_S16_Mod
- mlib_SignalIFFTW_3_F32
- mlib_SignalIFFTW_3_F32C
- mlib_SignalIFFTW_3_F32C_F32C
- mlib_SignalIFFTW_3_F32_F32
- mlib_SignalIFFTW_3_F32_F32C
- mlib_SignalIFFTW_3_S16C_Mod
- mlib_SignalIFFTW_3_S16C_S16C_Mod
- mlib_SignalIFFTW_3_S16_Mod
- mlib_SignalIFFTW_3_S16_S16C_Mod
- mlib_SignalIFFTW_3_S16_S16_Mod
- mlib_SignalIFFTW_4_S16
- mlib_SignalIFFTW_4_S16C
- mlib_SignalIFFTW_4_S16C_S16C
- mlib_SignalIFFTW_4_S16_S16
- mlib_SignalIFFTW_4_S16_S16C
- mlib_SignalIIR_Biquad_F32_F32
- mlib_SignalIIR_Biquad_F32S_F32S
- mlib_SignalIIR_Biquad_S16_S16_Sat
- mlib_SignalIIR_Biquad_S16S_S16S_Sat
- mlib_SignalIIRFree_Biquad_F32_F32
- mlib_SignalIIRFree_Biquad_F32S_F32S
- mlib_SignalIIRFree_Biquad_S16_S16
- mlib_SignalIIRFree_Biquad_S16S_S16S
- mlib_SignalIIRInit_Biquad_F32_F32
- mlib_SignalIIRInit_Biquad_F32S_F32S
- mlib_SignalIIRInit_Biquad_S16_S16
- mlib_SignalIIRInit_Biquad_S16S_S16S
- mlib_SignalIIRInit_P4_F32_F32
- mlib_SignalIIRInit_P4_F32S_F32S
- mlib_SignalIIRInit_P4_S16_S16
- mlib_SignalIIRInit_P4_S16S_S16S
- mlib_SignalIIR_P4_F32_F32
- mlib_SignalIIR_P4_F32S_F32S
- mlib_SignalIIRFree_P4_F32_F32
- mlib_SignalIIRFree_P4_F32S_F32S
- mlib_SignalIIRFree_P4_S16_S16
- mlib_SignalIIRFree_P4_S16S_S16S
- mlib_SignalIIRInit_Biquad_F32_F32
- mlib_SignalIIRInit_Biquad_F32S_F32S
- mlib_SignalIIRInit_Biquad_S16_S16
- mlib_SignalIIRInit_Biquad_S16S_S16S
- mlib_SignalIIRInit_P4_F32_F32
- mlib_SignalIIRInit_P4_F32S_F32S
- mlib_SignalIIRInit_P4_S16_S16
- mlib_SignalIIRInit_P4_S16S_S16S
- mlib_SignalIIR_P4_F32_F32
- mlib_SignalIIR_P4_F32S_F32S
- mlib_SignalIIR_P4_S16_S16_Sat
- mlib_SignalIIR_P4_S16S_S16S_Sat
- mlib_SignalMDCT_D64
- mlib_SignalMDCT_F32
- mlib_SignalMDCTSplit_D64
- mlib_SignalMDCTSplit_F32
- mlib_SignalLimit_F32
- mlib_SignalLimit_F32_F32
- mlib_SignalLimit_F32S
- mlib_SignalLimit_F32S_F32S
- mlib_SignalLimit_S16
- mlib_SignalLimit_S16S
- mlib_SignalLimit_S16_S16
- mlib_SignalLimit_S16S_S16S
- mlib_SignalLinear2ADPCM2Bits
- mlib_SignalLinear2ADPCM3Bits
- mlib_SignalLinear2ADPCM4Bits
- mlib_SignalLinear2ADPCM5Bits
- mlib_SignalLinear2ALaw
- mlib_SignalLinear2uLaw
- mlib_SignalLMSFilter_F32_F32
- mlib_SignalLMSFilter_F32S_F32S
- mlib_SignalLMSFilterFree_F32_F32
- mlib_SignalLMSFilterFree_F32S_F32S
- mlib_SignalLMSFilterFree_S16_S16
- mlib_SignalLMSFilterFree_S16S_S16S
- mlib_SignalLMSFilterInit_F32_F32
- mlib_SignalLMSFilterInit_F32S_F32S
- mlib_SignalLMSFilterInit_S16_S16
- mlib_SignalLMSFilterInit_S16S_S16S
- mlib_SignalLMSFilterNonAdapt_F32_F32
- mlib_SignalLMSFilterNonAdapt_F32S_F32S
- mlib_SignalLMSFilterNonAdapt_S16_S16_Sat
- mlib_SignalLMSFilterNonAdapt_S16S_S16S_Sat
- mlib_SignalLPC2Cepstral_F32
- mlib_SignalLPC2Cepstral_S16
- mlib_SignalLPC2Cepstral_S16_Adpt
- mlib_SignalLPC2LSP_F32
- mlib_SignalLPC2LSP_S16
- mlib_SignalLPCAuCorrel_F32
- mlib_SignalLPCAuCorrelFree_F32
- mlib_SignalLPCAuCorrelFree_S16
- mlib_SignalLPCAutoCorrelGetEnergy_F32
- mlib_SignalLPCAutoCorrelGetEnergy_S16
- mlib_SignalLPCAutoCorrelGetEnergy_S16_Adp
- mlib_SignalLPCAutoCorrelGetPARCOR_F32
- mlib_SignalLPCAutoCorrelGetPARCOR_S16
- mlib_SignalLPCAutoCorrelGetPARCOR_S16_Adp
- mlib_SignalLPCAutoCorrelInit_F32
- mlib_SignalLPCAutoCorrelInit_S16
- mlib_SignalLPCAutoCorrel_S16
- mlib_SignalLPCAutoCorrel_S16_Adp
- mlib_SignalLPCCovariance_F32
- mlib_SignalLPCCovarianceFree_F32
- mlib_SignalLPCCovarianceFree_S16
- mlib_SignalLPCCovarianceInit_F32
- mlib_SignalLPCCovarianceInit_S16
- mlib_SignalLPCCovariance_S16
- mlib_SignalLPCCovariance_S16_Adp
- mlib_SignalLPCPerceptWeight_F32
- mlib_SignalLPCPerceptWeightFree_F32
- mlib_SignalLPCPerceptWeightFree_S16
- mlib_SignalLPCPerceptWeightInit_F32
- mlib_SignalLPCPerceptWeightInit_S16
- mlib_SignalLPCPerceptWeight_S16
- mlib_SignalLPCPerceptWeight_S16_Adp
- mlib_SignalLPCPitchAnalyze_F32
- mlib_SignalLPCPitchAnalyze_S16
- mlib_SignalLSP2LPC_F32
- mlib_SignalLSP2LPC_S16
- mlib_SignalLSP2LPC_S16_Adp
- mlib_SignalMelCepstral_F32
- mlib_SignalMelCepstralFree_F32
- mlib_SignalMelCepstralFree_S16
- mlib_SignalMelCepstralInit_F32
- mlib_SignalMelCepstralInit_S16
- mlib_SignalMelCepstral_S16
- mlib_SignalMelCepstral_S16_Adp
- mlib_SignalMerge_F32S_F32
- mlib_SignalMerge_S16S_S16
- mlib_SignalMulBartlett_F32
- mlib_SignalMulBartlett_F32_F32
- mlib_SignalMulBartlett_F32S
- mlib_SignalMulBartlett_F32S_F32S
- mlib_SignalMulBartlett_S16
- mlib_SignalMulBartlett_S16S
- mlib_SignalMulBartlett_S16_S16
- mlib_SignalMulBartlett_S16_S16S
- mlib_SignalMulBlackman_F32
- mlib_SignalMulBlackman_F32_F32
- mlib_SignalMulBlackman_F32S
- mlib_SignalMulBlackman_F32S_F32S
- mlib_SignalMulBlackman_S16
- mlib_SignalMulBlackman_S16S
- mlib_SignalMulBlackman_S16_S16
- mlib_SignalMulBlackman_S16S_S16S
- mlib_SignalMul_F32
- mlib_SignalMul_F32_F32
- mlib_SignalMul_F32S
- mlib_SignalMul_F32S_F32S
- mlib_SignalMulHamming_F32
- mlib_SignalMulHamming_F32_F32
- mlib_SignalMulHamming_F32S
- mlib_SignalMulHamming_F32S_F32S
- mlib_SignalMulHamming_S16
- mlib_SignalMulHamming_S16S
- mlib_SignalMulHamming_S16_S16
- mlib_SignalMulHamming_S16S_S16S
- mlib_SignalMulHanning_F32
- mlib_SignalMulHanning_F32_F32
- mlib_SignalMulHanning_F32S
- mlib_SignalMulHanning_F32S_F32S
- mlib_SignalMulHanning_S16
- mlib_SignalMulHanning_S16S
- mlib_SignalMulHanning_S16_S16
- mlib_SignalMulHanning_S16S_S16S
- mlib_SignalMulKaiser_F32
- mlib_SignalMulKaiser_F32_F32
- mlib_SignalMulKaiser_F32S
- mlib_SignalMulKaiser_F32S_F32S
- mlib_SignalMulKaiser_S16
- mlib_SignalMulKaiser_S16S
- mlib_SignalMulKaiser_S16_S16
- mlib_SignalMulKaiser_S16S_S16S
- mlib_SignalMulRectangular_F32
- mlib_SignalMulRectangular_F32_F32
- mlib_SignalMulRectangular_F32S
- mlib_SignalMulRectangular_F32S_F32S
- mlib_SignalMulRectangular_S16
- mlib_SignalMulRectangular_S16S
- mlib_SignalMulRectangular_S16_S16
- mlib_SignalMulRectangular_S16S_S16S
mlib_SignalMulRectangular_S16_S16S
mlib_SignalMul_S16_S16_Sat
mlib_SignalMul_S16S_S16S_Sat
mlib_SignalMul_S16S_Sat
mlib_SignalMulSAdd_F32
mlib_SignalMulSAdd_F32_F32
mlib_SignalMulSAdd_F32S
mlib_SignalMulSAdd_F32S_F32S
mlib_SignalMulSAdd_S16_S16_Sat
mlib_SignalMulSAdd_S16_Sat
mlib_SignalMulSAdd_S16S_S16S_Sat
mlib_SignalMulSAdd_S16S_Sat
mlib_SignalMulS_F32
mlib_SignalMulS_F32_F32
mlib_SignalMulS_F32S
mlib_SignalMulS_F32S_F32S
mlib_SignalMulShift_S16_S16_Sat
mlib_SignalMulShift_S16_Sat
mlib_SignalMulShift_S16S_S16S_Sat
mlib_SignalMulShift_S16S_Sat
mlib_SignalMulSShiftAdd_S16_S16_Sat
mlib_SignalMulSShiftAdd_S16_Sat
mlib_SignalMulSShiftAdd_S16S_S16S_Sat
mlib_SignalMulSShiftAdd_S16S_Sat
mlib_SignalMulSShift_S16_S16_Sat
mlib_SignalMulSShift_S16_Sat
mlib_SignalMulSShift_S16S_S16S_Sat
mlib_SignalMulSShift_S16S_Sat
mlib_SignalMulWindow_F32
mlib_SignalMulWindow_F32_F32
mlib_SignalMulWindow_F32S
mlib_SignalMulWindow_F32S_F32S
mlib_SignalMulWindow_S16
mlib_SignalMulWindow_S16S
mlib_SignalMulWindow_S16_S16
mlib_SignalMulWindow_S16S_S16S
mlib_SignalNLMSFilter_F32_F32
mlib_SignalNLMSFilter_F32S_F32S
mlib_SignalNLMSFilterFree_F32_F32
- mlib_SignalNLMSFilterFree_F32S_F32S
- mlib_SignalNLMSFilterFree_S16_S16
- mlib_SignalNLMSFilterFree_S16S_S16S
- mlib_SignalNLMSFilterInit_F32_F32
- mlib_SignalNLMSFilterInit_F32S_F32S
- mlib_SignalNLMSFilterInit_S16_S16
- mlib_SignalNLMSFilterInit_S16S_S16S
- mlib_SignalNLMSFilterNonAdapt_F32_F32
- mlib_SignalNLMSFilterNonAdapt_F32S_F32S
- mlib_SignalNLMSFilterNonAdapt_S16_S16_Sat
- mlib_SignalNLMSFilterNonAdapt_S16S_S16S_Sat
- mlib_SignalNLMSFilter_S16_S16_Sat
- mlib_SignalNLMSFilter_S16S_S16S_Sat
- mlib_SignalQuant2_S16_F32
- mlib_SignalQuant2_S16S_F32S
- mlib_SignalQuant_S16_F32
- mlib_SignalQuant_S16S_F32S
- mlib_SignalQuant_U8_F32
- mlib_SignalQuant_U8_S16
- mlib_SignalQuant_U8S_F32S
- mlib_SignalQuant_U8S_S16S
- mlib_SignalReSampleFIR_F32_F32
- mlib_SignalReSampleFIR_F32S_F32S
- mlib_SignalReSampleFIRFree_F32_F32
- mlib_SignalReSampleFIRFree_F32S_F32S
- mlib_SignalReSampleFIRFree_S16_S16
- mlib_SignalReSampleFIRFree_S16S_S16S
- mlib_SignalReSampleFIRInit_F32_F32
- mlib_SignalReSampleFIRInit_F32S_F32S
- mlib_SignalReSampleFIRInit_S16_S16
- mlib_SignalReSampleFIRInit_S16S_S16S
- mlib_SignalReSampleFIR_S16_S16_Sat
- mlib_SignalReSampleFIR_S16S_S16S_Sat
- mlib_SignalSineWave_F32
- mlib_SignalSineWaveFree_F32
- mlib_SignalSineWaveFree_S16
- mlib_SignalSineWaveInit_F32
- mlib_SignalSineWaveInit_S16
- mlib_SignalSineWave_S16
- mlib_SignalSplit_F32_F32S
- mlib_SignalSplit_S16_S16S
- mlib_SignaluLaw2ALaw
- mlib_SignaluLaw2Linear
- mlib_SignalUpSample_F32_F32

Library Interfaces and Headers 271
Video Processing Functions

- mlib_SignalUpSample_F32S_F32S
- mlib_SignalUpSampleFIR_F32_F32
- mlib_SignalUpSampleFIR_F32S_F32S
- mlib_SignalUpSampleFIRFree_F32_F32
- mlib_SignalUpSampleFIRFree_F32S_F32S
- mlib_SignalUpSampleFIRInit_F32_F32
- mlib_SignalUpSampleFIRInit_F32S_F32S
- mlib_SignalUpSampleFIRInit_S16_S16
- mlib_SignalUpSampleFIRInit_S16S_S16S
- mlib_SignalUpSampleFIR_S16_S16_Sat
- mlib_SignalUpSampleFIR_S16S_S16S_Sat
- mlib_SignalUpSample_S16_S16
- mlib_SignalUpSample_S16S_S16S
- mlib_SignalWhiteNoise_F32
- mlib_SignalWhiteNoiseFree_F32
- mlib_SignalWhiteNoiseFree_S16
- mlib_SignalWhiteNoiseInit_F32
- mlib_SignalWhiteNoiseInit_S16
- mlib_SignalWhiteNoise_S16
- mlib_SignalWhiteNoise_S16_S16
- mlib_VideoAddBlock_U8_S16
- mlib_VideoColorABGR2JFIFYCC420
- mlib_VideoColorABGR2JFIFYCC422
- mlib_VideoColorABGR2JFIFYCC444
- mlib_VideoColorABGR2RGB
- mlib_VideoColorABGRint_to_ARGBint
- mlib_VideoColorARGB2JFIFYCC420
- mlib_VideoColorARGB2JFIFYCC422
- mlib_VideoColorARGB2JFIFYCC444
- mlib_VideoColorARGB2RGB
- mlib_VideoColorBG2JFIFYCC420
- mlib_VideoColorBG2JFIFYCC422
- mlib_VideoColorBG2JFIFYCC444
- mlib_VideoColorBG2JFIFYCC444_S16
- mlib_VideoColorGRaint_to_ABGRint
- mlib_VideoColorGRint_to_ABGRint
- mlib_VideoColorBlendABGR
- mlib_VideoColorBlendABGR_Inp
- mlib_VideoColorBlendABGR_ResetAlpha
- mlib_VideoColorBlendABGR_ResetAlpha_Inp
- mlib_VideoColorCMYK2JFIFYCC444
- mlib_VideoColorJFIFYCC2ABGR444
- mlib_VideoColorJFIFYCC2ARGB444
libmlib(3LIB)

- mlib_VideoColorJFIFYCC2RGB420
- mlib_VideoColorJFIFYCC2RGB420_Nearest
- mlib_VideoColorJFIFYCC2RGB422
- mlib_VideoColorJFIFYCC2RGB422_Nearest
- mlib_VideoColorJFIFYCC2RGB444
- mlib_VideoColorJFIFYCC2RGB444_S16
- mlib_VideoColorJFIFYCK2CMYK444
- mlib_VideoColorMerge2
- mlib_VideoColorMerge2_S16
- mlib_VideoColorMerge3
- mlib_VideoColorMerge3_S16
- mlib_VideoColorMerge4
- mlib_VideoColorMerge4_S16
- mlib_VideoColorResizeABGR
- mlib_VideoColorRGB2ABGR
- mlib_VideoColorRGB2ARGB
- mlib_VideoColorRGB2JFIFYCC420
- mlib_VideoColorRGB2JFIFYCC422
- mlib_VideoColorRGB2JFIFYCC444
- mlib_VideoColorRGB2JFIFYCC444_S16
- mlib_VideoColorRGBAint_to_ABGRint
- mlib_VideoColorRGBInt_to_ABGRInt
- mlib_VideoColorRGBInt_to_BGRAint
- mlib_VideoColorRGBseq_to_ABGRint
- mlib_VideoColorRGBXint_to_ABGRint
- mlib_VideoColorRGBXint_to_ARGBint
- mlib_VideoColorSplit2
- mlib_VideoColorSplit2_S16
- mlib_VideoColorSplit3
- mlib_VideoColorSplit3_S16
- mlib_VideoColorSplit4
- mlib_VideoColorSplit4_S16
- mlib_VideoColorUYV444int_to_ABGRint
- mlib_VideoColorUYV444int_to_AGBRint
- mlib_VideoColorUYV444int_to_UVY422int
- mlib_VideoColorUYV444int_to_UYV422int
- mlib_VideoColorUYVY422int_to_AGBRint
- mlib_VideoColorUYVY422int_to_ARGBInt
- mlib_VideoColorXRGBInt_to_AGBRInt
- mlib_VideoColorXRGBInt_to_ARGBInt
- mlib_VideoColorYUV2ABGR411
- mlib_VideoColorYUV2ABGR420
- mlib_VideoColorYUV2ABGR420_W
- mlib_VideoColorYUV2ABGR420_WX2
- mlib_VideoColorYUV2ABGR420_WX3
- mlib_VideoColorYUV2ABGR420_X2
- mlib_VideoColorYUV2ABGR420_X3
- mlib_VideoColorYUV2ABGR422
- mlib_VideoColorYUV2ABGR444
- mlib_VideoColorYUV2ARG411
- mlib_VideoColorYUV2ARG420
- mlib_VideoColorYUV2ARG422
- mlib_VideoColorYUV2ARG444
- mlib_VideoColorYUV2RGB411
- mlib_VideoColorYUV2RGB420
- mlib_VideoColorYUV2RGB422
- mlib_VideoColorYUV2RGB444
- mlib_VideoColorYUV411seq_to_ABGRint
- mlib_VideoColorYUV411seq_to_ARGBint
- mlib_VideoColorYUV411seq_to_UYVY422int
- mlib_VideoColorYUV411seq_to_YUYV422int
- mlib_VideoColorYUV420seq_to_ABGRint
- mlib_VideoColorYUV420seq_to_ARGBint
- mlib_VideoColorYUV420seq_to_UYVY422int
- mlib_VideoColorYUV420seq_to_YUYV422int
- mlib_VideoColorYUV422seq_to_ABGRint
- mlib_VideoColorYUV422seq_to_ARGBint
- mlib_VideoColorYUV422seq_to_UYVY422int
- mlib_VideoColorYUV422seq_to_YUYV422int
- mlib_VideoColorYUV444int_to_ABGRint
- mlib_VideoColorYUV444int_to_ARGBint
- mlib_VideoColorYUV444int_to_UYVY422int
- mlib_VideoColorYUV444int_to_YUYV422int
- mlib_VideoColorYUV444seq_to_ABGRint
- mlib_VideoColorYUV444seq_to_ARGBint
- mlib_VideoColorYUV444seq_to_UYVY422int
- mlib_VideoColorYUV444seq_to_YUYV422int
- mlib_VideoCopyRefAve_U8_U8
- mlib_VideoCopyRefAve_U8_U8_16x16
- mlib_VideoCopyRefAve_U8_U8_16x8
- mlib_VideoCopyRefAve_U8_U8_8x16
- mlib_VideoCopyRefAve_U8_U8_8x4
- mlib_VideoCopyRef_S16_U8
- mlib_VideoCopyRef_S16_U8_16x16
- mlib_VideoCopyRef_S16_U8_16x8
mlib_VideoCopyRef_S16_U8_8x16
mlib_VideoCopyRef_S16_U8_8x4
mlib_VideoCopyRef_S16_U8_8x8
mlib_VideoCopyRef_U8_U8
mlib_VideoCopyRef_U8_U8_16x16
mlib_VideoCopyRef_U8_U8_16x8
mlib_VideoCopyRef_U8_U8_8x16
mlib_VideoCopyRef_U8_U8_8x8
mlib_VideoCopyRef_U8_U8_8x4
mlib_VideoCopyRef_U8_U8_8x8
mlib_VideoDCT16x16_S16_S16
mlib_VideoDCT16x16_S16_S16_B10
mlib_VideoDCT2x2_S16_S16
mlib_VideoDCT4x4_S16_S16
mlib_VideoDCT8x8Quantize_S16_S16_B12
mlib_VideoDCT8x8Quantize_S16_S16_B12NA
mlib_VideoDCT8x8Quantize_S16_S16_U8
mlib_VideoDCT8x8Quantize_S16_S16_U8NA
mlib_VideoDCT8x8Quantize_S16_S16
mlib_VideoDCT8x8Quantize_S16_S16NA
mlib_VideoIDCT8x8_S16_S16
mlib_VideoIDCT8x8_S16_S16_B12
mlib_VideoIDCT8x8_S16_S16_B12NA
mlib_VideoIDCT8x8_S16_S16_DC
mlib_VideoIDCT8x8_S16_S16NA
mlib_VideoIDCT8x8_S16_S16Q1
mlib_VideoIDCT8x8_S16_S16Q1Mismatch
mlib_VideoIDCT8x8_U8_S16
mlib_VideoDownSample420
mlib_VideoDownSample420S16
mlib_VideoDownSample422
mlib_VideoDownSample422S16
mlib_VideoH263OverlappedMC_S16_U8
mlib_VideoH263OverlappedMC_U8_U8
mlib_VideoIDCT8x8_S16_S16
mlib_VideoIDCT8x8_S16_S16B12
mlib_VideoIDCT8x8_S16_S16B12NA
mlib_VideoIDCT8x8_S16_S16DC
mlib_VideoIDCT8x8_S16_S16NA
mlib_VideoIDCT8x8_S16_S16Q1
mlib_VideoIDCT8x8_S16_S16Q1Mismatch
mlib_VideoIDCT8x8_U8_S16
- mlib_VideoIDCT8x8_U8_S16_DC
- mlib_VideoIDCT8x8_U8_S16_NA
- mlib_VideoIDCT8x8_U8_S16_Q1
- mlib_VideoIDCT_IEEE_S16_S16
- mlib_VideoInterpAveX_U8_U8
- mlib_VideoInterpAveX_U8_U8_16x16
- mlib_VideoInterpAveX_U8_U8_16x8
- mlib_VideoInterpAveX_U8_U8_8x16
- mlib_VideoInterpAveX_U8_U8_8x4
- mlib_VideoInterpAveX_U8_U8_8x8
- mlib_VideoInterpAveXY_U8_U8
- mlib_VideoInterpAveXY_U8_U8_16x16
- mlib_VideoInterpAveXY_U8_U8_16x8
- mlib_VideoInterpAveXY_U8_U8_8x16
- mlib_VideoInterpAveXY_U8_U8_8x4
- mlib_VideoInterpAveXY_U8_U8_8x8
- mlib_VideoInterpX_S16_U8
- mlib_VideoInterpX_S16_U8_16x16
- mlib_VideoInterpX_S16_U8_16x8
- mlib_VideoInterpX_S16_U8_8x16
- mlib_VideoInterpX_S16_U8_8x4
- mlib_VideoInterpX_S16_U8_8x8
- mlib_VideoInterpX_U8_U8
- mlib_VideoInterpX_U8_U8_16x16
- mlib_VideoInterpX_U8_U8_16x8
- mlib_VideoInterpX_U8_U8_8x16
- mlib_VideoInterpX_U8_U8_8x4
- mlib_VideoInterpX_U8_U8_8x8
- mlib_VideoInterpXY_S16_U8
- mlib_VideoInterpXY_S16_U8_16x16
- mlib_VideoInterpXY_S16_U8_16x8
- mlib_VideoInterpXY_S16_U8_8x16
- mlib_VideoInterpXY_S16_U8_8x4
- mlib_VideoInterpXY_S16_U8_8x8
- mlib_VideoInterpXY_U8_U8_8x4
- mlib_VideoInterpXY_U8_U8_8x8
- mlib_VideoInterpXY_XY_U8_U8
- mlib_VideoInterpY_S16_U8
- mlib_VideoInterpY_S16_U8_16x16
- mlib_VideoInterpY_S16_U8_16x8
- mlib_VideoInterpY_S16_U8_8x16
- mlib_VideoInterpY_S16_U8_8x8
- mlib_VideoInterpY_U8_U8
- mlib_VideoInterpY_U8_U8_16x16
- mlib_VideoInterpY_U8_U8_16x8
- mlib_VideoInterpY_U8_U8_8x16
- mlib_VideoInterpY_U8_U8_8x8
- mlib_VideoInterpY_S16_U8_16x16
- mlib_VideoInterpY_S16_U8_16x8
- mlib_VideoInterpY_S16_U8_8x16
- mlib_VideoInterpY_S16_U8_8x8
- mlib_VideoInterpY_S16_U8_8x4
- mlib_VideoInterpY_S16_U8_8x8
- mlib_VideoInterpY_S16_U8_16x16
- mlib_VideoInterpY_S16_U8_16x8
- mlib_VideoInterpY_S16_U8_8x16
- mlib_VideoInterpY_S16_U8_8x8
- mlib_VideoP64Decimate_U8_U8
- mlib_VideoP64Loop_S16_U8
- mlib_VideoP64Loop_U8_U8
- mlib_VideoQuantizeInit_S16
- mlib_VideoQuantizeInit_S16
- mlib_VideoReversibleColorRGB2YUV_S16_S16
- mlib_VideoReversibleColorRGB2YUV_S16_U8
- mlib_VideoReversibleColorRGB2YUV_S32_S16
- mlib_VideoReversibleColorRGB2YUV_S16_U8
- mlib_VideoReversibleColorYUV2RGB_S16_S16
- mlib_VideoReversibleColorYUV2RGB_S16_S32
- mlib_VideoReversibleColorYUV2RGB_U8_S16
- mlib_VideoReversibleColorYUV2RGB_U8_U8
- mlib_VideoSignMagnitudeConvert_S16
- mlib_VideoSignMagnitudeConvert_S16_S16
- mlib_VideoSignMagnitudeConvert_S32
- mlib_VideoSignMagnitudeConvert_S32_S32
- mlib_VideoSumAbsDiff
- mlib_VideoUpSample420
- mlib_VideoUpSample420_Nearest
- mlib_VideoUpSample420_Nearest_S16
- mlib_VideoUpSample420_S16
- mlib_VideoUpSample422
- mlib_VideoUpSample422_Nearest
- mlib_VideoUpSample422_Nearest_S16
- mlib_VideoUpSample422_S16
- mlib_VideoWaveletForwardTwoTenTrans_S16_S16
- mlib_VideoWaveletForwardTwoTenTrans_S16_U8
- mlib_VideoWaveletForwardTwoTenTrans_S32_S16
libmlib(3LIB)

- mlib_VideoWaveletForwardTwoTenTrans_S32_S32
- mlib_VideoWaveletInverseTwoTenTrans_S16_S16
- mlib_VideoWaveletInverseTwoTenTrans_S16_S32
- mlib_VideoWaveletInverseTwoTenTrans_S32_S32
- mlib_VideoWaveletInverseTwoTenTrans_U8_S16
- mlib_VolumeFindMaxBMask_S16
- mlib_VolumeFindMaxBMask_U8
- mlib_VolumeFindMaxCMask_S16
- mlib_VolumeFindMaxCMask_U8
- mlib_VolumeFindMax_S16
- mlib_VolumeFindMax_U8
- mlib_VolumeRayCast_Blocked_Divergent_Nearest_S16_S16
- mlib_VolumeRayCast_Blocked_Divergent_Nearest_U8_U8
- mlib_VolumeRayCast_Blocked_Divergent_Trilinear_S16_S16
- mlib_VolumeRayCast_Blocked_Divergent_Trilinear_U8_U8
- mlib_VolumeRayCast_Blocked_Parallel_Nearest_S16_S16
- mlib_VolumeRayCast_Blocked_Parallel_Nearest_U8_U8
- mlib_VolumeRayCast_Blocked_Parallel_Trilinear_S16_S16
- mlib_VolumeRayCast_Blocked_Parallel_Trilinear_U8_U8
- mlib_VolumeRayCast_General_Divergent_Nearest_S16_S16
- mlib_VolumeRayCast_General_Divergent_Nearest_U8_Bit
- mlib_VolumeRayCast_General_Divergent_Nearest_U8_U8
- mlib_VolumeRayCast_General_Divergent_Trilinear_S16_S16
- mlib_VolumeRayCast_General_Divergent_Trilinear_U8_U8
- mlib_VolumeRayCast_General_Parallel_Nearest_S16_S16
- mlib_VolumeRayCast_General_Parallel_Nearest_U8_Bit
- mlib_VolumeRayCast_General_Parallel_Nearest_U8_U8
- mlib_VolumeRayCast_General_Parallel_Trilinear_S16_S16
- mlib_VolumeRayCast_General_Parallel_Trilinear_U8_U8
- mlib_VolumeWindowLevel

Files

- /usr/lib/libmlib.so.2 shared object
- /usr/lib64/libmlib.so.2 64-bit shared object

Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE NAME</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWmlib</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Committed</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>
See Also  Intro(3), attributes(5)
          mediaLib User's Manual
libmlib_mt(3LIB)

Name  libmlib_mt – multi-threaded mediaLib

Synopsis  cc [ flag... ] file... -lmlib_mt -lmlib [ library... ]
#include <mlib.h>

Description  Interfaces in this library provide functions for multimedia processing. Multi-threaded (MT) mediaLib is a software layer developed on top of mediaLib using OpenMP. When it is used with a large data set on a multi-processor system, MT mediaLib will partition data into subsets and process the subsets in parallel, thus greatly improving performance of applications that use mediaLib.

Interfaces  The shared object libmlib_mt.so.2 provides the same public interfaces as those defined in libmlib(3LIB). See Intro(3) for additional information on shared object interfaces.

Usage  There are two ways to use MT mediaLib.

1. Pre-load a multi-threaded mediaLib library during runtime by setting the LD_PRELOAD environment variable as follows before starting your application, in Bourne/Korn shell:

LD_PRELOAD=libmlib_mt.so
export LD_PRELOAD

or in C shell:

setenv LD_PRELOAD libmlib_mt.so

In this way, you can take advantage of MT mediaLib without rebuilding your application.

2. Link your application with a multi-threaded mediaLib library directly as shown under SYNOPSIS. In this way, an MT mediaLib library is always used whenever your application is started.

The parallelization of MT mediaLib is controlled, in part, by the PARALLEL environment variable. You can change its setting to adjust the degree of parallelization before starting your application, in Bourne/Korn shell:

PARALLEL=n
export PARALLEL

or in C shell:

setenv PARALLEL n

where n is a positive integer for number of threads. Note that other factors also affect the degree of parallelization in MT mediaLib.

Files  /usr/lib/libmlib_mt.so.2  shared object
/usr/lib/64/libmlib_mt.so.2  64-bit shared object
Attributes  See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNW/mlibt</td>
</tr>
<tr>
<td>MT-Level</td>
<td>MT-Safe</td>
</tr>
</tbody>
</table>

See Also  Intro(3), libmlib(3LIB), attributes(5)

mediaLib User's Manual
libmp – multiple precision library

Synopsis

cc [ flag... ] file... -lmp [ library... ]
#include <mp.h>

Description

Functions in this library provide various multiple precision routines.

Interfaces

The shared object libmp.so.2 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

```
mp_gcd   mp_itom
mp_madd  mp_mcmp
mp_mdiv  mp_mfree
mp_min   mp_mout
mp_sqrt  mp_msub
mp_mtox  mp_mult
mp_pow   mp_rpow
mp_sdiv  mp_xtom
```

Files

```
/lib/libmp.so.1     shared object for binary compatibility only
/lib/libmp.so.2     shared object
/lib/64/libmp.so.2  64-bit shared object
```

Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcs1 (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcs1x (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

See Also

pvs(1), Intro(3), exp(3M), mp(3MP), attributes(5)
Name  libMPAPI, libmpapi – Common Multipath Management library

Synopsis  
```
cc [ flag... ] file... -lMPAPI [ library... ]
#include <mpapi.h>
```

Description  The functions in this library allow a management application to administer the multipath devices and associated resources through standard interfaces, independent of a vendor-unique multipathing solution.

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

- MP_AssignLogicalUnitToTPG
- MP_CancelOverridePath
- MP_CompareOIDs
- MP_DeregisterForObjectPropertyChanges
- MP_DeregisterForObjectVisibilityChanges
- MP_DeregisterPlugin
- MP_DisableAutoFailback
- MP_DisableAutoProbing
- MP_DisablePath
- MP_EnableAutoFailback
- MP_EnableAutoProbing
- MP_EnablePath
- MP_FreeOidList
- MP_GetAssociatedPathOidList
- MP_GetAssociatedPluginOid
- MP_GetAssociatedTPGOidList
- MP_GetDeviceProductOidList
- MP_GetDeviceProductProperties
- MP_GetLibraryProperties
- MP_GetMPLuOidListFromTPG
- MP_GetMultipathLus
- MP_GetObjectType
- MP_GetPathLogicalUnitProperties
- MP_GetPluginOidList
- MP_GetPluginProperties
- MP_GetProprietaryLoadBalanceOidList
- MP_GetProprietaryLoadBalanceProperties
- MP_GetTargetPortGroupProperties
- MP_GetTargetPortOidList
- MP_GetTargetPortProperties
- MP_RegisterForObjectPropertyChanges
Client applications link with the Common Library (using `-lMPAPI`) to access the interfaces. The Common Library dynamically loads an individual vendor-provided plugin library that is available through `MP_RegisterPlugin` on the host system.

Using `libMPAPI` involves the following steps:

1. Optionally calling `MP_GetLibraryProperties()` to retrieve the properties of the Common Library.
2. Calling `MP_GetPluginOidList()` to retrieve the registered plugin libraries.
3. Optionally calling `MP_GetPluginProperties()` to retrieve the properties of the plugin library.
4. Retrieve discovery information and property information on multipath devices and associated resources by calling the following:
   - `MP_GetAssociatedPathOidList()`
   - `MP_GetAssociatedTPGOidList()`
   - `MP_GetDeviceProductOidList()`
   - `MP_GetDeviceProductProperties()`
   - `MP_GetInitiatorPortOidList()`
   - `MP_GetInitiatorPortProperties()`
   - `MP_GetMPLuOidListFromTPG()`
   - `MP_GetMPLogicalUnitProperties()`
   - `MP_GetMultipathLus()`
   - `MP_GetPathLogicalUnitProperties()`
   - `MP_GetProprietaryLoadBalanceOidList()`
   - `MP_GetProprietaryLoadBalanceProperties()`
   - `MP_GetTargetPortGroupProperties()`
   - `MP_GetTargetPortOidList()`
   - `MP_GetTargetPortProperties()`
5. Register and deregister for property and visibility changes on multipath devices and associated resources by calling:
   - `MP_RegisterForObjectPropertyChanges()`
   - `MP_RegisterForObjectVisibilityChanges()`
6. Perform administrative operations on multipath devices and associated resources by calling:

- MP_DeregisterForObjectPropertyChanges()
- MP_DeregisterForObjectVisibilityChanges()
- MP_AssignLogicalUnitToTPG()
- MP_CancelOverridePath()
- MP_DisableAutoFailback()
- MP_DisableAutoProbing()
- MP_DisablePath()
- MP_EnableAutoFailback()
- MP_EnableAutoProbing()
- MP_EnablePath()
- MP_SetLogicalUnitLoadBalanceType()
- MP_SetOverridePath()
- MP_SetPathWeight()
- MP_SetPluginLoadBalanceType()
- MP_SetFailbackPollingRate()
- MP_SetProbingPollingRate()
- MP_SetProprietaryProperties()
- MP_SetTPGAccess()

**Errors** Errors are generally returned from the underlying VSL and can include any of the following values:

- **MP_STATUS_SUCCESS** This status value is returned when the requested operation is successfully carried out.
- **MP_STATUS_INVALID_PARAMETER** This status value is returned when parameters passed to an API are detected to be invalid or inappropriate for a particular API parameter. If the parameter is an object ID, this status indicates that the object type subfield is defined in this specification, but is not appropriate for this API.
- **MP_STATUS_UNKNOWN_FN** This status value is returned when a client function passed into the API is not a previously registered or known function.
- **MP_STATUS_FAILED** This status value is returned when the requested operation could not be carried out.
- **MP_STATUS_INSUFFICIENT_MEMORY** This status value is returned when the API could [not] allocate the memory required to complete the requested operation.
MP_STATUS_INVALID_OBJECT_TYPE  This status value is returned when an object ID includes a type subfield that is not defined in this specification.

MP_STATUS_OBJECT_NOT_FOUND  This status value is returned when the object associated with the ID specified in the API could not be located, or has been deleted. Note that an invalid object type is covered by MP_STATUS_INVALID_OBJECT_TYPE so this status is limited to an invalid object owner identifier or sequence number.

MP_STATUS_UNSUPPORTED  This status value is returned when the implementation does not support the requested function.

MP_STATUS_FN_REPLACED  This status value is returned when a client function passed into the API replaces a previously registered function.

MP_STATUS_ACCESS_STATE_INVALID  This status value is returned when a device processing MP_SetTPGAccess returns a status indicating that the caller is attempting to establish an illegal combination of access states.

MP_STATUS_PATH_NONOPERATIONAL  This status is returned when communication cannot be established with the path selected by the caller.

MP_STATUS_TRY_AGAIN  This status is returned when the plugin or driver is unable to complete the request, but might be able to complete it later.

MP_STATUS_NOT_PERMITTED  The operation is not permitted in the current configuration, but might be permitted in other configurations.

Files  /usr/lib/libMPAPI.so  shared object
       /usr/lib/64/libMPAPI.so  64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWmpapir</td>
</tr>
<tr>
<td></td>
<td>SUNWmpapi (Header file)</td>
</tr>
</tbody>
</table>
### Interface Stability

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard: ANSI INCITS 412 Multipath Management API</td>
<td></td>
</tr>
</tbody>
</table>

| MT-Level            | Safe                                                  |

**See Also**  
[Intro(3), MP_RegisterPlugin(3MPAPI), attributes(5)]

*Multipath Management API Version 1.0*
**libmtmalloc** – multi-threaded memory allocator library

**Synopsis**

```bash
cc [ flag... ] file... -lmtmalloc [ library... ]
#include <mtmalloc.h>
```

**Description**

Functions in this library provide concurrent access to heap space.

**Interfaces**

The shared object libmtmalloc.so.1 provides the public interfaces defined below. See *Intro(3)* for additional information on shared object interfaces.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>free</td>
<td></td>
</tr>
<tr>
<td>malloc</td>
<td></td>
</tr>
<tr>
<td>mallocctl</td>
<td></td>
</tr>
<tr>
<td>memalign</td>
<td></td>
</tr>
<tr>
<td>realloc</td>
<td></td>
</tr>
<tr>
<td>valloc</td>
<td></td>
</tr>
</tbody>
</table>

**MTMALLOC_OPTIONS**

A comma separated list of options. The supported options are:

- **MTEXCLUSIVE=Y**
  
  By default, libmtmalloc allocates $2^{NCPUS}$ buckets from which allocations occur. Threads share buckets based on their thread ID. If MTEXCLUSIVE is invoked, then $4^{NCPUS}$ buckets are used. Threads with thread id less than $2^{NCPUS}$ receive an exclusive bucket and thus do not need to use locks. Allocation performance for these buckets may be dramatically increased. One enabled MTEXCLUSIVE can not be disabled. This feature can be enabled by setting the MTMALLOC_OPTION MTEXCLUSIVE to “Y” or “y” or anything beginning with “y”. Alternatively it can be enabled by a call to mallocctl(3MALLOC).

- **MTMAXCACHE=16, 17, 18, 19, 20, or 21**
  
  By default, allocations less than $2^{16}$ bytes are allocated from buckets indexed by thread id. Using this MTMALLOC_OPTION setting, variable size of the cached allocations can be increased to $2^{17}, 2^{18}, 2^{19}, 2^{20},$ or $2^{21}$ by setting MTMAXCACHE to 17, 18, 19, 20, or 21. If MTMAXCACHE is set to less than 16 it is reset to 16. If MTMAXCACHE is set to more than 21, then it is reset to 21. This all occurs silently.

- **MTCHUNKSIZE=xx**
  
  Allocation buckets are sized by the chunk size and the size of the allocation request. The default setting is 9 for 32-bit applications and 64 for 64 bit applications. For the cost of address space, performance can sometimes be enhanced by increasing this parameter. See mallocctl(3MALLOC).

- **MTREAlsFREE=xx**
  
  If $xx > 1$, set the threshold for calling madvise(3C) with MADV_FREE. Calling madvise() will result in the memory associated with the allocation being returned to the kernel. When freed, allocations greater than $xx * pageSize$ will have madvise() called. If $xx$ is less than 2, it will be set to 2.
MTDEBUGPATTERN=Y
Writes misaligned data into the buffer after free(). When the buffer is deallocated, the contents are verified to ensure that there was no access to the buffer after the free. If the buffer has been dirtied, a SIGABRT signal is delivered to the process. The default behavior is not to write misaligned data. The pattern used is 0xdeadbeef. Use of this option results in a performance penalty.

MTINITBUFFER=Y
Writes misaligned data into the newly allocated buffer. This option is useful for detecting some accesses before initialization. The default behavior is not to write misaligned data to the newly allocated buffer. The pattern used is 0xbaddcafe. Use of this option results in a performance penalty.

MTDOUBLEFREE=Y
Allows double free of a pointer. The default behavior of double free results in a core dump.

Files
/usr/lib/libmtmalloc.so.1 shared object
/usr/lib/64/libmtmalloc.so.1 64–bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>system/library (32–bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64–bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also
pvs(1), sbrk(2), Intro(3), malloc(3C), malloc(3MALLOC), mapmalloc(3MALLOC), mtmalloc(3MALLOC), attributes(5)
libmvec(3LIB)

Name  libmvec – vector math library

Synopsis  cc [ flag... ] file... -lmvec [ library... ]

Description  This library contains functions to evaluate common mathematical functions for several arguments at once. The argument values are specified by one or more vectors (arrays) of data, and the corresponding result values are stored in another vector.

Interfaces  The shared object \libmvec.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

vatan_       vatanf_
 vatan2_      vatan2f_
 vc_abs_      vc_exp_
 vc_log_      vc_pow_
 vcos_        vcosf_
 vcospi_      vcosfpi_
 vexp_        vexpf_
 vhypot_      vhypotf_
 vlog_        vlogf_
 vpow_        vpowf_
 vrhypot_     vrhypotf_
 vsqrt_       vrsqrtf_
 vsin_        vsinf_
 vsinpi_      vsinpif_
 vsincos_     vsincosf_
 vsincospi_   vsincosfpi_
 vsqrt_       vsqrtf_
 vz_abs_      vz_exp_
 vz_log_      vz_pow_

Files  /lib/libmvec.so.1  shared object
       /lib/64/libmvec.so.1  64-bit shared object
Attributes  See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWlibmsr</td>
</tr>
<tr>
<td>MT-Level</td>
<td>MT-Safe</td>
</tr>
</tbody>
</table>

See Also  Intro(3), complex.h(3HEAD), libm(3LIB), attributes(5)
The functions in this library interact with the network listener daemon, `listen(1M)`. The functions are provided for services invoked by the listener daemon and for clients that connect to the services using `listen`.

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

```
nlsgetcall nlsprovider
nlsrequest
```

**Files**

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

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsal (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcsalx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

**See Also**

`listen(1M), Intro(3), attributes(5)`
**Name**  
libnsl – network services library

**Synopsis**  
cc \{ flag \} \file \*\nsl \{ library \*\}

**Description**  
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.

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 _xti prefix. Those symbols are used in implementing the X/Open Transport Interface (XTI) interfaces documented in libxnet. See libxnet(3LIB).

**Interfaces**  
The shared object \libnsl.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

```
__rpc_createerr  __t_errno
__nderror       __null_auth
_xti_accept     __xti_ALLOC
_xti_bind       __xti_close
_xti_connect    __xti_error
_xti_free       __xti_getinfo
_xti_getprotaddr __xti_getstate
_xti_listen     __xti_look
_xti_open       __xti_optmgmt
_xti_rcv        __xti_rcvconnect
_xti_rcvdis     __xti_rcvrel
_xti_rcvreldata __xti_rcvdata
_xti_rcvudata   __xti_rcvudata
_xti_rcvudata   __xti_snd
_xti_snddis     __xti_sndrel
_xti_sndreldata __xti_sndudata
_xti_sndv       __xti_sndvdata
_xti_sterror    __xti_sync
_xti_sysconf    __xti_unbind
```
_xti_xns$_accept  _xti_xns$_snd
auth_destroy          authdes_create
authdes_getucred      authdes_lock
authdes_seccreate     authnone_create
authsys_create        authsys_create_default
callrpc               clnt_broadcast
clnt_call             clnt_control
clnt_create           clnt_create_timed
clnt_create_vers      clnt_create_vers_timed
clnt_destroy          clnt_dg_create
clnt_door_create      clnt_freeres
clnt_geterr           clnt_pcreateerror
clnt_perrno           clnt_perror
clnt_raw_create       clnt_spcreateerror
clnt_sperrno          clnt_sperror
clnt_tli_create       clnt_tp_create
clnt_tp_create_timed  clnt_vc_create
clnttcp_create
clntudp_create
clntudp_bufcreate
dbmcloese
\delete
\dial
\endhostent
\endnetconfig
\endpath
\fetch
\freehostent
\freenectconfigent
\get_myaddress
\gethostbyaddr
\gethostbyaddr\_r
\gethostbyname
\gethostbyname\_r
gethostent_r  getipnodebyaddr
getipnodebyname  getipsecalgbyname
getipsecalgbyname  getipsecprotobynum
getipsecprotobynum  getnetconfig
getnetconfig  getnetname
getnetpath  getpublickey
getrpcbyname  getrpcbyname_r
getrpcbyname_r  getrpcbynumber
getrpcbynumber  getrpcbynumber_r
getrpcbynumber_r  getrpcbyname
getrpcent  getrpcent_r
getrpcent_r  getsecretkey
h_errno  host2netname
inet_addr  inet_netof
inet_netof  inet_ntoa_r
inet_ntoa_r  inet_ntop
inet_ntop  key_decryptsession
key_decryptsession  key_encryptsession
key_encryptsession  key_gendes
key_gendes  key_secretkey_is_set
key_secretkey_is_set  maxbno
maxbno  nc_errno
nc_errno  nc_sperror
nc_sperror  netdir_free
netdir_free  netdir_getbyaddr
netdir_getbyaddr  netdir_options
netdir_options  netdir_serror
netdir_serror  netname2host
netname2host  netname2user
netname2user  nis_add
nis_add  nis_add_entry
nis_add_entry  nis_addmember
nis_addmember  nis_checkpoint
nis_checkpoint  nis_clone_object
nis_clone_object  nis_creategroup
nis_creategroup  nis_data
nis_data  nis_destroy_object
nis_destroy_object  nis_destroygroup
nis_destroygroup  nis_dir_cmp
nis_dir_cmp  nis_domain_of
nis_domain_of  nis_dump
nis_dump
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nis_find_item</td>
<td>nis_finddirectory</td>
</tr>
<tr>
<td>nis_first_entry</td>
<td>nis_free_request</td>
</tr>
<tr>
<td>nis_freenames</td>
<td>nis_freeresult</td>
</tr>
<tr>
<td>nis_freeservlist</td>
<td>nis_freetags</td>
</tr>
<tr>
<td>nis_get_request</td>
<td>nis_get_static_storage</td>
</tr>
<tr>
<td>nis_getnames</td>
<td>nis_getservlist</td>
</tr>
<tr>
<td>nis_in_table</td>
<td>nis_insert_item</td>
</tr>
<tr>
<td>nis_insert_name</td>
<td>nis_ismember</td>
</tr>
<tr>
<td>nis_leaf_of</td>
<td>nis_leaf_of_r</td>
</tr>
<tr>
<td>nis_lerror</td>
<td>nis_list</td>
</tr>
<tr>
<td>nis_local_directory</td>
<td>nis_local_group</td>
</tr>
<tr>
<td>nis_local_host</td>
<td>nis_local_principal</td>
</tr>
<tr>
<td>nis_lookup</td>
<td>nis_make_error</td>
</tr>
<tr>
<td>nis_make_rpchandle</td>
<td>nis_mkdir</td>
</tr>
<tr>
<td>nis_modify</td>
<td>nis_modify_entry</td>
</tr>
<tr>
<td>nis_name_of</td>
<td>nis_next_entry</td>
</tr>
<tr>
<td>nis_perror</td>
<td>nis_ping</td>
</tr>
<tr>
<td>nis_print_directory</td>
<td>nis_print_entry</td>
</tr>
<tr>
<td>nis_print_group</td>
<td>nis_print_group_entry</td>
</tr>
<tr>
<td>nis_print_link</td>
<td>nis_print_object</td>
</tr>
<tr>
<td>nis_print_rights</td>
<td>nis_print_table</td>
</tr>
<tr>
<td>nis_read_obj</td>
<td>nis_remove</td>
</tr>
<tr>
<td>nis_remove_entry</td>
<td>nis_remove_item</td>
</tr>
<tr>
<td>nis_remove_name</td>
<td>nis_removemember</td>
</tr>
<tr>
<td>nis_rmdir</td>
<td>nis_servstate</td>
</tr>
<tr>
<td>nis_sperrno</td>
<td>nis_sperror</td>
</tr>
<tr>
<td>nis_sperror_r</td>
<td>nis_stats</td>
</tr>
<tr>
<td>nis_verifygroup</td>
<td>nis_write_obj</td>
</tr>
<tr>
<td>pmap_getmaps</td>
<td>pmap_getport</td>
</tr>
</tbody>
</table>
pmap_rmtcall
pmap_unset
rpc_broadcast
rpc_call
rpc_createerr
rpc_gss_get_mech_info
rpc_gss_get_principal_name
rpc_gss_getcred
rpc_gss_max_data_length
rpc_gss_qop_to_num
rpc_gss_set_callback
rpc_gss_svc_max_data_length
rpc_reg
rpcb_getaddr
rpcb_getmaps
rpcb_rmtcall
rpcb_unset
setnetconfig
setrpccent
svc_auth_reg
svc_create
svc_dg_create
svc_done
svc_exit
svc_fdset
svc_get_local_cred
svc_getreq
svc_getreq_poll
svc_getrpccaller
svc_pollfd

pmap_set
registerrpc
rpc_broadcast_exp
rpc_control
rpc_gss_get_error
rpc_gss_get_mechanisms
rpc_gss_get_versions
rpc_gss_is_installed
rpc_gss_mech_to_oid
rpc_gss_seccreate
rpc_gss_set_defaults
rpc_gss_svc_max_data_length
rpcb_gettime
rpcb_set
sethostent
setnetpath
store
svc_control
svc_destroy
svc_dg_enablecache
svc_door_create
svc_fd_create
svc_freeargs
svc_getargs
svc_getreq_common
svc_getreqset
svc_max_pollfd
svc_raw_create
svc_reg  svc_register
svc_run  svc_sendreply
svc_tli_create  svc_tp_create
svc_unreg  svc_unregister
svc_vc_create  svcerr_auth
svcerr_decode  svcerr_noproc
svcerr_noprograms  svcerr_progvers
svcerr_systemerr  svcerr_weakauth
svcfd_create  svcraw_create
svctcp_create  svcudp_bufcreate
svctcp_create  t_accept
v_alloc  t_bind
v_close  t_connect
t_errno  t_error
t_free  t_getinfo
t_getname  t_getstate
t_listen  t_look
v_nerr  t_open
v_optmgmt  t_rcv
t_rcvconnect  t_rcvdis
t_rcvrel  t_rcvdata
t_rcvuderr  t_snd
t_snddis  t_sndrel
t_sndudata  t_strerror
t_sync  t_unbind
taddr2uaddr  unaddr2taddr
undial  user2netname
xdr_accepted_reply  xdr_array
xdr_authsys_parms  xdr_bool
<table>
<thead>
<tr>
<th>Function</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>xdr_bytes</td>
<td>xdr_callhdr</td>
</tr>
<tr>
<td>xdr_callmsg</td>
<td>xdr_char</td>
</tr>
<tr>
<td>xdr_destroy</td>
<td>xdr_double</td>
</tr>
<tr>
<td>xdr_enum</td>
<td>xdr_float</td>
</tr>
<tr>
<td>xdr_free</td>
<td>xdr_getpos</td>
</tr>
<tr>
<td>xdr_hyper</td>
<td>xdr_inline</td>
</tr>
<tr>
<td>xdr_int</td>
<td>xdr_int16_t</td>
</tr>
<tr>
<td>xdr_int32_t</td>
<td>xdr_int64_t</td>
</tr>
<tr>
<td>xdr_int8_t</td>
<td>xdr_long</td>
</tr>
<tr>
<td>xdr_longlong_t</td>
<td>xdr_opaque</td>
</tr>
<tr>
<td>xdr_opaque_auth</td>
<td>xdr_pointer</td>
</tr>
<tr>
<td>xdr_quadruple</td>
<td>xdr_reference</td>
</tr>
<tr>
<td>xdr_rejected_reply</td>
<td>xdr_replymsg</td>
</tr>
<tr>
<td>xdr_setpos</td>
<td>xdr_short</td>
</tr>
<tr>
<td>xdr_sizeof</td>
<td>xdr_string</td>
</tr>
<tr>
<td>xdr_u_char</td>
<td>xdr_u_hyper</td>
</tr>
<tr>
<td>xdr_u_int</td>
<td>xdr_u_long</td>
</tr>
<tr>
<td>xdr_u_longlong_t</td>
<td>xdr_u_short</td>
</tr>
<tr>
<td>xdr_uint16_t</td>
<td>xdr_uint32_t</td>
</tr>
<tr>
<td>xdr_uint64_t</td>
<td>xdr_uint8_t</td>
</tr>
<tr>
<td>xdr_union</td>
<td>xdr_vector</td>
</tr>
<tr>
<td>xdr_void</td>
<td>xdr_wrapstring</td>
</tr>
<tr>
<td>xdrmem_create</td>
<td>xdrrec_create</td>
</tr>
<tr>
<td>xdrrec_endofrecord</td>
<td>xdrrec_eof</td>
</tr>
<tr>
<td>xdrrec_readbytes</td>
<td>xdrrec_skiprecord</td>
</tr>
<tr>
<td>xdrstdio_create</td>
<td>xprt_register</td>
</tr>
<tr>
<td>xprt_unregister</td>
<td>yp_all</td>
</tr>
<tr>
<td>yp_bind</td>
<td>yp_first</td>
</tr>
<tr>
<td>yp_get_default_domain</td>
<td>yp_master</td>
</tr>
</tbody>
</table>

*Library Interfaces and Headers 299*
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:

_new_svc_fdset

Files /lib/libnsl.so.1  shared object
       /lib/64/libnsl.so.1  64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe with exceptions</td>
</tr>
</tbody>
</table>

See Also pvs(1), Intro(2), Intro(3), libxnet(3LIB), attributes(5)
The `libnvpair` library exports a set of functions for managing name-value pairs.

The library defines four opaque handles:

- `nvpair_t` handle to a name-value pair
- `nvlist_t` handle to a list of name-value pairs
- `nv_alloc_t` handle to a pluggable allocator
- `nv_alloc_ops_t` handle to pluggable allocator operations

The library supports the following operations:

- Allocate and free an `nvlist_t`.
- Specify the allocator to be used when manipulating an `nvlist_t`.
- Add and remove an `nvpair_t` from a list.
- Search `nvlist_t` for a specified name pair.
- Pack an `nvlist_t` into a contiguous buffer.
- Expand a packed `nvlist` into a searchable `nvlist_t`.

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

```c
nvlist_add_boolean
nvlist_add_boolean_value
nvlist_add_boolean_array
nvlist_add_byte
nvlist_add_byte_array
nvlist_add_int8
nvlist_add_int8_array
nvlist_add_int16
nvlist_add_int16_array
nvlist_add_int32
nvlist_add_int32_array
nvlist_add_int64
nvlist_add_int64_array
nvlist_add_nvlist
nvlist_add_nvlist_array
nvlist_add_nvpair
nvlist_add_string
nvlist_add_string_array
nvlist_add_uint8
nvlist_add_uint8_array
nvlist_add_uint16
nvlist_add_uint16_array
nvlist_add_uint32
nvlist_add_uint32_array
```
<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>nvlist_add_uint64</td>
</tr>
<tr>
<td>nvlist_alloc</td>
</tr>
<tr>
<td>nvlist_free</td>
</tr>
<tr>
<td>nvlist_lookup_boolean_value</td>
</tr>
<tr>
<td>nvlist_lookup_byte</td>
</tr>
<tr>
<td>nvlist_lookup_int8</td>
</tr>
<tr>
<td>nvlist_lookup_int16</td>
</tr>
<tr>
<td>nvlist_lookup_int32</td>
</tr>
<tr>
<td>nvlist_lookup_int64</td>
</tr>
<tr>
<td>nvlist_lookup_nvlist</td>
</tr>
<tr>
<td>nvlist_lookup_nv_alloc</td>
</tr>
<tr>
<td>nvlist_lookup_string</td>
</tr>
<tr>
<td>nvlist_lookup_uint8</td>
</tr>
<tr>
<td>nvlist_lookup_uint16</td>
</tr>
<tr>
<td>nvlist_lookup_uint32</td>
</tr>
<tr>
<td>nvlist_lookup_uint64</td>
</tr>
<tr>
<td>nvlist_merge</td>
</tr>
<tr>
<td>nvlist_pack</td>
</tr>
<tr>
<td>nvlist_remove_all</td>
</tr>
<tr>
<td>nvlist_unpack</td>
</tr>
<tr>
<td>nvlist_xdup</td>
</tr>
<tr>
<td>nvlist_xunpack</td>
</tr>
<tr>
<td>nvpair_type</td>
</tr>
<tr>
<td>nvpair_value_boolean_value</td>
</tr>
<tr>
<td>nvpair_value_byte_array</td>
</tr>
<tr>
<td>nvpair_value_int8_array</td>
</tr>
<tr>
<td>nvpair_value_int16_array</td>
</tr>
<tr>
<td>nvpair_value_int32_array</td>
</tr>
<tr>
<td>nvpair_value_int64_array</td>
</tr>
<tr>
<td>nvpair_value_nvlist</td>
</tr>
</tbody>
</table>
libnvpair(3LIB)

nvpair_value_nvlist_array nvpair_value_string
nvpair_value_string_array nvpair_value_uint8
nvpair_value_uint8_array nvpair_value_uint16
nvpair_value_uint16_array nvpair_value_uint32
nvpair_value_uint32_array nvpair_value_uint64
nvpair_value_uint64_array nv_alloc_init
nv_alloc_fini nv_alloc_reset

Files
/lib/libnvpair.so.1 shared object
/lib/64/libnvpair.so.1 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>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>

See Also Intro(3), 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.

- `pam_acct_mgmt`
- `pam_authenticate`
- `pam_chauthtok`
- `pam_close_session`
- `pam_end`
- `pam_get_data`
- `pam_get_item`
- `pam_get_user`
- `pam_getenv`
- `pam_getenvlist`
- `pam_open_session`
- `pam_putenv`
- `pam_set_data`
- `pam_set_item`
- `pam_setcred`
- `pam_start`
- `pam_strerror`

**Files**  
/lib/libpam.so.1  
/etc/pam.conf  
/usr/lib/security/pam_dial_auth.so.1  
/usr/lib/security/pam_rhosts_auth.so.1  
/usr/lib/security/pam_sample.so.1

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl</td>
</tr>
<tr>
<td>MT Level</td>
<td>MT-Safe with exceptions</td>
</tr>
</tbody>
</table>

**See Also**  
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 is no longer supported. 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).`
libpanel(3LIB)

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.

bottom_panel del_panel
hide_panel move_panel
new_panel panel_above
panel_below panel_hidden
panel_userptr panel_window
replace_panel set_panel_userptr
show_panel top_panel
update_panels

Files /usr/lib/libpanel.so.1 shared object
/usr/lib/64/libpanel.so.1 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcsdx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

See Also Intro(3), libcurses(3LIB), attributes(5)
### Name
libpapi – Free Standards Group Open Printing API (PAPI) library functions

### Synopsis
```c
cc [ flag... ] file... -lpapi [ library... ]
#include <papi.h>
```

### Description
Functions in this library provide an interface for interaction with print services as described in v1.0 of the Free Standards Group (FSG) Open Printing API (PAPI).

This particular implementation of the PAPI includes naming support as described in the `printers.conf(4)` and `printers(4)` manual pages. It also supplies support for interaction with local LP services, remote LPD services, and remote IPP services through the use of loadable modules that export the same interface. These modules should not be linked with directly, but can be used directly at runtime through the use of `LD_PRELOAD` for debugging purposes.

### Interfaces
The shared object `libpapi.so.0` provides the public interfaces defined below. See `Intro(3)` for additional information on shared object interfaces.

#### Attribute
- `papiAttributeListAddBoolean`
- `papiAttributeListAddCollection`
- `papiAttributeListAddDatetime`
- `papiAttributeListAddInteger`
- `papiAttributeListAddMetadata`
- `papiAttributeListAddRange`
- `papiAttributeListAddResolution`
- `papiAttributeListAddString`
- `papiAttributeListAddValue`
- `papiAttributeListDelete`
- `papiAttributeListFind`
- `papiAttributeListFromString`
- `papiAttributeListGetBoolean`
- `papiAttributeListGetCollection`
- `papiAttributeListGetDatetime`
- `papiAttributeListGetInteger`
- `papiAttributeListGetMetadata`
- `papiAttributeListGetNext`
- `papiAttributeListGetRange`
- `papiAttributeListGetResolution`
- `papiAttributeListGetString`
- `papiAttributeListGetValue`
- `papiAttributeListToString`

#### Service
- `papiServiceCreate`
- `papiServiceDestroy`
- `papiServiceGetAppData`
- `papiServiceGetAttributeList`
- `papiServiceGetEncryption`
- `papiServiceGetPassword`
- `papiServiceGetServiceName`
- `papiServiceGetStatusMessage`
- `papiServiceGetUserName`
- `papiServiceSetAppData`
### libpapi(3LIB)

<table>
<thead>
<tr>
<th>Printer</th>
<th>Job</th>
<th>Miscellaneous</th>
<th>Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>papiPrinterAdd</td>
<td>papiJobCancel</td>
<td>papiLibrarySupportedCall</td>
<td>/usr/lib/libpapi.so.0</td>
</tr>
<tr>
<td>papiPrinterEnable</td>
<td>papiJobGetAttributeList</td>
<td>papiLibrarySupportedCalls</td>
<td>/usr/lib/libpapi-common.so.0</td>
</tr>
<tr>
<td>papiPrinterGetAttributeList</td>
<td>papiJobGetId</td>
<td></td>
<td>/usr/lib/print/psm-lpd.so</td>
</tr>
<tr>
<td>papiPrinterListJobs</td>
<td>papiJobGetPrinterName</td>
<td></td>
<td>/usr/lib/print/psm-lpshced.so</td>
</tr>
<tr>
<td>papiPrinterPause</td>
<td>papiJobHold</td>
<td></td>
<td>/usr/lib/print/psm-ipp.so</td>
</tr>
<tr>
<td>papiPrinterQuery</td>
<td>papiJobModify</td>
<td></td>
<td>/usr/lib/libipp-core.so</td>
</tr>
<tr>
<td>papiPrinterRemove</td>
<td>papiJobPromote</td>
<td></td>
<td></td>
</tr>
<tr>
<td>papiPrinterResume</td>
<td>papiJobRelease</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>papiJobStreamClose</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>papiJobStreamClose</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>papiJobSubmit</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>papiJobSubmitByReference</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>papiJobSubmitByReference</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- papiPrinterAdd
- papiPrinterEnable
- papiPrinterGetAttributeList
- papiPrinterListJobs
- papiPrinterPause
- papiPrinterQuery
- papiPrinterRemove
- papiPrinterResume
- papiPrinterAdd
- papiJobCancel
- papiJobGetAttributeList
- papiJobGetJobTicket
- papiJobHold
- papiJobModify
- papiJobPromote
- papiJobRelease
- papiJobStreamClose
- papiJobStreamWrite
- papiJobSubmitByReference
- papiJobSubmit
- papiJobSubmitByReference
- papiLibrarySupportedCall
- papiLibrarySupportedCalls
- papiStatusString

**Files**
- /usr/lib/libpapi.so.0  
  - shared object
- /usr/lib/libpapi-common.so.0  
  - private shared code
- /usr/lib/print/psm-lpd.so  
  - private rfc1179 support
- /usr/lib/print/psm-lpsched.so  
  - private LP support
- /usr/lib/print/psm-ipp.so  
  - private IPP support
- /usr/lib/libipp-core.so  
  - private IPP marshalling support
Attributes

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWpapi</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Volatile</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also: Intro(3), printers(4), printers.conf(4), attributes(5)
Name  libpctx – process context library

Synopsis  cc [ flag... ] file... -lpctx [ library... ]

Description  Functions in this library provide a simple means to access the underlying facilities of proc(4) to allow a controlling process to manipulate the state of a controlled process.

This library is primarily for use in conjunction with the libcpc(3LIB) library. Used together, these libraries allow developers to construct tools that can manipulate CPU performance counters in other processes. The cputrack(1) utility is an example of such a tool.

Interfaces  The shared object libpctx.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

pctx_capture  pctx_create
pctx_release  pctx_run
pctx_set_events

Files  /usr/lib/libpctx.so.1  shared object
/usr/lib/64/libpctx.so.1  64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcpcu (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcpcux (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also  cputrack(1), Intro(3), cpc(3CPC), libcpc(3LIB), proc(4), attributes(5)
**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.

- `picl_find_node`  
- `picl_get_first_prop`  
- `picl_get_frootree_parent`  
- `picl_get_next_by_col`  
- `picl_get_next_by_row`  
- `picl_get_next_prop`  
- `picl_get_node_by_path`  
- `picl_get_propinfo`  
- `picl_get_propinfo_by_name`  
- `picl_get_propval`  
- `picl_get_propval_by_name`  
- `picl_get_root`  
- `picl_set_propval`  
- `picl_set_propval_by_name`  
- `picl_shutdown`  
- `picl_strerror`  
- `picl_walk_tree_by_class`  

**Files**  
/usr/lib/libpicl.so.1  
/usr/lib/64/libpicl.so.1

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWpiclu (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWpiclx (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>

**See Also**  
pvs(1), Intro(3), libpicl(3PICL), attributes(5)
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.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>picld_plugin_register</td>
<td>ptree_add_node</td>
</tr>
<tr>
<td>ptree_add_prop</td>
<td>ptree_add_row_to_table</td>
</tr>
<tr>
<td>ptree_create_and_add_node</td>
<td>ptree_create_and_add_prop</td>
</tr>
<tr>
<td>ptree_create_node</td>
<td>ptree_create_prop</td>
</tr>
<tr>
<td>ptree_create_table</td>
<td>ptree_delete_node</td>
</tr>
<tr>
<td>ptree_delete_prop</td>
<td>ptree_destroy_node</td>
</tr>
<tr>
<td>ptree_destroy_prop</td>
<td>ptree_find_node</td>
</tr>
<tr>
<td>ptree_get_first_prop</td>
<td>ptree_get_frtree_parent</td>
</tr>
<tr>
<td>ptree_get_next_by_col</td>
<td>ptree_get_next_by_row</td>
</tr>
<tr>
<td>ptree_get_next_prop</td>
<td>ptree_get_node_by_path</td>
</tr>
<tr>
<td>ptree_get_prop_by_name</td>
<td>ptree_get_propinfo</td>
</tr>
<tr>
<td>ptree_get_propval</td>
<td>ptree_get_propval_by_name</td>
</tr>
<tr>
<td>ptree_get_root</td>
<td>ptree_init_propinfo</td>
</tr>
<tr>
<td>ptree_post_event</td>
<td>ptree_register_handler</td>
</tr>
<tr>
<td>ptree_unregister_handler</td>
<td>ptree_update_propval</td>
</tr>
<tr>
<td>ptree_update_propval_by_name</td>
<td>ptree_walk_tree_by_class</td>
</tr>
</tbody>
</table>

Files: /usr/lib/libpicltree.so.1 shared object

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

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWpiclu</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Evolving</td>
</tr>
<tr>
<td>ATTRIBUTE TYPE</td>
<td>ATTRIBUTE VALUE</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>MT-Level</td>
<td>MT-Safe</td>
</tr>
</tbody>
</table>

See Also  `pvs(1), Intro(3), libpicltree(3PICLTREE), attributes(5)`
**Name**
libpkcs11 – PKCS#11 Cryptographic Framework library

**Synopsis**
```
cc [ flag... ] file... -lpkcs11 [ library... ]
#include <security/cryptoki.h>
#include <security/pkcs11.h>
```

**Description**
The libpkcs11 library implements the RSA Security Inc. PKCS#11 Cryptographic Token Interface (Cryptoki), v2.20 specification by using plug-ins to provide the slots.

Each plug-in, which also implements RSA PKCS#11 v2.20, represents one or more slots.

The libpkcs11 library provides a special slot called the meta slot. The meta slot provides a virtual union of capabilities of all other slots. When available, the meta slot is always the first slot provided by libpkcs11.

The meta slot feature can be configured either system-wide or by individual users. System-wide configuration for meta slot features is done with the cryptoadm(1M) utility. User configuration for meta slot features is performed with environment variables.

By default, the following is the system-wide configuration for meta slot. Meta slot is enabled. Meta slot provides token-based object support with the Software RSA PKCS#11 softtoken (pkcs11 SoftToken(5)). Meta slot is allowed to move sensitive token objects to other slots if that is necessary to perform an operation.

Users can overwrite one or more system-wide configuration options for meta slot using these environment variables.

The `$METASLOT_OBJECTSTORE_SLOT` and `$METASLOT_OBJECTSTORE_TOKEN` environment variables are used to specify an alternate token object store. A user can specify either slot-description in `$METASLOT_OBJECTSTORE_SLOT` or token-label in `$METASLOT_OBJECTSTORE_TOKEN`, or both. Valid values for slot-description and token-label are available from output of the command:
```
cryptoadm list -v
```

The `$METASLOT_ENABLED` environment variable is used to specify whether the user wants to turn the metaslot feature on or off. Only two values are recognized. The value “true” means meta slot will be on. The value “false” means meta slot will be off.

The `$METASLOT_AUTO_KEY_MIGRATE` environment variable is used to specify whether the user wants sensitive token objects to move to other slots for cryptographic operations. Only two values are recognized. The value “true” means meta slot will migrate sensitive token objects to other slots if necessary. The value “false” means meta slot will not migrate sensitive token objects to other slots even if it is necessary.

When the meta slot feature is enabled, the slot that provides token-based object support is not shown as one of the available slots. All of its functionality can be used with the meta slot.

This library filters the list of mechanisms available from plug-ins based on the policy set by cryptoadm(1M).
This library provides entry points for all PKCS#11 v2.20 functions. See the RSA PKCS#11 v2.20 specification at http://www.rsasecurity.com.

Plug-ins are added to libpkcs11 by the pkcs11conf class action script during execution of pkgadd(1M). The available mechanisms are administered by the cryptoadm(1M) utility.

Plug-ins must have all of their library dependancies specified, including libc(3LIB). Libraries that have unresolved symbols, including those from libc, will be rejected and a message will be sent to syslog(3C) for such plug-ins.

Due to U.S. Export regulations, all plug-ins are required to be cryptographically signed using the elfsign utility.

Any plug-in that is not signed or is not a compatible version of PKCS#11 will be dropped by libpkcs11. When a plug-in is dropped, the administrator is alerted by the syslog(3C) utility.

The <security/pkcs11f.h> header contains function definitions. The <security/pkcs11t.h> header contains type definitions. Applications can include either of these headers in place of <security/pkcs11.h>, which contains both function and type definitions.

The shared object libpkcs11.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

<table>
<thead>
<tr>
<th>PKCS#11 Standard</th>
<th>Interfaces</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_CloseAllSessions</td>
<td>C_CloseSession</td>
</tr>
<tr>
<td>C_CopyObject</td>
<td>C_CreateObject</td>
</tr>
<tr>
<td>C_Decrypt</td>
<td>C_DecryptDigestUpdate</td>
</tr>
<tr>
<td>C_DecryptFinal</td>
<td>C_DecryptInit</td>
</tr>
<tr>
<td>C_DecryptUpdate</td>
<td>C_DecryptVerifyUpdate</td>
</tr>
<tr>
<td>C_DeriveKey</td>
<td>C_DestroyObject</td>
</tr>
<tr>
<td>C_DIGEST</td>
<td>C_DigestEncryptUpdate</td>
</tr>
<tr>
<td>C_DigestFinal</td>
<td>C_DigestInit</td>
</tr>
<tr>
<td>C_DigestKey</td>
<td>C_DigestUpdate</td>
</tr>
<tr>
<td>C_Encrypt</td>
<td>C_EncryptFinal</td>
</tr>
<tr>
<td>C_EncryptInit</td>
<td>C_EncryptUpdate</td>
</tr>
<tr>
<td>C_Finalize</td>
<td>C_FindObjects</td>
</tr>
<tr>
<td>C_FindObjectsFinal</td>
<td>C_FindObjectsInit</td>
</tr>
<tr>
<td>C_generateKey</td>
<td>C_generateKeyPair</td>
</tr>
</tbody>
</table>
libpckcs11(3LIB)

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>C_GenerateRandom</td>
</tr>
<tr>
<td>C_GetAttributeValue</td>
</tr>
<tr>
<td>C_GetFunctionList</td>
</tr>
<tr>
<td>C_GetInfo</td>
</tr>
<tr>
<td>C_GetMechanismInfo</td>
</tr>
<tr>
<td>C_GetMechanismList</td>
</tr>
<tr>
<td>CGetObjectSize</td>
</tr>
<tr>
<td>C_GetOperationState</td>
</tr>
<tr>
<td>C_GetSessionInfo</td>
</tr>
<tr>
<td>C_GetSlotInfo</td>
</tr>
<tr>
<td>C_GetSlotList</td>
</tr>
<tr>
<td>C_GetTokenInfo</td>
</tr>
<tr>
<td>C_InitPIN</td>
</tr>
<tr>
<td>C_InitToken</td>
</tr>
<tr>
<td>C_Initialize</td>
</tr>
<tr>
<td>C_Login</td>
</tr>
<tr>
<td>C_Logout</td>
</tr>
<tr>
<td>C_OpenSession</td>
</tr>
<tr>
<td>C_SeedRandom</td>
</tr>
<tr>
<td>C_SetAttributeValue</td>
</tr>
<tr>
<td>C_SetOperationState</td>
</tr>
<tr>
<td>C_SetPIN</td>
</tr>
<tr>
<td>C_Sign</td>
</tr>
<tr>
<td>C_SignFinal</td>
</tr>
<tr>
<td>C_SignInit</td>
</tr>
<tr>
<td>C_SignRecover</td>
</tr>
<tr>
<td>C_SignRecoverInit</td>
</tr>
<tr>
<td>C_SignUpdate</td>
</tr>
<tr>
<td>C_UnwrapKey</td>
</tr>
<tr>
<td>C_Verify</td>
</tr>
<tr>
<td>C_VerifyFinal</td>
</tr>
<tr>
<td>C_VerifyInit</td>
</tr>
<tr>
<td>C_VerifyRecover</td>
</tr>
<tr>
<td>C_VerifyRecoverInit</td>
</tr>
<tr>
<td>C_VerifyUpdate</td>
</tr>
<tr>
<td>C_WaitForSlotEvent</td>
</tr>
<tr>
<td>C_WrapKey</td>
</tr>
</tbody>
</table>

SUNW Extensions

<table>
<thead>
<tr>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUNW_C_GetMechSession</td>
</tr>
<tr>
<td>SUNW_C_KeyToObject</td>
</tr>
</tbody>
</table>

Files

<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/usr/lib/libpckcs11.so.1</td>
<td>shared object</td>
</tr>
<tr>
<td>/usr/lib/64/libpckcs11.so.1</td>
<td>64–bit shared object</td>
</tr>
</tbody>
</table>

Attributes

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

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcs1 (32–bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcs1x (64–bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>See below.</td>
</tr>
</tbody>
</table>
The SUNW Extension functions are Evolving. The PKCS#11 Standard functions are Standard: PKCS#11 v2.20.

The SUNW Extension functions are MT-Safe. The PKCS#11 Standard functions are MT-Safe with exceptions. See Section 6.6.2 of RSA PKCS#11 v2.20.

See Also  
cryptoadm(1M), pkgadd(1M), Intro(3), SUNW_C_GetMechSession(3EXT), syslog(3C), attributes(5), pkcs11_kernel(5), pkcs11_softtoken(5)

RSA PKCS#11 v2.20 http://www.rsasecurity.com

Notes  
If an application calls C_WaitForSlotEvent() without the CKF_DONT_BLOCK flag set, libpkcs11 must create threads internally. If, however, CKF_LIBRARY_CANT_CREATE_OS_THREADS is set, C_WaitForSlotEvent() returns CKR_FUNCTION_FAILED.

The PKCS#11 library does not work with Netscape 4.x but does work with more recent versions of Netscape and Mozilla.

Because C_Initialize() might have been called by both an application and a library, it is not safe for a library or its plugins to call C_Finalize(). A library can be finished calling functions from libpkcs11, while an application might not.
**libplot(3LIB)**

Name libplot, lib300, lib300s, lib4014, lib450, libvt0 – graphics interface libraries

**Synopsis**

```c
cc [ flag... ] file... -lplot [ library... ]
#include <plot.h>
```

**Description**

Functions in this library generate graphics output.

**Interfaces**

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

```
arc     box
circle  closepl
closevt cont
erase   label
line    linemod
move    openpl
openvt  point
space   
```

**Files**

```
/usr/lib/libplot.so.1 shared object
/usr/lib/64/libplot.so.1 64-bit shared object
/usr/lib/lib300.so.1 shared object
/usr/lib/64/lib300.so.1 64-bit shared object
/usr/lib/lib300s.so.1 shared object
/usr/lib/64/lib300s.so.1 64-bit shared object
/usr/lib/lib4014.so.1 shared object
/usr/lib/64/lib4014.so.1 64-bit shared object
/usr/lib/lib450.so.1 shared object
/usr/lib/64/lib450.so.1 64-bit shared object
/usr/lib/libvt0.so.1 shared object
/usr/lib/64/libvt0.so.1 64-bit shared object
```

**Attributes**

See `attributes(5)` for descriptions of the following attributes:
### libplot(3LIB)

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

See Also  
`pvs(1), Intro(3), attributes(5)`
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 `<pool.h>` header provides type and function declarations for all library services.

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 libpool library provides a C language API for accessing this functionality, while `pooladm(1M), poolbind(1M), and poolcfg(1M)` 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 pset resources for processor sets.

When the pools facility is enabled on a system, the behavior of the following functions is modified.

<table>
<thead>
<tr>
<th>System Call</th>
<th>Error Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>pset_assign(pset!=PS_QUERY)</td>
<td>ENOTSUP</td>
</tr>
<tr>
<td>pset_bind(pset!=PS_QUERY)</td>
<td>ENOTSUP</td>
</tr>
<tr>
<td>pset_create()</td>
<td>ENOTSUP</td>
</tr>
<tr>
<td>pset_destroy()</td>
<td>ENOTSUP</td>
</tr>
<tr>
<td>pset_setattr()</td>
<td>ENOTSUP</td>
</tr>
</tbody>
</table>

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 can be one of five types: boolean (bool), signed (int64) and unsigned (uint64) integers, floating point (double), and string values.

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 poolcfg utility, in which case an informative message is placed in the system.comment property for that configuration.
<table>
<thead>
<tr>
<th>Property name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>system.allocate-method</td>
<td>string</td>
<td>Allocation method to use when this configuration is instantiated</td>
</tr>
<tr>
<td>system.bind-default</td>
<td>bool</td>
<td>If specified pool not found, bind to pool with 'pool.default' property set to true</td>
</tr>
<tr>
<td>system.comment</td>
<td>string</td>
<td>User description of system</td>
</tr>
<tr>
<td>system.name</td>
<td>string</td>
<td>User name for the configuration</td>
</tr>
<tr>
<td>system.version</td>
<td>int64</td>
<td>Libpool version required to manipulate this configuration</td>
</tr>
<tr>
<td>system.poold.log-level</td>
<td>string</td>
<td>poold logging level</td>
</tr>
<tr>
<td>system.poold.log-location</td>
<td>string</td>
<td>poold logging location</td>
</tr>
<tr>
<td>system.poold.history-file</td>
<td>string</td>
<td>poold decision history location</td>
</tr>
<tr>
<td>system.poold.monitor-interval</td>
<td>uint64</td>
<td>poold monitoring sample interval</td>
</tr>
<tr>
<td>system.poold.objectives</td>
<td>string</td>
<td>poold objectives for a system.</td>
</tr>
</tbody>
</table>

The `system.allocate-method`, `system.bind-default`, `system.comment`, `system.name`, `system.poold.log-level`, `system.poold.log-location`, `system.poold.history-file`, `system.poold.monitor-interval`, and `system.poold.objectives` properties are writable; the `system.version` property is not.

The `system.allocate-method` property accepts only two values, “importance based” and “surplus to default”. The default value for this property is “importance based”. The property is optional and if it is not present the library will allocate resources as though it were present and had the default value. These strings are defined in `<pool.h>` as `POA_IMPORTANCE` and `POA_SURPLUS_TO_DEFAULT`.

If “importance based” allocation is defined, then during a commit the library will allocate resources to pools using an algorithm that observes minimum and maximum constraints for resources but favors those resources with greater importance.

If ”surplus to default” is defined, then during a commit the library will allocate minimum resources to all resource sets apart from default which will receive any surplus.

The `system.bind-default` property defaults to true. This property interacts with the `project.pool` resource control to specify the binding behavior for processes associated with a project. If `project.pool` is not specified, then this property has no effect. If `project.pool` is specified and the specified pool exists, this property has no effect. If the specified pool does not exist, perhaps because of a reconfiguration, then this property controls the binding behavior for the project member. If `system.bind-default` is true, then the project member is bound to the default pool (identified as the pool for which `pool.default` is true); otherwise the project
member is refused access to the system. Care should be taken with the pools configuration if this property is set to false, so as to avoid denying users access to the system.

The various poold properties are used to configure the operation of poold(1M).

The system.poold.log-level property is used to specify the level of detail provided in log messages. Valid values are: ALERT, CRIT, ERR, WARNING, NOTICE, INFO, and DEBUG.

ALERT provides the least level of detail, DEBUG the greatest. See syslog(3C) for more information about the meaning of these debug levels. If this property is not specified, the default value NOTICE is used.

The system.poold.log-location property is used to specify the location of the logfiles generated by poold. The special value of "syslog" indicates that logged messages should be written to syslog(). If this property is not specified, the default location /var/log/pool is used.

The system.poold.history-file specifies the location of the decision history file which is used by poold to improve the quality of its decision making over time. If this property is not specified, the default location /var/adm/pool is used.

The system.poold.monitor-interval property specifies the monitoring interval (in milliseconds) to be used by poold when sampling utilization statistics. If this property is not specified, the default value of 15 seconds is used.

The system.poold.objectives property specifies any system wide objectives. An objectives property has the following syntax:

```
objectives = objective [, objective] *
objective = [n:] keyword [op] [value]
```

All objectives are prefixed with an optional importance. The importance acts as a multiplier for the objective and thus increases the significance of its contribution to the objective function evaluation. If no importance is specified, the default value is 1.

The "wt-load" objective is the only objective to which a system element can be set. This objective favors configurations that match resource allocations to resource utilization. A resource set that uses more resources will be given more resources when this objective is active. An administrator should use this objective when he is relatively satisfied with the constraints established using the minimum and maximum properties and would like the DRP to manipulate resources freely within those constraints.

<table>
<thead>
<tr>
<th>Pools</th>
<th>Property name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>pool.active</td>
<td>bool</td>
<td>Mark this pool as active, if true.</td>
</tr>
<tr>
<td></td>
<td>pool.comment</td>
<td>string</td>
<td>User description of pool.</td>
</tr>
</tbody>
</table>
### Processor Sets

<table>
<thead>
<tr>
<th>Property name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pset.default</td>
<td>bool</td>
<td>Marks default processor set.</td>
</tr>
<tr>
<td>pset.load</td>
<td>uint64</td>
<td>The load for this processor set.</td>
</tr>
<tr>
<td>pset.max</td>
<td>uint64</td>
<td>Maximum number of CPUs permitted in this processor set.</td>
</tr>
<tr>
<td>pset.min</td>
<td>uint64</td>
<td>Minimum number of CPUs permitted in this processor set.</td>
</tr>
<tr>
<td>pset.name</td>
<td>string</td>
<td>User name for resource.</td>
</tr>
<tr>
<td>pset.size</td>
<td>uint64</td>
<td>Current number of CPUs in this processor set.</td>
</tr>
<tr>
<td>pset.sys_id</td>
<td>int64</td>
<td>System-assigned processor set ID.</td>
</tr>
</tbody>
</table>

### Property List

<table>
<thead>
<tr>
<th>Property name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool.default</td>
<td>bool</td>
<td>Mark this pool as the default pool, if true; see system.bind-default property.</td>
</tr>
<tr>
<td>pool.importance</td>
<td>int64</td>
<td>Relative importance of this pool; for possible resource dispute resolution.</td>
</tr>
<tr>
<td>pool.name</td>
<td>string</td>
<td>User name for pool; used by setproject(3PROJECT) as value for 'project.pool' project attribute in project(4) database.</td>
</tr>
<tr>
<td>pool.scheduler</td>
<td>string</td>
<td>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.</td>
</tr>
<tr>
<td>pool.sys_id</td>
<td>int64</td>
<td>System-assigned pool ID.</td>
</tr>
<tr>
<td>pool.temporary</td>
<td>bool</td>
<td>Mark this pool as a temporary resource; if true, this pool can exist only in the dynamic configuration and cannot be committed to a configuration file.</td>
</tr>
</tbody>
</table>

The pool.default, pool.sys_id, and pool.temporary properties are not writable; all other listed properties are writable.

If pool.scheduler is specified, it must be set to the name of a valid scheduling class for the system. See the -c option for priocntl(1) for a list of valid class names.
<table>
<thead>
<tr>
<th>Property name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pset.temporary</td>
<td>bool</td>
<td>Mark this processor set as a temporary resource; if true, this processor set can exist only in the dynamic configuration and cannot be committed to a configuration file.</td>
</tr>
<tr>
<td>pset.type</td>
<td>string</td>
<td>Names resource type; value for all processor sets is pset.</td>
</tr>
<tr>
<td>pset.units</td>
<td>string</td>
<td>Identifies meaning of size-related properties; value for all processor sets is population.</td>
</tr>
<tr>
<td>pset.poold.objectives</td>
<td>string</td>
<td>Specifies the poold objectives for a pset.</td>
</tr>
</tbody>
</table>

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

The pset.load property represents the load on a processor set. The lowest value for this property is 0. The value of pset.load increases in a linear fashion with the load on the set, as measured by the number of jobs in the system run queue.

The pset.poold.objectives property specifies an objective which is specific to a particular pset. See the system.poold.objectives entry for the specification of this property’s syntax.

There are two types of objectives that can be set on a pset:

- **locality**
  - This objective influences the impact that locality, as measured by lgroup data, has upon the chosen configuration. This objective can take one of three values:
    - **tight**
      - If set, configurations that maximize resource locality are favored.
    - **loose**
      - If set, configurations that minimize resource locality are favored.
    - **none**
      - This is the default value for this objective. If set, configuration favorability is uninfluenced by resource locality.

- **utilization**
  - This objective favors configurations that allocate resources to partitions that are failing to preserve the specified utilization objective.

These objectives are specified in terms of an operator and a value. The operators are

- `<` The “less than” operator is used to indicate that the specified value should be treated as a maximum target value.
- `>` The “greater than” operator is used to indicate that the specified value should be treated as a minimum target value.
The "about" operator is used to indicate that the specified value should be treated as a target value about which some fluctuation is acceptable.

Only one objective of each type of operator can be set. For example, if the ~ operator is set, the < and > operators cannot be set. It is possible to set a < and a > operator together; the values will be validated to ensure that they do not overlap.

<table>
<thead>
<tr>
<th>Property name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cpu.comment</td>
<td>string</td>
<td>User description of CPU.</td>
</tr>
<tr>
<td>cpu.pinned</td>
<td>bool</td>
<td>CPU pinned to this processor set.</td>
</tr>
<tr>
<td>cpu.status</td>
<td>int64</td>
<td>Processor status, on-line, offline or interrupts disabled.</td>
</tr>
<tr>
<td>cpu.sys_id</td>
<td>int64</td>
<td>System-assigned processor ID.</td>
</tr>
</tbody>
</table>

The cpu.comment, cpu.pinned, and cpu.status properties are writeable.

The cpu.status property can be set only to the following values:

- off-line: Set the CPU offline.
- on-line: Set the CPU online.
- no-intr: Disable interrupt processing on the CPU.

These values are defined in <sys/processor.h> as the PS_OFFLINE, PS_ONLINE, and PS_NOINTR macros.

**Interfaces**

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

```c
pool_associate pool_component_info
pool_component_to_elem pool_conf_alloc
pool_conf_close pool_conf_commit
pool_conf_export pool_conf_free
pool_conf_info pool_conf_location
pool_conf_open pool_conf_remove
pool_conf_rollback pool_conf_status
pool_conf_to_elem pool_conf_update
```
libpool(3LIB)

pool_conf_validate  pool_create
pool_destroy        pool_dissociate
pool_dynamic_location pool_error
pool_get_binding    pool_get_owning_resource
pool_get_pool       pool_get_property
pool_get_resource   pool_get_resource_binding
pool_get_status     pool_info
pool_put_property   pool_query_components
pool_query_pool_resources pool_query_pools
pool_query_resource_components pool_query_resources
pool_resource_create pool_resource_destroy
pool_resource_info  pool_resource_to_elem
pool_resource_transfer pool_resource_type_list
pool_resource_xtransfer pool_rm_property
pool_set_binding    pool_set_status
pool_to_static_location pool_strerror
pool_to_elem        pool_value_alloc
pool_value_free     pool_value_get_bool
pool_value_get_double pool_value_get_int64
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_int64 pool_value_set_uint64
pool_value_set_string
pool_version        pool_walk_components
pool_walk_pools     pool_walk_properties
pool_walk_resources

Files
/usr/lib/libpool.so.1 shared object
/usr/lib/64/libpool.so.1 64-bit shared object
Attributes
See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWpool (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWpoolx (64-bit)</td>
</tr>
<tr>
<td>CSI</td>
<td>Enabled</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Unstable</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

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), smf(5)

Notes
Functions in libpool can be used to manipulate static configurations even when the pools facility is not enabled. See pooladm(1M) and pool_set_status(3POOL) for more information about enabling the pools facility. The pools facility must be enabled, however, to modify the dynamic configuration.

Since the Resource Pools facility is an smf(5) service, it can also be enabled and disabled using the standard Service Management Facility (SMF) interfaces.
libproject (3LIB)

Name: libproject – project database access library

Synopsis: cc [ flag... ] file... -lproject [ library... ]
# include <project.h>

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.

endprojent
getdefaultproj
getprojbyname
getprojidbyname
project_walk
setproject

Files: /usr/lib/libproject.so.1 shared object
/usr/lib/64/libproject.so.1 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>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>

See Also: pvs(1), Intro(3), getprojent(3PROJECT), project(4), attributes(5), standards(5)
**Name**  
libpthread – POSIX threads library

**Synopsis**  
cc -mt [ flag... ] file... -lpthread [ -lrt library... ]

**Description**  
Historically, functions in this library provided POSIX threading support. See standards(5). This functionality now resides in libc(3LIB).

This library is maintained to provide backward compatibility for both runtime and compilation environments. The shared object is implemented as a filter on libc.so.1. New application development needs to specify -lpthread only to obtain POSIX semantics for fork(2) that assumes the behavior of fork1(2) rather than the default behavior that forks all threads.

**Interfaces**  
The shared object libpthread.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

```c
__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_destroy
pthread_cond_init
pthread_cond_signal
pthread_cond_signal
pthread_cond_timedwait
pthread_cond_wait
pthread_condattr_destroy
pthread_condattr_getpshared
pthread_condattr_init
pthread_condattr_setspshared
pthread_create
pthread_detach
pthread_equal
pthread_exit
pthread_getconcurrency
pthread_getschedparam
```
See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
</tbody>
</table>
### libpthread(3LIB)

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

**See Also**  
`pvs(1), Intro(2), Intro(3), libc(3LIB), libc_db(3LIB), libthread(3LIB), attributes(5), standards(5), threads(5)`
Name  libresolv–resolver library

Synopsis  cc [ flag... ] file... -lresolv -lsocket -lnsl [ library... ]
   #include <sys/types.h>
   #include <netinet/in.h>
   #include <arpa/nameser.h>
   #include <resolv.h>
   #include <netdb.h>

Description  Functions in this library provide for creating, sending, and interpreting packets to the Internet domain name servers.

Interfaces  The shared object libresolv.so.2 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

    __dn_skipname  __fp_query
    __hostalias    __p_cname
    __p_class      __p_query
    __p_type       __p_type
    __putlong      __getlong
    __getshort     __res
    dn_comp        dn_expand
    fp_resstat     h_errno
    herror         hstrerror
    res_hostalias  res_init
    res_mkquery    res_nclose
    res_ninit      res_nmkquery
    res_nquery     res_qquerydomain
    res_nsearch    res_nsend
    res_nsendsigned res_query
    res_querydomain res_search
    res_send       res_update

Programs are expected to use the aliases defined in <resolv.h> rather than calling the "__" prefixed procedures, as indicated in the following table. Use of the routines in the first column is discouraged.
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**

<table>
<thead>
<tr>
<th>Path</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/lib/libresolv.so.1</td>
<td>shared object for backward compatibility only</td>
</tr>
<tr>
<td>/lib/64/libresolv.so.1</td>
<td>64-bit shared object for backward compatibility only</td>
</tr>
<tr>
<td>/lib/libresolv.so.2</td>
<td>shared object</td>
</tr>
<tr>
<td>/lib/64/libresolv.so.2</td>
<td>64-bit shared object</td>
</tr>
</tbody>
</table>

**Attributes**

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

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Standard: BIND 8.2.4</td>
</tr>
<tr>
<td>MT-Level</td>
<td>See resolver(3RESOLV)</td>
</tr>
</tbody>
</table>

**See Also**

pvs(1), Intro(3), resolver(3RESOLV), attributes(5)
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 with 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.

```
clnttcp_create          clntudp_bufcreate
clntudp_create          get_myaddress
getrpcport              rtime
svcfd_create            svctcp_create
svcudp_bufcreate        svcudp_create
svcudp_enablecache      ...
```

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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscpu (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWscpux (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

See Also

`pvs(1), Intro(3), rpc_soc(3NSL), libnsl(3LIB), attributes(5)`
Name  librpcsvc – RPC services library

Synopsis  cc [ flag... ] file... -lrpcsvc [ library... ]
#include <rpc/rpc.h>
#include <rpcsvc/rstat.h>

Description  Functions in this library provide RPC services. See the manual pages in Section 3RPC for the individual functions.

Interfaces  The shared object librpcsvc.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

havedisk                      rnusers
rstat                          rusers
rwall                          xdr_statstime
xdr_statsvar                   xdr_utmpdlearr

Files  /lib/librpcsvc.so.1 shared object
       /lib/64/librpcsvc.so.1 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32–bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64–bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also  pvs(1), Intro(3), rstat(3RPC), attributes(5)
librsm – remote shared memory interface library

Synopsis

cc [ flag... ] file... -lrsman library... 
#include <rsmapi.h>

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.

librsm(3LIB)

Name
librsm – remote shared memory interface library

Synopsis
cc [ flag... ] file... -lrsman library... 
#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.

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_intr_signal_wait_pollfd
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

336
man pages section 3: Library Interfaces and Headers • Last Revised 22 May 2003
Files /usr/lib/librsm.so.2 shared object
/usr/lib/64/librsm.so.2 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWrsmsm (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWrsmsmx (64-bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Evolving</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also Intro(2), Intro(3), attributes(5)
librt(3LIB)

Name  librt, libposix4 – POSIX.1b Realtime Extensions library

Synopsis  cc [ flag... ] file... -lrt [ library... ]

Description  Functions in this library provide most of the interfaces specified by the POSIX.1b Realtime Extension. See standards(5). 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 libc(3LIB).

See the man pages for the individual interfaces in section 3RT for information on required headers.

The name libposix4 is maintained for backward compatibility and should be avoided. librt is the preferred name for this library.

Interfaces  The shared objects librt.so.1 and libposix4.so.1 provide the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

    aio_cancel  aio_error
    aio_fsync   aio_read
    aio_return  aio_suspend
    aio_waite   aio_write
    clock_getres clock_gettime
    clock_nanosleep clock_settime
    close       fdatasync
    fork        lio_listio
    mq_close    mq_getattr
    mq_notify   mq_open
    mq_receive  mq_reltimedreceive_np
    mq_reltimedsend_np mq_send
    mq_setattr  mq_timedreceive
    mq_timedsend mq_unlink
    nanosleep   sched_get_priority_max
    sched_get_priority_min sched_getparam
    sched_getscheduler sched_rr_get_interval
sched_setparam  sched_setscheduler
sched_yield  sem_close
sem_destroy  sem_getvalue
sem_init  sem_open
sem_post  sem_reltimedwait_np
sem_timedwait  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_waitn64  aio_write64
lio_listio64

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

Attributes

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32–bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64–bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>
See Also  pvs(1), Intro(3), libc(3LIB), attributes(5), standards(5)
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.

To use librtld_db, applications need to implement the interfaces documented in ps_pread(3PROC) and proc_service(3PROC).

Interfaces

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

rd_delete rd_errstr
rd_event_addr rd_event_enable
rd_event_getmsg rd_init
rd_loadobj_iter rd_log
rd_new rd_objpad_enable
rd_plt_resolution rd_reset

Files

/lib/librtld_db.so.1     shared object
/lib/64/librtld_db.so.1  64-bit shared object

Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also

ld.so.1(1), pvs(1), Intro(3), proc_service(3PROC), ps_pread(3PROC), rtld_db(3EXT), attributes(5)

Linker and Libraries Guide
Name  
libsasl – simple authentication and security layer library

Synopsis  
cc [ flag... ] file... -lsasl [ library... ]
#include <sasl/sasl.h>
#include <sasl/prop.h>
#include <sasl/saslutil.h>

Description  
SASL is a security framework used by connection-oriented network applications primarily for authentication. Another way to describe SASL is that it is a glue layer between a network application and some security mechanisms that allow applications to authenticate each other and provide additional security services such as data encryption. As a glue layer, SASL hides the interface specifics of the security mechanism from the application, which allows greater portability and flexibility as new security mechanisms are implemented. SASL is similar to the GSS-API in that it provides a layer of abstraction between an application and one or more security mechanisms.

/libsasl/ provides both an API for applications and an SPI for various plug-ins. To link with this library, specify -lsasl on the cc command line.

Interfaces  
The shared object /libsasl.so.1 and associated include files provide the public interfaces defined below. The *_t interfaces are function prototypes for callbacks that are defined in the public SASL header files. While /libsasl provides default versions for some of the callbacks, this structure allows an application to define its own version of the some of the callback functions.

See Intro(3) for additional information on shared object interfaces.

prop_clear  prop Dispose
prop_dup  prop_erase
prop_format  prop_get
prop_getnames  prop_new
prop_request  prop_set
prop_setvals  sasl_authorize_t
sasl_auxprop  sasl_auxprop_add_plugin
sasl_auxprop_getctx  sasl_auxprop_request
sasl_canon_user_t  sasl_canonuser_add_plugin
sasl_chalprompt_t  sasl_checkapp
sasl_checkpass  sasl_client_add_plugin
sasl_client_init  sasl_client_new
sasl_client_plug_init_t sasl_client_start
sasl_client_step sasl_decode
sasl_decode64 sasl_dispose
sasl_done sasl_encode
sasl_encode64 sasl_encodev
sasl_erasebuffer sasl_errdetail
sasl_errors sasl_errstring
sasl_getcallback_t sasl_getopt_t
sasl_getpath_t sasl_getprop
sasl_getrealm_t sasl_getsecret_t
sasl_getsimple_t sasl_global_listmech
sasl_idle sasl_listmech
sasl_log_t sasl_server_add_plugin
sasl_server_init sasl_server_new
sasl_server_plug_init_t sasl_server_start
sasl_server_step sasl_server_userdb_checkpass_t
sasl_server_userdb_setpass_t sasl_set_alloc
sasl_set_mutex sasl_seterror
sasl_setpass sasl_setprop
sasl_utf8verify sasl_verifyfile_t
sasl_version

Files /usr/lib/libsasl.so.1 shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWlibsasl</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Evolving</td>
</tr>
</tbody>
</table>

See Also Intro(3), attributes(5),
libscf – service configuration facility library

**Synopsis**
```
cc [ flag ... ] file ... -lscf [ library ... ]
#include <libscf.h>
```

**Description**
Functions in this library define the interface for reading, writing, and manipulating service configurations.

**Interfaces**
The shared object `libscf.so.1` provides the public interfaces defined below. See `Intro(3)` for additional information on shared object interfaces.

```
scf_entry_add_value       scf_entry_create
scf_entry_destroy         scf_entry_destroy_children
scf_entry_handle          scf_entry_reset
scf_error                 scf_handle_bind
scf_handle_create         scf_handle_decode_fMRI
scf_handle_decorate       scf_handle_destroy
scf_handle_get_scope      scf_handle_unbind
scf_instance_add_pg       scf_instance_create
scf_instance_delete       scf_instance_destroy
scf_instance_get_name     scf_instance_get_parent
scf_instance_get_pg       scf_instance_get_pg_composed
scf_instance_get_snapshot scf_instance_handle
scf_instance_to_fMRI      scf_iter_create
scf_iter_destroy          scf_iter_handle
scf_iter_handle_scopes    scf_iter_instance_pgs
scf_iter_instance_pgs_composed scf_iter_instance_pgs_typed_composed
scf_iter_instance_pgs_typed scf_iter_instance_snapshots
scf_iter_next_instance    scf_iter_next_pg
scf_iter_next_property    scf_iter_next_scope
scf_iter_next_service     scf_iter_next_snapshot
scf_iter_next_value       scf_iter_pg_properties
scf_iter_property_values  scf_iter_reset
```
scf_iter_scope_services  scf_iter_service_instances
scf_iter_service_pgs  scf_iter_service_pgs_typed
scf_iter_snaplevel_pgs  scf_iter_snaplevel_pgs_typed
scf_limit
scf_pg_create
scf_pg_destroy
scf_pg_get_name
scf_pg_get_parent_service
scf_pg_get_property
scf_pg_get_underlying_pg
scf_pg_to_fmri
scf_property_create
scf_property_destroy
scf_property_get_name
scf_property_get_value
scf_property_handle
scf_property_to_fmri
scf_scope_add_service
scf_scope_create
scf_scope_destroy
scf_scope_get_service
scf_scope_to_fmri
scf_service_add_instance
scf_service_add_pg
scf_service_delete
scf_service_get_instance
scf_service_get_parent
scf_service_handle
scf_service_simple_app_props_free
scf_service_simple_app_props_next
scf_service_simple_prop_free
scf_service_simple_prop_name
scf_service_simple_prop_next_boolean
scf_service_simple_prop_next_count
libscf(3LIB)

scf_simple_prop_next_integer
scf_simple_prop_next_reset
scf_simple_prop_next_ustring
scf_simple_prop_pgname
scf_simple_walk_instances
scf_snaplevel_destroy
scf_snaplevel_get_instance_name
scf_snaplevel_get_next_snaplevel
scf_snaplevel_get_parent
scf_snaplevel_get_service_name
scf_snapshot_create
scf_snapshot_get_base_snaplevel
scf_snapshot_get_name
scf_snapshot_get_parent
scf_strerror
scf_transaction_add
scf_transaction_commit
scf_transaction_destroy
scf_transaction_destroy_children
scf_transaction_handle
scf_transaction_property_change
scf_transaction_property_change_type
scf_transaction_property_delete
scf_transaction_property_new
scf_transaction_reset
scf_transaction_reset_all
scf_transaction_start
scf_type_base_type
scf_value_create
scf_value_destroy
scf_value_get_as_string
scf_value_get_astring
scf_value_get_count
scf_value_get_integer
scf_value_get_opaque
scf_value_get_ustring
scf_value_is_type
scf_value_reset
scf_value_set_astring
scf_value_set_base_type
scf_value_set_boolean
scf_value_set_count
scf_value_set_from_string
scf_value_set_integer
scf_value_set_time
scf_value_set_ustring
scf_value_set_boolean
scf_value_set_count
scf_value_set_string
The content of the image is as follows:

```
libscf(3LIB)

scf_value_set_integer  scf_value_setOpaque
scf_value_set_time     scf_value_set_ustring
scf_value_type         smf_degrade_instance
smf_disable_instance   smf_enable_instance
smf_get_state          smf_maintain_instance
smf_refresh_instance   smf_restart_instance
smf_restore_instance

Files
/usr/lib/libscf.so.1    shared object
/usr/lib/64/libscf.so.1  64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcslr</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Evolving</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also
Intro(3), attributes(5), smf(5)
```
Name  libsctp – SCTP sockets library

Synopsis  cc [ flag ]... file... -lsctp [ library ]...

Description  Functions in this library provide the SCTP socket interface.

Interfaces  The shared object libsctp.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

    sctp_bindx  sctp_freeladdr
    sctp_freepaddr  sctp_getladdr
    sctp_getpaddr  sctp_opt_info
    sctp_peeloff  sctp_recmmsg
    sctp_send  sctp_sendsmsg

Files  /usr/lib/libsctp.so.1  shared object

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

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Evolving</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also  Intro(2), Intro(3), attributes(5), sctp(7P)
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.

```
  acl_check          acl_free
  acl_fromtext       acl_get
  acl_set            acl_strip
  acl_totext         acl_trivial
  aclcheck           aclfrommode
  aclfromtext        aclsort
  acltomode          acltotext
  facl_get           facl_set
```

Files  /lib/libsec.so.1    shared object
       /lib/64/libsec.so.1  64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

See Also  pvs(1), Intro(3), attributes(5)
libsecdb(3LIB)

**Name**
libsecdb – security attributes database library

**Synopsis**
```
cc [ flag... ] file... -lsecdb [ library... ]
```
```c
#include <secdb.h>
#include <user_attr.h>
#include <prof_attr.h>
#include <exec_attr.h>
#include <auth_attr.h>
```

**Description**
Functions in this library provide routines for manipulation of security attribute databases.

**Interfaces**
The shared object `libsecdb.so.1` provides the public interfaces defined below. See `Intro(3)` for additional information on shared object interfaces.

```c
chkauthattr endauthattr
dendexecattr endprofattr
denduserattr fgetuserattr
free_authattr free_execattr
free_profattr free_proflist
free_userattr getauthattr
gethauthnam
getexecattr
getexecprof getexecuser
getprofattr getproflist
getprofnam getuserattr
get usernam getuseruid
kva_match match_execattr
setauthattr setexecattr
setprofattr setuserattr
```

**Files**

```text
/lib/libsecdb.so.1 shared object
/lib/64/libsecdb.so.1 64-bit shared object
```

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32–bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64–bit)</td>
</tr>
<tr>
<td>ATTRIBUTE TYPE</td>
<td>ATTRIBUTE VALUE</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>MT-Level</td>
<td>MT-Safe</td>
</tr>
</tbody>
</table>

**See Also**  
[Intro(3), attributes(5)]
libsendfile(3LIB)

Name  libsendfile – sendfile library

Synopsis  cc [ flag... ] file... -lsendfile [ library... ]
          #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 libsendfile.so.1 provides the public interfaces defined below. See
Intro(3) for additional information on shared object interfaces.

sendfile  sendfilev

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

sendfile64  sendfilev64

Files  /lib/libsendfile.so.1  shared object
       /lib/64/libsendfile.so.1  64–bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32–bit)</td>
</tr>
<tr>
<td></td>
<td>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>

See Also  pvs(1), Intro(3), sendfile(3EXT), sendfilev(3EXT), attributes(5)
Name  libsip – Session Initiation Protocol (SIP) library

Synopsis  cc [ flag... ] file... -lsip [ library... ]
#include <sip.h>

Description  SIP is a control protocol that can establish, modify, and terminate multimedia sessions, conferences, such as Internet telephony calls. Functions in `libsip` provide interfaces to write SIP components and applications.

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

```c
sip_add_accept  sip_add_accept_enc
sip_add_accept_lang  sip_add_alert_info
sip_add_allow  sip_add_allow_events
sip_add_authen_info  sip_add_author
sip_add_branchid_to_via  sip_add_call_info
sip_add_callid  sip_add_contact
sip_add_content  sip_add_content_disp
sip_add_content_enc  sip_add_content_lang
sip_add_content_type  sip_add_cseq
sip_add_date  sip_add_error_info
sip_add_event  sip_add_expires
sip_add_from  sip_add_header
sip_add_in_reply_to  sip_add_maxforward
sip_add_mime_version  sip_add_min_expires
sip_add_org  sip_add_param
sip_add_passertedid  sip_add_ppreferredid
sip_add_priority  sip_add_privacy
sip_add_proxy_authen  sip_add_proxy_author
sip_add_proxy_require  sip_add Rack
sip_add_record_route  sip_add_reply_to
sip_add_request_line  sip_add_require
sip_add_response_line  sip_add_retry_after
```
<table>
<thead>
<tr>
<th>Function Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>sip_get_content_length</td>
</tr>
<tr>
<td>sip_get_content_type</td>
</tr>
<tr>
<td>sip_get_counter_value</td>
</tr>
<tr>
<td>sip_get_date_day</td>
</tr>
<tr>
<td>sip_get_date_time</td>
</tr>
<tr>
<td>sip_get_date_wkday</td>
</tr>
<tr>
<td>sip_get_dialog_callid</td>
</tr>
<tr>
<td>sip_get_dialog_local_tag</td>
</tr>
<tr>
<td>sip_get_dialog_local_contact_uri</td>
</tr>
<tr>
<td>sip_get_dialog_msgcnt</td>
</tr>
<tr>
<td>sip_get_dialog_remote_tag</td>
</tr>
<tr>
<td>sip_get_dialog_remote_uri</td>
</tr>
<tr>
<td>sip_get_dialog_state</td>
</tr>
<tr>
<td>sip_get_error_info_uri</td>
</tr>
<tr>
<td>sip_get_expires</td>
</tr>
<tr>
<td>sip_get_from_tag</td>
</tr>
<tr>
<td>sip_get_header</td>
</tr>
<tr>
<td>sip_get_in_reply_to</td>
</tr>
<tr>
<td>sip_get_mime_version</td>
</tr>
<tr>
<td>sip_get_msg_len</td>
</tr>
<tr>
<td>sip_get_num_via</td>
</tr>
<tr>
<td>sip_get_param_value</td>
</tr>
<tr>
<td>sip_get_passertedid_display_name</td>
</tr>
<tr>
<td>sip_get_ppreferredid_display_name</td>
</tr>
<tr>
<td>sip_get_priority</td>
</tr>
<tr>
<td>sip_get_proxy_authen_param</td>
</tr>
<tr>
<td>sip_get_proxy_author_param</td>
</tr>
<tr>
<td>sip_get_proxy_require</td>
</tr>
<tr>
<td>sip_get_rack_method</td>
</tr>
<tr>
<td>sip_get_content_sub_type</td>
</tr>
<tr>
<td>sip_get_content</td>
</tr>
<tr>
<td>sip_get_cseq</td>
</tr>
<tr>
<td>sip_get_date_month</td>
</tr>
<tr>
<td>sip_get_date_timezone</td>
</tr>
<tr>
<td>sip_get_date_year</td>
</tr>
<tr>
<td>sip_get_dialog_local_cseq</td>
</tr>
<tr>
<td>sip_get_dialog_local_uri</td>
</tr>
<tr>
<td>sip_get_dialog_method</td>
</tr>
<tr>
<td>sip_get_dialog_remote_cseq</td>
</tr>
<tr>
<td>sip_get_dialog_remote_target_uri</td>
</tr>
<tr>
<td>sip_get_dialog_route_set</td>
</tr>
<tr>
<td>sip_get_dialog_type</td>
</tr>
<tr>
<td>sip_get_event</td>
</tr>
<tr>
<td>sip_get_from_display_name</td>
</tr>
<tr>
<td>sip_get_from_uri_str</td>
</tr>
<tr>
<td>sip_get_header_value</td>
</tr>
<tr>
<td>sip_get_maxforward</td>
</tr>
<tr>
<td>sip_get_minexpires</td>
</tr>
<tr>
<td>sip_get_next_value</td>
</tr>
<tr>
<td>sip_get_org</td>
</tr>
<tr>
<td>sip_get_params</td>
</tr>
<tr>
<td>sip_get_passertedid_uri_str</td>
</tr>
<tr>
<td>sip_get_ppreferredid_uri_str</td>
</tr>
<tr>
<td>sip_get_priv_value</td>
</tr>
<tr>
<td>sip_get_proxy_authen_scheme</td>
</tr>
<tr>
<td>sip_get_proxy_author_scheme</td>
</tr>
<tr>
<td>sip_get_rack_cseq_num</td>
</tr>
<tr>
<td>sip_get_rack_resp_num</td>
</tr>
<tr>
<td>Function</td>
</tr>
<tr>
<td>----------------------------------------------</td>
</tr>
<tr>
<td>sip_get_replyto_display_name</td>
</tr>
<tr>
<td>sip_get_request_method</td>
</tr>
<tr>
<td>sip_get_require</td>
</tr>
<tr>
<td>sip_get_response_code</td>
</tr>
<tr>
<td>sip_get_retry_after_cmts</td>
</tr>
<tr>
<td>sip_get_route_display_name</td>
</tr>
<tr>
<td>sip_get_rseq</td>
</tr>
<tr>
<td>sip_get_server</td>
</tr>
<tr>
<td>sip_get_subject</td>
</tr>
<tr>
<td>sip_get_supported</td>
</tr>
<tr>
<td>sip_get_to_tag</td>
</tr>
<tr>
<td>sip_get_trans</td>
</tr>
<tr>
<td>sip_get_trans_conn_obj</td>
</tr>
<tr>
<td>sip_get_trans_orig_msg</td>
</tr>
<tr>
<td>sip_get_trans_state</td>
</tr>
<tr>
<td>sip_get_tstamp_value</td>
</tr>
<tr>
<td>sip_get_uri_errflags</td>
</tr>
<tr>
<td>sip_get_uri_host</td>
</tr>
<tr>
<td>sip_get_uri_params</td>
</tr>
<tr>
<td>sip_get_uri_password</td>
</tr>
<tr>
<td>sip_get_uri_port</td>
</tr>
<tr>
<td>sip_get_uri_regname</td>
</tr>
<tr>
<td>sip_get_uri_user</td>
</tr>
<tr>
<td>sip_get_via_sent_by_host</td>
</tr>
<tr>
<td>sip_get_via_sent_protocol_name</td>
</tr>
<tr>
<td>sip_get_via_sent_transport</td>
</tr>
<tr>
<td>sip_get_warning_code</td>
</tr>
<tr>
<td>sip_get_www_authen_param</td>
</tr>
<tr>
<td>sip_guid</td>
</tr>
</tbody>
</table>
**libsip(3LIB)**

- `sip_hold_dialog`  
- `sip_hold_trans`  
- `sip_is_dialog_secure`  
- `sip_is_uri`  
- `sip_msg_is_request`  
- `sip_msg_to_str`  
- `sip_parse_uri`  
- `sip_register_sent_by`  
- `sip_release_trans`  
- `sip_respline_to_str`  
- `sip_sent_by_to_str`  
- `sip_unregister_all_sent_by`  
- `sip_uri_errflags_to_str`  

**Files**

- `/lib/libsip.so.1` shared object
- `/lib/64/libsip.so.1` 64-bit shared object

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Committed</td>
</tr>
<tr>
<td>MT-Level</td>
<td>MT-Safe</td>
</tr>
</tbody>
</table>

**See Also**

Intro(3), attributes(5)
**Name**  libslp – service location protocol library

**Synopsis**  

```c
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
```

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWslpu</td>
</tr>
</tbody>
</table>

**See Also**  `pvs(1), Intro(2), Intro(3), attributes(5)}`
libsmartcard – smartcard library

Synopsis

cc [...] 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.

<table>
<thead>
<tr>
<th>Interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCF_Card_close</td>
</tr>
<tr>
<td>SCF_Card_exchangeAPDU</td>
</tr>
<tr>
<td>SCF_Card_freeInfo</td>
</tr>
<tr>
<td>SCF_Card_getInfo</td>
</tr>
<tr>
<td>SCF_Card_lock</td>
</tr>
<tr>
<td>SCF_Card_reset</td>
</tr>
<tr>
<td>SCF_Card_unlock</td>
</tr>
<tr>
<td>SCF_Card_waitForCardRemoved</td>
</tr>
<tr>
<td>SCF_Session_close</td>
</tr>
<tr>
<td>SCF_Session_freeInfo</td>
</tr>
<tr>
<td>SCF_Session_getInfo</td>
</tr>
<tr>
<td>SCF_Session_getSession</td>
</tr>
<tr>
<td>SCF_Session_getTerminal</td>
</tr>
<tr>
<td>SCF_Terminal_addEventListener</td>
</tr>
<tr>
<td>SCF_Terminal_close</td>
</tr>
<tr>
<td>SCF_Terminal_freeInfo</td>
</tr>
<tr>
<td>SCF_Terminal_getCard</td>
</tr>
<tr>
<td>SCF_Terminal_getInfo</td>
</tr>
<tr>
<td>SCF_Terminal_removeEventListener</td>
</tr>
<tr>
<td>SCF_Terminal_waitForCardAbsent</td>
</tr>
<tr>
<td>SCF_Terminal_waitForCardPresent</td>
</tr>
<tr>
<td>SCF_strerror</td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWocf</td>
</tr>
<tr>
<td>Availability</td>
<td>SUNWocf(32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWocfx(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>
See Also   smartcard(1M), Intro(3), attributes(5), smartcard(5)
Name  libsocket – sockets library

Synopsis  cc [ flag... ] file... -lsocket [ library... ]

Description  Functions in this library provide the socket internetworking interface, primarily used with the TCP/IP protocol suite.

Interfaces  The shared object libsocket.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

```
__xnet_bind  __xnet_connect
__xnet_getsockopt  __xnet_listen
__xnet_recvmsg  __xnet_sendmsg
__xnet_sendto  __xnet_socket
__xnet_socketpair  accept
bind  connect
endnetent  endprotoent
endservent  ether_aton
ether_hostton  ether_line
ether_ntoa  ether_ntohost
freeaddrinfo  gai_strerror
getaddrinfo  getnameinfo
getnetbyaddr  getnetbyaddr_r
getnetbyname  getnetbyname_r
getnetent  getnetent_r
getpeername  getprotobyname
getprotobyname  getprotobyname_r
getprotobynumber  getprotobynumber_r
getprotobynumber_r  getprotoent
getprotoent_r  getservbyname
getservbyname  getservbyname_r
getservbyport  getservbyport
getservbyport_r  getservent
getservent_r  getsockname
getsockopt  htonl
```
htons if_freenamindex
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
rexe rexec_rexc
rresvport rresvport_af
ruserok send
sendmsg sendto
setnetent setprotoent
setservent setsockopt
shutdown socket
socketpair

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

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcs (32–bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcsix (64–bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>See the manual page for each interface.</td>
</tr>
</tbody>
</table>

See Also
pvs(1), Intro(2), Intro(3), socket.h(3HEAD), attributes(5)
The `libssagent` library is a high level API library that is dependent on `libssasnmp`. 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.

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

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSAGetTrapPort</td>
<td></td>
</tr>
<tr>
<td>SSARegSubagent</td>
<td></td>
</tr>
<tr>
<td>_SSASendTrap</td>
<td></td>
</tr>
<tr>
<td>_SSASendTrap2</td>
<td></td>
</tr>
<tr>
<td>_SSASendTrap3</td>
<td></td>
</tr>
<tr>
<td>callItem</td>
<td></td>
</tr>
<tr>
<td>numCallItem</td>
<td></td>
</tr>
<tr>
<td>numTrapElem</td>
<td></td>
</tr>
<tr>
<td>trapAnyEnterpriseInfo</td>
<td></td>
</tr>
<tr>
<td>trapBucket</td>
<td></td>
</tr>
<tr>
<td>trapEnterpriseInfo</td>
<td></td>
</tr>
<tr>
<td>trapTableMap</td>
<td></td>
</tr>
</tbody>
</table>

**Files**
- `/usr/lib/libssagent.so.1` shared object
- `/usr/lib/64/libssagent.so.1` 64-bit shared object

**Attributes**

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWsasnm</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Obsolete</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

**See Also**
- `Intro(3)`, `libssasnmp(3LIB)`, `attributes(5)`
Name libssasnmp – Sun Solstice Enterprise SNMP library

Synopsis cc [ flag... ] file... -lssasnmp [ library... ]

Description The libssasnmp library provides low-level SNMP API functions.

- 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 The shared object libssasnmp.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSAOidCmp</td>
<td>SSAOidCpy</td>
</tr>
<tr>
<td>SSAOidDup</td>
<td>SSAOidFree</td>
</tr>
<tr>
<td>SSAOidInit</td>
<td>SSAOidNew</td>
</tr>
<tr>
<td>SSAOidToStr0id</td>
<td>SSAOidString</td>
</tr>
<tr>
<td>SSAOidZero</td>
<td>SSAStringCpy</td>
</tr>
<tr>
<td>SSAStringInit</td>
<td>SSAStringToChar</td>
</tr>
<tr>
<td>SSAStringZero</td>
<td></td>
</tr>
</tbody>
</table>

Files /usr/lib/libssasnmp.so.1 shared object

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

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

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWsasnm</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Obsolete</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

See Also Intro(3), libssagent(3LIB), attributes(5)
Function in this library provide basic system services. This library is implemented as a filter on the C library (see \texttt{libc(3LIB)}).

**Interfaces** The shared object \texttt{libsys.so.1} provides the public interfaces defined below. See \texttt{Intro(3)} for additional information on shared object interfaces.

```
__ctype __huge_val __access
_acct _alarm _altzone
_catclose _catgets _catopen
_chdir _chmod _chown
_chroot _close _closedir
_creat _daylight _dup
_environ _excei _execle
_exel _execv _execve
_exelp _execvp _execve
_exel _execvp _exit _fattach
_fchdir _fchmod _fchown
_fcntl _fdetach _fork
_fpathconf _fstat _fstatvfs
_fsynchron _fsynck _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
_lseek _lstat _makecontext
```
<table>
<thead>
<tr>
<th>Function(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>_memcntl</td>
</tr>
<tr>
<td>_mkdir</td>
</tr>
<tr>
<td>_mknod</td>
</tr>
<tr>
<td>_mlock</td>
</tr>
<tr>
<td>_mmap</td>
</tr>
<tr>
<td>_mount</td>
</tr>
<tr>
<td>_mprotect</td>
</tr>
<tr>
<td>_msgctl</td>
</tr>
<tr>
<td>_msgget</td>
</tr>
<tr>
<td>_msgrcv</td>
</tr>
<tr>
<td>_msgsnd</td>
</tr>
<tr>
<td>_msync</td>
</tr>
<tr>
<td>_munlock</td>
</tr>
<tr>
<td>_munmap</td>
</tr>
<tr>
<td>_nice</td>
</tr>
<tr>
<td>_numeric</td>
</tr>
<tr>
<td>_open</td>
</tr>
<tr>
<td>_opendir</td>
</tr>
<tr>
<td>_pathconf</td>
</tr>
<tr>
<td>_pause</td>
</tr>
<tr>
<td>_pipe</td>
</tr>
<tr>
<td>_poll</td>
</tr>
<tr>
<td>_profil</td>
</tr>
<tr>
<td>_ptrace</td>
</tr>
<tr>
<td>_ptsname</td>
</tr>
<tr>
<td>_putmsg</td>
</tr>
<tr>
<td>_putpmsg</td>
</tr>
<tr>
<td>_read</td>
</tr>
<tr>
<td>_readdir</td>
</tr>
<tr>
<td>_readlink</td>
</tr>
<tr>
<td>_readv</td>
</tr>
<tr>
<td>_rename</td>
</tr>
<tr>
<td>_rewinddir</td>
</tr>
<tr>
<td>_rmdir</td>
</tr>
<tr>
<td>_seekdir</td>
</tr>
<tr>
<td>_semct1</td>
</tr>
<tr>
<td>_semget</td>
</tr>
<tr>
<td>_semop</td>
</tr>
<tr>
<td>_setcontext</td>
</tr>
<tr>
<td>_setgid</td>
</tr>
<tr>
<td>_setgroups</td>
</tr>
<tr>
<td>_setpgid</td>
</tr>
<tr>
<td>_setpgrp</td>
</tr>
<tr>
<td>_setrlimit</td>
</tr>
<tr>
<td>_setsid</td>
</tr>
<tr>
<td>_setuid</td>
</tr>
<tr>
<td>_shmat</td>
</tr>
<tr>
<td>_shmct1</td>
</tr>
<tr>
<td>_shmdt</td>
</tr>
<tr>
<td>_shmget</td>
</tr>
<tr>
<td>_sigaction</td>
</tr>
<tr>
<td>_sigaddset</td>
</tr>
<tr>
<td>_sigaltstack</td>
</tr>
<tr>
<td>_sigdelset</td>
</tr>
<tr>
<td>_sigemptyset</td>
</tr>
<tr>
<td>_sigfillset</td>
</tr>
<tr>
<td>_sighold</td>
</tr>
<tr>
<td>_sigignore</td>
</tr>
<tr>
<td>_sigismember</td>
</tr>
<tr>
<td>_siglongjmp</td>
</tr>
<tr>
<td>_sigpause</td>
</tr>
<tr>
<td>_sigpending</td>
</tr>
<tr>
<td>_sigprocmask</td>
</tr>
<tr>
<td>_sigreelse</td>
</tr>
<tr>
<td>_sigsend</td>
</tr>
<tr>
<td>_sigsendset</td>
</tr>
<tr>
<td>_sigset</td>
</tr>
<tr>
<td>_sigsetjmp</td>
</tr>
<tr>
<td>_sigsuspend</td>
</tr>
<tr>
<td>_stat</td>
</tr>
<tr>
<td>_statvfs</td>
</tr>
<tr>
<td>_stime</td>
</tr>
<tr>
<td>_swapcontext</td>
</tr>
<tr>
<td>_symlink</td>
</tr>
<tr>
<td>_sync</td>
</tr>
<tr>
<td>_sysconf</td>
</tr>
<tr>
<td>_telldir</td>
</tr>
<tr>
<td>_time</td>
</tr>
<tr>
<td>_timezone</td>
</tr>
<tr>
<td>_ttyname</td>
</tr>
<tr>
<td>_tzname</td>
</tr>
<tr>
<td>_ulimit</td>
</tr>
<tr>
<td>_umask</td>
</tr>
<tr>
<td>_umount</td>
</tr>
<tr>
<td>_uname</td>
</tr>
<tr>
<td>_unlink</td>
</tr>
<tr>
<td>Function</td>
</tr>
<tr>
<td>--------------------------</td>
</tr>
<tr>
<td>_unlockpt</td>
</tr>
<tr>
<td>_waitid</td>
</tr>
<tr>
<td>_writev</td>
</tr>
<tr>
<td>alarm</td>
</tr>
<tr>
<td>catclose</td>
</tr>
<tr>
<td>chdir</td>
</tr>
<tr>
<td>chroot</td>
</tr>
<tr>
<td>creat</td>
</tr>
<tr>
<td>environ</td>
</tr>
<tr>
<td>execvp</td>
</tr>
<tr>
<td>exit</td>
</tr>
<tr>
<td>fchdir</td>
</tr>
<tr>
<td>fcntl</td>
</tr>
<tr>
<td>fpathconf</td>
</tr>
<tr>
<td>fstatvfs</td>
</tr>
<tr>
<td>getcontext</td>
</tr>
<tr>
<td>geteuid</td>
</tr>
<tr>
<td>getenam</td>
</tr>
<tr>
<td>getmsg</td>
</tr>
<tr>
<td>getpnam</td>
</tr>
<tr>
<td>getpid</td>
</tr>
<tr>
<td>getsid</td>
</tr>
<tr>
<td>grantpt</td>
</tr>
<tr>
<td>isastream</td>
</tr>
<tr>
<td>link</td>
</tr>
<tr>
<td>lstat</td>
</tr>
<tr>
<td>memcntl</td>
</tr>
<tr>
<td>nlock</td>
</tr>
<tr>
<td>mprotect</td>
</tr>
<tr>
<td>Function</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>msgrcv</td>
</tr>
<tr>
<td>munlock</td>
</tr>
<tr>
<td>open</td>
</tr>
<tr>
<td>pause</td>
</tr>
<tr>
<td>profil</td>
</tr>
<tr>
<td>putmsg</td>
</tr>
<tr>
<td>readdir</td>
</tr>
<tr>
<td>realloc</td>
</tr>
<tr>
<td>rewinddir</td>
</tr>
<tr>
<td>semctl</td>
</tr>
<tr>
<td>setcontext</td>
</tr>
<tr>
<td>setlocale</td>
</tr>
<tr>
<td>setrlimit</td>
</tr>
<tr>
<td>shmat</td>
</tr>
<tr>
<td>shmget</td>
</tr>
<tr>
<td>sigaltstack</td>
</tr>
<tr>
<td>sigfillset</td>
</tr>
<tr>
<td>sigismember</td>
</tr>
<tr>
<td>sigpause</td>
</tr>
<tr>
<td>sigreelse</td>
</tr>
<tr>
<td>sigset</td>
</tr>
<tr>
<td>stat</td>
</tr>
<tr>
<td>strcoll</td>
</tr>
<tr>
<td>strxfrm</td>
</tr>
<tr>
<td>sync</td>
</tr>
<tr>
<td>telldir</td>
</tr>
<tr>
<td>timezone</td>
</tr>
<tr>
<td>ulimit</td>
</tr>
<tr>
<td>uname</td>
</tr>
</tbody>
</table>
The following interfaces are unique to the SPARC version of this library:

```
div  mul  rem
stret1  stre2  stre4
stret8  udiv  umul
urem  _Q_add  _Q_cmp
_Q_cmpe  _Q_div  _Q_dtoq
_Q_feq  _Q_fge  _Q_fgt
_Q_fle  _Q_flt  _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
```

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

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>
See Also    pvs(1), Intro(2), Intro(3), libc(3LIB), attributes(5)
Name  libsysevent – system event interface library

Synopsis  cc [ flag... ] file... -lsysevent [ library... ]
          #include <sysevent.h>

Description  Functions in this library extract specific identifier, publisher, and attribute information from a system event (sysevent) handle, defined as sysevent_t, and allow privileged user-level applications to queue system events for delivery to the system event daemon, syseventd(1M).

The libsysevent interfaces do not work at all in non-global zones.

Interfaces  The shared object libsysevent.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

    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

Files  /usr/lib/libsysevent.so.1  shared object
       /usr/lib/64/libsysevent.so.1  64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32–bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64–bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Committed</td>
</tr>
<tr>
<td>MT-Level</td>
<td>MT-Safe</td>
</tr>
</tbody>
</table>

See Also  syseventd(1M), Intro(3), attributes(5)
libtecla – interactive command line input library

Synopsis  
cc [ flag... ] file... -ltecla [ library... ]
#include <libtecla.h>

Description  
This library provides programs with interactive command line editing facilities, similar to those of the UNIX tcsh shell. In addition to simple command-line editing, it supports recall of previously entered command lines, TAB completion of file names or other tokens, and in-line wildcard expansion of filenames. The internal functions that perform file-name completion and wild-card expansion are also available externally for optional use by the calling program.

Thread Safety  
The terminfo functions setupterm(3CURSES), tigetstr(3CURSES), tigetnum(3CURSES), and tputs(3CURSES) are not reentrant. This condition, however, should not prevent use of this library in threaded applications, since few applications will want to interact with multiple terminals.

Interfaces  
The shared object libtecla.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

cfc_file_start  cfc_literal_escapes

cfc_set_check_fn  cpl_add_completion

cpl_check_exe  cpl_complete_word

cpl_file_completions  cpl_last_error

cpl_list_completions  cpl_recall_matches

cpl_record_error  del_CplFileConf

del_ExpandFile  del_GetLine

del_PathCache  del_PcaPathConf

del_WordCompletion  ef_expand_file

ef_last_error  ef_list_expansions

gl_abandon_line  gl_append_history

gl_automatic_history  gl_bind_keyseq

gl_catch_blocked  gl_change_terminal

gl_clear_history  gl_completion_action

gl_configure_getline  gl_customize_completion

gl_display_text  gl_echo_mode

gl_erase_terminal  gl_error_message
### Files
- /usr/lib/libtecla.so.1  
  shared object
- /usr/lib/64/libtecla.so.1  
  64–bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl</td>
</tr>
</tbody>
</table>
libtecla(3LIB)

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Evolving</td>
</tr>
<tr>
<td>MT-Level</td>
<td>MT-Safe</td>
</tr>
</tbody>
</table>

See Also  enhance(1), Intro(3), cpl_complete_word(3TECLA), ef_expand_file(3TECLA), gl_get_line(3TECLA), gl_io_mode(3TECLA), pca_lookup_file(3TECLA), attributes(5), tecla(5)
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.so.1 shared object
/usr/ucblib/64/libtermcap.so.1 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

See Also Intro(3), curs_termcap(3CURSES), terminfo(4), attributes(5)
libthread(3LIB)

Name libthread – threads library

Synopsis cc -mt [ flag... ] file... [ library... ]

Description Historically, functions in libthread provided threading support. This functionality now resides in libc(3LIB).

This library is maintained to provide backward compatibility for both runtime and compilation environments. The shared object is implemented as a filter on libc.so.1. New application development need not specify -lthread.

Interfaces The shared object libthread.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

```
cond_broadcast    cond_destroy
cond_init         cond_reltimedwait
cond_signal       cond_timedwait
cond_wait         mutex_destroy
mutex_init        mutex_lock
mutex_trylock     mutex_unlock
rw_rdlock         rw_tryrdlock
rw_trywrlock      rw_unlock
rw_wrlock         rwlock_destroy
rwlock_init       sema_destroy
sema_init         sema_post
sema_trywait      sema_wait
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**

**Files**
- `/lib/libthread.so.1` a filter on `libc.so.1`
- `/lib/64/libthread.so.1` a filter on `64/libc.so.1`

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

**See Also** pvs(1), Intro(2), Intro(3), libc(3LIB), libc_db(3LIB), libpthread(3LIB), attributes(5), threads(5)
**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.

- tnfctl_buffer_alloc
- tnfctl_buffer_dealloc
- tnfctl_check_libs
- tnfctl_close
- tnfctl_continue
- tnfctl_exec_open
- tnfctl_filter_list_add
- tnfctl_filter_list_delete
- tnfctl_filter_list_get
- tnfctl_filter_state_set
- tnfctl_indirect_open
- tnfctl_internal_open
- tnfctl_kernel_open
- tnfctl_pid_open
- tnfctl_probe_apply
- tnfctl_probe_apply_ids
- tnfctl_probe_connect
- tnfctl_probe_disable
- tnfctl_probe_disconnect_all
- tnfctl_probe_enable
- tnfctl_probe_get
- tnfctl_probe_untrace
- tnfctl_register_funcs
- tnfctl_strerror
- tnfctl_trace_attrs_get
- tnfctl_trace_state_set

**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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWtnfc (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWtnfxc (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>MT-Safe with exceptions</td>
</tr>
</tbody>
</table>
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.
libtsalarm – Telco-Alarm library

Synopsis  cc [ flag... ] file... -ltsalarm [ library... ]
         #include <tsalarm.h>

Description Functions in this library are used to interface with the service processor through telco-alarm LDC channel to get or set status of telco alarms.

Interfaces The shared object libtsalarm.so.1 provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

        tsalarm_get                     tsalarm_set

Files  /usr/platform/"uname -i")/lib/libtsalarm.so.1
        shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWkvmt200.v</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Uncommitted</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also  tsalarm_get(3EXT), attributes(5)
Name: libtsnet – Solaris Trusted Extensions network library

Synopsis: cc [ flag... ] file... [ library... ]
#include <libtsnet.h>
#include <sys/tsol/tndb.h>

Description: Functions in this library provide programmatic access to Solaris Trusted Extensions features such as labels and Mandatory Access Policy (MAC). These functions are available on systems that are configured with Trusted Extensions software.

Interfaces: The shared object libtsnet.so.1 provides the public interfaces that are defined below. See Intro(3) for additional information on shared object interfaces.

tsol_getrtype

Files: /lib/libtsnet.so.1 shared object
/lib/64/libtsnet.so.1 64–bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Stable</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

See Also: Intro(3), libtsol(3LIB), attributes(5)
# NAME
libtsol – Solaris Trusted Extensions library

# SYNOPSIS
cc [flag...] file... -ltsol [library...]
#include <tsol.h>

# DESCRIPTION
Functions in this library provide programmatic access to Solaris Trusted Extensions features such as labels and Mandatory Access Policy (MAC) on systems that are configured with Trusted Extensions software.

# Interfaces
The shared object libtsol.so.2 provides the public interfaces that are defined below. See Intro(3) for additional information on shared object interfaces.

## Committed Functions
- bldominates
- bldistictdom
- getlabel
- getzoneidbylabel
- getzoneroootbyid
- getzoneroootbyname
- label_to_str
- labelbuilder
- m_label_alloc
- m_label_free
- str_to_label
- tsol_lbuild_create
- tsol_lbuild_destroy
- tsol_lbuild_set
- Xbclearotos

## Obsolete Functions
- The following functions are preserved to aid porting.

<table>
<thead>
<tr>
<th>Function</th>
<th>Committed Replacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>bcleartoh</td>
<td>label_to_str</td>
</tr>
<tr>
<td>bcleartoh_r</td>
<td>label_to_str</td>
</tr>
<tr>
<td>bcleartos</td>
<td>label_to_str</td>
</tr>
<tr>
<td>bltcolor</td>
<td>label_to_str</td>
</tr>
<tr>
<td>bltcolor_r</td>
<td>label_to_str</td>
</tr>
<tr>
<td>bsltoh</td>
<td>label_to_str</td>
</tr>
</tbody>
</table>

libtsol(3LIB)
### Function Committed Replacement

<table>
<thead>
<tr>
<th>Function</th>
<th>Committed Replacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>bsltoh_r</td>
<td>label_to_str</td>
</tr>
<tr>
<td>bsltos</td>
<td>label_to_str</td>
</tr>
<tr>
<td>h_alloc</td>
<td>label_to_str</td>
</tr>
<tr>
<td>h_free</td>
<td>label_to_str</td>
</tr>
<tr>
<td>hto_bclear</td>
<td>str_to_label</td>
</tr>
<tr>
<td>hto_bsl</td>
<td>str_to_label</td>
</tr>
<tr>
<td>sbcleartos</td>
<td>str_to_label</td>
</tr>
<tr>
<td>sbsltos</td>
<td>str_to_label</td>
</tr>
<tr>
<td>stobsl</td>
<td>str_to_label</td>
</tr>
<tr>
<td>stob_clear</td>
<td>str_to_label</td>
</tr>
</tbody>
</table>

### Files

- `/lib/libtsol.so.2` shared object
- `/lib/64/libtsol.so.2` 64-bit shared object

### Attributes

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

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>system/library</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>See the manual pages for the individual functions.</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

### See Also

Intro(3), libtsnet(3LIB), attributes(5)

### Notes

The functionality described on this manual page is available only if the system has been configured with Trusted Extensions.
libucb(3LIBUCB)

**Name**  
libucb – UCB source compatibility library

**Synopsis**  
cc [ flag...] -I /usr/ucbinclude file... -L /usr/libucb 
-R /usr/ucblib -lucb [ library... ]

**Description**  
Functions in this library provide UCB source compatibility.

**Interfaces**  
The shared object libucb.so.1 provides the public interfaces defined below. See `Intro(3)` for additional information on shared object interfaces.

<table>
<thead>
<tr>
<th>Name</th>
<th>Synopsis</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>alphasort</td>
<td>bcmp</td>
<td>bcopy</td>
</tr>
<tr>
<td>bzero</td>
<td>flock</td>
<td>fopen</td>
</tr>
<tr>
<td>fprintf</td>
<td>freopen</td>
<td>fstatfs</td>
</tr>
<tr>
<td>ftime</td>
<td>getdtablesize</td>
<td>gethostid</td>
</tr>
<tr>
<td>gethostname</td>
<td>getpagesize</td>
<td>getususage</td>
</tr>
<tr>
<td>gettimeofday</td>
<td>getwd</td>
<td>index</td>
</tr>
<tr>
<td>killpg</td>
<td>longjmp</td>
<td>mctl</td>
</tr>
<tr>
<td>nice</td>
<td>nlist</td>
<td>printf</td>
</tr>
<tr>
<td>psignal</td>
<td>rand</td>
<td>re_comp</td>
</tr>
<tr>
<td>re_exec</td>
<td>readdir</td>
<td>reboot</td>
</tr>
<tr>
<td>rindex</td>
<td>scandir</td>
<td>setbuffer</td>
</tr>
<tr>
<td>sethostname</td>
<td>setjmp</td>
<td>setlinebuf</td>
</tr>
<tr>
<td>setpgid</td>
<td>settimeofday</td>
<td>sigblock</td>
</tr>
<tr>
<td>siginterrupt</td>
<td>signal</td>
<td>sigpause</td>
</tr>
<tr>
<td>sigsetmask</td>
<td>sigstack</td>
<td>sigvec</td>
</tr>
<tr>
<td>sigvecandler</td>
<td>sleep</td>
<td>sprintf</td>
</tr>
<tr>
<td>srand</td>
<td>statfs</td>
<td>sys_siglist</td>
</tr>
<tr>
<td>times</td>
<td>ualarm</td>
<td>usignal</td>
</tr>
<tr>
<td>usigpause</td>
<td>usleep</td>
<td>vfprintf</td>
</tr>
<tr>
<td>vprintf</td>
<td>vsprintf</td>
<td>wait3</td>
</tr>
<tr>
<td>wait4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The following interfaces are unique to the 32-bit version of this library:
alphasort64 fopen64 freopen64
readdir64 scandir64

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWscpu, SUNWsra (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWscpux (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe with exceptions</td>
</tr>
</tbody>
</table>

**See Also** `pvs(1), Intro(3), attributes(5)`
libumem - object-caching memory allocation library

Synopsis

cc [ flag... ] file... -lumem [ library... ]
#include <umem.h>

Description

Functions in this library provide fast, scalable object-caching memory allocation with multithreaded application support. In addition to the standard malloc(3C) family of functions and the more flexible umem_alloc(3MALLOC) family, libumem provides powerful object-caching services as described in umem_cache_create(3MALLOC).

The libumem 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 umem_debug(3MALLOC).

Interfaces

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

    calloc  free
    malloc  memalign
    realloc  umem_alloc
    umem_cache_alloc  umem_cache_create
    umem_cache_destroy  umem_cache_free
    umem_free  umemnofail_callback
    umem_zalloc  valloc

Files

/usr/lib/libumem.so.1     shared object
/usr/lib/64/libumem.so.1   64–bit shared object

Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>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>

See Also

Intro(3), malloc(3C), umem_alloc(3MALLOC), umem_cache_create(3MALLOC), umem_debug(3MALLOC), attributes(5)
## Name
libusb – user-space USB device management library

## Synopsis
```
cc [ flag... ] -I/usr/sfw/include file... -L/usr/sfw/lib \
    -R /usr/sfw/lib -lusb [ library... ]
#include <usb.h>
```

## Description
The `libusb` library contains interfaces for managing USB devices without a kernel driver. It is an open-source API supported on Linux, MacOS X, and NetBSD. See [http://libusb.sourceforge.net](http://libusb.sourceforge.net).

The current implementation is version 0.1.8 of the `libusb` API.

Complete documentation for this library can be found at `/usr/sfw/share/doc/libusb/libusb.txt`.

## Interfaces
The shared object `libusb.so.1` provides the following public interfaces. See `Intro(3)` for additional information on shared object interfaces.

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>usb_bulk_read</td>
<td></td>
</tr>
<tr>
<td>usb_bulk_write</td>
<td></td>
</tr>
<tr>
<td>usb_claim_interface</td>
<td></td>
</tr>
<tr>
<td>usb_clear_halt</td>
<td></td>
</tr>
<tr>
<td>usb_close</td>
<td></td>
</tr>
<tr>
<td>usb_control_msg</td>
<td></td>
</tr>
<tr>
<td>usb_find_busses</td>
<td></td>
</tr>
<tr>
<td>usb_find_devices</td>
<td></td>
</tr>
<tr>
<td>usb_get_busses</td>
<td></td>
</tr>
<tr>
<td>usb_get_descriptor</td>
<td></td>
</tr>
<tr>
<td>usb_get_descriptor_by_endpoint</td>
<td></td>
</tr>
<tr>
<td>usb_get_string</td>
<td></td>
</tr>
<tr>
<td>usb_get_string_simple</td>
<td></td>
</tr>
<tr>
<td>usb_init</td>
<td></td>
</tr>
<tr>
<td>usb_interrupt_read</td>
<td></td>
</tr>
<tr>
<td>usb_interrupt_write</td>
<td></td>
</tr>
<tr>
<td>usb_open</td>
<td></td>
</tr>
<tr>
<td>usb_release_interface</td>
<td></td>
</tr>
<tr>
<td>usb_reset</td>
<td></td>
</tr>
<tr>
<td>usb_resetep</td>
<td></td>
</tr>
<tr>
<td>usb_set_altinterface</td>
<td></td>
</tr>
<tr>
<td>usb_set_configuration</td>
<td></td>
</tr>
<tr>
<td>usb_set_debug</td>
<td></td>
</tr>
<tr>
<td>usb_strerror</td>
<td></td>
</tr>
<tr>
<td>/usr/sfw/lib/libusb.so.1</td>
<td>shared object</td>
</tr>
<tr>
<td>/usr/sfw/lib/libusb_plugins</td>
<td>implementation-specific <code>libusb</code> modules</td>
</tr>
<tr>
<td>/usr/sfw/bin/libusb-config</td>
<td>script to determine linking environment</td>
</tr>
</tbody>
</table>

## Files
```
/usr/sfw/lib/libusb.so.1
/usr/sfw/lib/libusb_plugins
/usr/sfw/bin/libusb-config
```

## Attributes
See `attributes(5)` for descriptions of the following attributes:
### libusb(3LIB)

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWlibusb, SUNWlibusbut, SUNWlibugenusb</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>External</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

**See Also**  
Intro(3), attributes(5)

http://libusb.sourceforge.net
**Name** libuuid – UUID library

**Synopsis**
```
cc [ flag... | file... -luuid [ library... ]
#include <uuid/uuid.h>
```

**Description** The functions in this library perform operations on a universally unique identifier (UUID).

**Interfaces** The shared object libuuid.so.1 provides the public interfaces defined below. See `Intro(3)` for additional information on shared object interfaces.

- `uuid_clear`
- `uuid_compare`
- `uuid_copy`
- `uuid_generate`
- `uuid_generate_random`
- `uuid_generate_time`
- `uuid_is_null`
- `uuid_parse`
- `uuid_time`
- `uuid_unparse`

**Files**
- `/lib/libuuid.so.1` shared object
- `/lib/64/libuuid.so.1` 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Committed</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

**See Also** `Intro(3), uuid_clear(3UUID), attributes(5)`
libv12n(3LIB)

**Name**  libv12n – virtualization domain information interface library

**Synopsis**  cc [ flag... ] file... -lv12n [ library... ]

#include <libv12n.h>

The functions in this library extract specific virtualization domain information. For Logical Domains, this information comes from one of the following:

- Domain's machine description
- Domain service of the control domain that is provided by the Logical Domains agents daemon (ldmad)

**Interfaces**  The `libv12n.so.1` shared object provides the public interfaces defined below. See `Intro(3)` for additional information on shared object interfaces.

- `v12n_capabilities`
- `v12n_chassis_serialno`
- `v12n_ctrl_domain`
- `v12n_domain_name`
- `v12n_domain_roles`
- `v12n_domain_uuid`

**Files**  /usr/lib/libv12n.so.1  shared object

/usr/lib/64/libv12n.so.1  64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>system/library (32-bit), SUNWcalc (64-bit), SUNWhea</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Committed</td>
</tr>
<tr>
<td>MT-Level</td>
<td>MT-Safe</td>
</tr>
</tbody>
</table>

**See Also**  `virtinfo(1M), Intro(3), v12n(3EXT), attributes(5)`
libvolmgt – volume management library

**Synopsis**

```c
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.

```plaintext
media_findname  media_getattr
media_getid     media_setattr
volmgt_acquire  volmgt_check
volmgt_feature_enabled volmgt_inuse
volmgt_ownspath volmgt_release
volmgt_root     volmgt_running
volmgt_symdev   volmgt_symname
```

**Files**

```plaintext
/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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe with exceptions</td>
</tr>
</tbody>
</table>

**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  cc [ flag... ] file... [ library... ]

#include <wchar.h>

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 is implemented as a filter on \libc.so.1. New application development need not specify -lw.

Interfaces  The shared object \libw.so.1 provides the public interfaces defined below. See \Intro(3) for additional information on shared object interfaces.

fgetwc  fgetws  fputwc
fputws  getwc  getwchar
getws  isenglish  isideogram
isnumber  isphonogram  isspecial
ismalnum  iswalpha  iswcntrl
iswctype  iswdigit  iswgraph
iswlower  iswprint  iswpunct
iswspace  iswupper  iswdxigit
putwc  putwchar  putws
strtows  tolower  toupper
ungetwc  watoll  wcscat
wcchr  wcscmp  wcscoll
wcscpy  wcscspn  wcftime
wcslen  wscncat  wcscmp
wcscpy  wcspbrk  wcsrchr
wcspn  wcstod  wcstok
wcstol  wcstoul  wcswcs
wcwidth  wcsxfrm  wctype
wcwidth  wcasecmp  wcat
wschr  wscmp  wscol
### Files

<table>
<thead>
<tr>
<th>Files</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/lib/libw.so.1</td>
<td>a filter on libc.so.1</td>
</tr>
<tr>
<td>/lib/64/libw.so.1</td>
<td>a filter on 64/libc.so.1</td>
</tr>
</tbody>
</table>

### Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

### See Also

pvs(1), Intro(3), libc(3LIB), attributes(5)
**Name**  
libwsreg - product install registry library

**Synopsis**  
cc [ flag... ] file... -lwsreg [ library... ]  
#include <wsreg.h>

**Description**  
Functions in this library provide access to the product install registry.

**Interfaces**  
The shared object libwsreg.so.1 provides the public interfaces defined below. See `Intro(3)` for additional information on shared object interfaces.

```
wsreg_add_child_component  wsreg_add_compatible_version
wsreg_add_dependent_component  wsreg_add_display_name
wsreg_add_required_component  wsreg_can_access_registry
wsreg_clone_component  wsreg_components_equal
wsreg_create_component  wsreg_free_component
wsreg_free_component_array  wsreg_get
wsreg_get_all  wsreg_get_child_components
wsreg_get_compatible_versions  wsreg_get_data
wsreg_get_data_pairs  wsreg_get_dependent_components
wsreg_get_display_languages  wsreg_get_display_name
wsreg_get_id  wsreg_get_instance
wsreg_get_location  wsreg_get_parent
wsreg_get_required_components  wsreg_get_type
wsreg_get_uninstaller  wsreg_get_unique_name
wsreg_get_vendor  wsreg_get_version
wsreg_initialize  wsreg_query_create
wsreg_query_free  wsreg_query_get_id
wsreg_query_get_instance  wsreg_query_get_location
wsreg_query_get_unique_name  wsreg_query_get_version
wsreg_query_set_id  wsreg_query_set_instance
wsreg_query_set_location  wsreg_query_set_unique_name
wsreg_query_set_version  wsreg_register
wsreg_remove_child_component  wsreg_remove_compatible_version
```
wsreg_remove_dependent_component  wsreg_remove_display_name
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:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWwsr2</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

See Also
prodreg(1M), Intro(3), attributes(5)
libxnet(3LIB)

Name  libxnet – X/Open Networking library

Synopsis  cc [ flag... ] file... -lxnet [ library... ]

Description  Functions in this library provide networking interfaces which comply with the X/Open CAE Specification, Networking Services, Issue 4.

Interfaces  The shared object libxnet.so provides the public interfaces defined below. See Intro(3) for additional information on shared object interfaces.

```
__t_errno __xnet_bind
__xnet_connect __xnet_getsockopt
__xnet_listen __xnet_recvmmsg
__xnet_sendmsg __xnet_sendto
__xnet_socket __xnet_socketpair
_xti_accept __xti_alloc
_xti_bind __xti_close
_xti_connect __xti_error
_xti_free __xti_getinfo
_xti_getprotaddr __xti_getstate
_xti_getsockopt __xti_look
_xti_open __xti_optmgmt
_xti_recvmmsg __xti_rcvconnect
_xti_recvdis __xti_rcvrel
_xti_recvreldata __xti_rcvudata
_xti_rcvdata __xti_snd
_xti_snddis __xti_sndrel
_xti_sndreldata __xti_sndudata
_xti_sndv __xti_sndvudata
_xti_sndvudata __xti_sndvudata
_xti_sndvudata __xti_send
_xti_sndvudata __xti_sndvudata
_xti_sndvudata __xti_sync
_xti_sysconf __xti_unbind
_xti_xns5_accept __xti_xns5_snd
```

libxnet(3LIB)
accept  bind
connect  endhostent
endnetent  endprotoent
endservent  freeaddrinfo
gai_strerror  getaddrinfo
gethostbyaddr  gethostbyname
gethostent  gethostname
getnameinfo  getnetbyaddr
gethostname  getnetbyaddr
getnetbyname  getnetent
getpeername  getprotobynumber
getprotobynumber  getprotoent
getservbyname  getservbyport
getservent  getsockopt
htonl  htons
if_freenameindex  if_indextoname
if_nameindex  if_nametoindex
inet_addr  inet_lnaof
inet_makeaddr  inet_netof
inet_network  inet_ntoa
inet_ntop  inet_pton
listen  h_errno
htonl  htons
if_freenameindex  if_indextoname
if_nameindex  if_nametoindex
inet_addr  inet_lnaof
inet_makeaddr  inet_netof
inet_network  inet_ntoa
inet_ntop  inet_pton
listen  ntohl
ntohs  recv
recvfrom  recvmsg
send  sendmsg
sendto  sethostent
setnetent  setprotoent
setservent  setsockopt
shutdown sockatmark
socket socketpair
t_errno

**Files**

/library/libxnet.so.1 shared object
/library/64/libxnet.so.1 64-bit shared object

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcsl (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcslx (64-bit)</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Safe</td>
</tr>
</tbody>
</table>

**See Also**

Intro(3), attributes(5), standards(5)
Name libXtsol, libxtsol – Trusted Extensions to X Windows Library

Synopsis cc [flag...] file... -lx11 -lxtsol [library...]
#include <X11/extensions/Xtsol.h>

Description Functions in this library provide Trusted Extensions to the X windows library.

The functions in this library are available only if the system is configured with Trusted Extensions.

Interfaces The shared object /lib/libXtsol.so.1 provides the public interfaces that are defined below. See Intro(3) for additional information on shared object interfaces.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XTSOLIsWindowTrusted</td>
<td>XTSOLMakeTPWindow</td>
</tr>
<tr>
<td>XTSOLgetClientAttributes</td>
<td>XTSOLgetPropAttributes</td>
</tr>
<tr>
<td>XTSOLgetPropLabel</td>
<td>XTSOLgetPropUID</td>
</tr>
<tr>
<td>XTSOLgetResAttributes</td>
<td>XTSOLgetResLabel</td>
</tr>
<tr>
<td>XTSOLgetResUID</td>
<td>XTSOLgetSSHeight</td>
</tr>
<tr>
<td>XTSOLgetWorkstationOwner</td>
<td>XTSOLsetPolyInstInfo</td>
</tr>
<tr>
<td>XTSOLsetPropLabel</td>
<td>XTSOLsetPropUID</td>
</tr>
<tr>
<td>XTSOLsetResLabel</td>
<td>XTSOLsetResUID</td>
</tr>
<tr>
<td>XTSOLsetSSHeight</td>
<td>XTSOLsetSessionHI</td>
</tr>
<tr>
<td>XTSOLsetSessionLO</td>
<td>XTSOLsetWorkstationOwner</td>
</tr>
</tbody>
</table>

Files /lib/libXtsol.so.1 shared object
/lib/64/libXtsol.so.1 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWxwts</td>
</tr>
<tr>
<td>Interface Stability</td>
<td>Committed</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

See Also Intro(3), libtsnet(3LIB), libtsol(3LIB), attributes(5)

Notes The functionality described on this manual page is available only if the system has been configured with Trusted Extensions.
The function in this library provides a user interface to the `yacc(1)` library.

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

```
main yyerror
```

**Files**
- `/usr/lib/liby.so.1` shared object
- `/usr/lib/64/liby.so.1` 64-bit shared object

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>SUNWcs1, SUNWbtool (32-bit)</td>
</tr>
<tr>
<td></td>
<td>SUNWcs1x (64-bit)</td>
</tr>
<tr>
<td>MT-Level</td>
<td>Unsafe</td>
</tr>
</tbody>
</table>

**See Also**
`yacc(1)`, `Intro(3)`, `attributes(5)`
The `<limits.h>` header defines various symbolic names. Different categories of names are described below.

The names represent various limits on resources that the implementation imposes on applications. Symbolic constant names beginning with `_POSIX` can be found in `unistd.h`(3HEAD).

Applications should not assume any particular value for a limit. An application wishing to avail itself of the full amount of a resource available on an implementation can make use of the value given in limits.h on that particular implementation by using the symbolic names listed below. Many of the listed limits are not invariant, and at runtime, the value of the limit might differ from those given in this header, for the following reasons:

- The limit is pathname-dependent.
- The limit differs between the compile and runtime machines.

For these reasons, an application can use the `fpathconf`, `pathconf`, and `sysconf` functions to determine the actual value of a limit at runtime.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIO_LISTIO_MAX</td>
<td>Maximum number of I/O operations in a single list I/O call supported by the implementation.</td>
</tr>
<tr>
<td>AIO_MAX</td>
<td>Maximum number of outstanding asynchronous I/O operations supported by the implementation.</td>
</tr>
<tr>
<td>AIO_PRIO_DELTA_MAX</td>
<td>The maximum amount by which a process can decrease its asynchronous I/O priority level from its own scheduling priority.</td>
</tr>
<tr>
<td>ARG_MAX</td>
<td>Maximum length of argument to the <code>exec</code> functions including environment data.</td>
</tr>
<tr>
<td>ATEXIT_MAX</td>
<td>Maximum number of functions that can be registered with <code>atexit</code>.</td>
</tr>
<tr>
<td>CHILD_MAX</td>
<td>Maximum number of simultaneous processes per real user ID.</td>
</tr>
<tr>
<td>CLK_TCK</td>
<td>Number of clock ticks per second returned by the <code>times</code> function.</td>
</tr>
<tr>
<td>DELAYTIMER_MAX</td>
<td>Maximum number of timer expiration overruns.</td>
</tr>
<tr>
<td>HOST_NAME_MAX</td>
<td>Maximum length of a host name (not including the terminating null) as returned from the <code>gethostname</code> function.</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IOV_MAX</td>
<td>Maximum number of iovec structures that one process has available for use with <code>read(2)</code> or <code>write(2)</code>.</td>
</tr>
<tr>
<td>LOGIN_NAME_MAX</td>
<td>Maximum length of a login name.</td>
</tr>
<tr>
<td>MQ_OPEN_MAX</td>
<td>The maximum number of open message queue descriptors a process is allowed to hold.</td>
</tr>
<tr>
<td>LOGIN_NAME_MAX</td>
<td>Maximum length of a login name.</td>
</tr>
<tr>
<td>MQ_OPEN_MAX</td>
<td>The maximum number of open message queue descriptors a process is allowed to hold.</td>
</tr>
<tr>
<td>MQ_PRIO_MAX</td>
<td>The maximum number of message priorities supported by the implementation.</td>
</tr>
<tr>
<td>OPEN_MAX</td>
<td>Maximum number of files that one process can have open at any one time.</td>
</tr>
<tr>
<td>PAGESIZE</td>
<td>Size in bytes of a page.</td>
</tr>
<tr>
<td>PAGE_SIZE</td>
<td>Equivalent to PAGESIZE. If either PAGESIZE or PAGE_SIZE is defined, the other is defined with the same value.</td>
</tr>
<tr>
<td>PASS_MAX</td>
<td>The maximum number of significant bytes in a password, not including the terminating null.</td>
</tr>
<tr>
<td>PTHREAD_DECSTRUCTOR_ITERATIONS</td>
<td>Maximum number of attempts made to destroy a thread’s thread-specific data values on thread exit.</td>
</tr>
<tr>
<td>PTHREAD_KEYS_MAX</td>
<td>Maximum number of data keys that can be created by a process.</td>
</tr>
<tr>
<td>PTHREAD_STACK_MIN</td>
<td>Minimum size in bytes of thread stack storage.</td>
</tr>
<tr>
<td>PTHREAD_THREADS_MAX</td>
<td>Maximum number of threads that can be created per process.</td>
</tr>
<tr>
<td>RE_DUP_MAX</td>
<td>The number of repeated occurrences of a BRE permitted by the <code>regexec(3C)</code> and <code>regcomp(3C)</code> functions when using the interval notation <code>{(m,n]}</code>.</td>
</tr>
<tr>
<td>RTSIG_MAX</td>
<td>Maximum number of realtime signals reserved for application use in this implementation.</td>
</tr>
<tr>
<td>SEM_NSEMS_MAX</td>
<td>Maximum number of semaphores that a process can have.</td>
</tr>
<tr>
<td>SEM_VALUE_MAX</td>
<td>The maximum value a semaphore can have.</td>
</tr>
<tr>
<td>Pathname Variable</td>
<td>Values</td>
</tr>
<tr>
<td>-------------------</td>
<td>--------</td>
</tr>
<tr>
<td>SIGQUEUE_MAX</td>
<td>Maximum number of queued signals that a process can send and have pending at the receiver(s) at any time.</td>
</tr>
<tr>
<td>SS_REPL_MAX</td>
<td>The maximum number of replenishment operations that may be simultaneously pending for a particular sporadic server scheduler.</td>
</tr>
<tr>
<td>STREAM_MAX</td>
<td>The number of streams that one process can have open at one time. If defined, it has the same value as FOPEN_MAX.</td>
</tr>
<tr>
<td>SYMLOOP_MAX</td>
<td>Maximum number of symbolic links that can be reliably traversed in the resolution of a pathname in the absence of a loop.</td>
</tr>
<tr>
<td>TIMER_MAX</td>
<td>Maximum number of timers per process supported by the implementation.</td>
</tr>
<tr>
<td>TRACE_EVENT_NAME_MAX</td>
<td>Maximum length of the trace event name.</td>
</tr>
<tr>
<td>TRACE_NAME_MAX</td>
<td>Maximum length of the trace generation version string or of the trace stream name.</td>
</tr>
<tr>
<td>TRACE_SYS_MAX</td>
<td>Maximum number of trace streams that may simultaneously exist in the system.</td>
</tr>
<tr>
<td>TRACE_USER_EVENT_MAX</td>
<td>Maximum number of user trace event type identifiers that may simultaneously exist in a traced process, including the predefined user trace event POSIX_TRACE_UNNAMED_USER_EVENT.</td>
</tr>
<tr>
<td>TTY_NAME_MAX</td>
<td>Maximum length of terminal device name.</td>
</tr>
<tr>
<td>TZNAME_MAX</td>
<td>Maximum number of bytes supported for the name of a timezone (not of the TZ variable).</td>
</tr>
</tbody>
</table>

The values in the following list can be constants within an implementation or can vary from one pathname to another. For example, file systems or directories can have different characteristics. The value supported for a specific pathname is provided by the pathconf(2) function.

<table>
<thead>
<tr>
<th>Pathname Variable</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>FILESIZEBITS</td>
<td>Minimum number of bits needed to represent, as a signed integer value, the maximum size of a regular file allowed in the specified directory.</td>
</tr>
<tr>
<td>LINK_MAX</td>
<td>Maximum number of links to a single file.</td>
</tr>
<tr>
<td>MAX_CANON</td>
<td>Maximum number of bytes in a terminal canonical input line.</td>
</tr>
</tbody>
</table>
MAX_INPUT: Minimum number of bytes for which space is available in a terminal input queue; therefore, the maximum number of bytes a conforming application may require to be typed as input before reading them.

NAME_MAX: Maximum number of bytes in a filename (not including terminating null).

PATH_MAX: Maximum number of bytes in a pathname, including the terminating null character.

PIPE_BUF: Maximum number of bytes that is guaranteed to be atomic when writing to a pipe.

POSIX_ALLOC_SIZE_MIN: Minimum number of bytes of storage actually allocated for any portion of a file.

POSIX_REC_INCR_XFER_SIZE: Recommended increment for file transfer sizes between the POSIX_REC_MIN_XFER_SIZE and POSIX_REC_MAX_XFER_SIZE values.

POSIX_REC_MAX_XFER_SIZE: Maximum recommended file transfer size.

POSIX_REC_MIN_XFER_SIZE: Minimum recommended file transfer size.

POSIX_REC_XFER_ALIGN: Recommended file transfer buffer alignment.

SYMLINK_MAX: Maximum number of bytes in a symbolic link.

The magnitude limitations in the following list are fixed by specific implementations. An application should assume that the value supplied by \texttt{limits.h} in a specific implementation is the minimum that pertains whenever the application is run under that implementation. A specific instance of a specific implementation can increase the value relative to that supplied by \texttt{limits.h} for that implementation. The actual value supported by a specific instance is provided by the \texttt{sysconf(3C)} function.

BC_BASE_MAX: Maximum obase values allowed by the \texttt{bc(1)} utility.

BC_DIM_MAX: Maximum number of elements permitted in an array by the \texttt{bc} utility.

BC_SCALE_MAX: Maximum scale value allowed by the \texttt{bc} utility.

BC_STRING_MAX: Maximum length of a string constant accepted by the \texttt{bc} utility.

CHARCLASS_NAME_MAX: Maximum number of bytes in a character class name.

COLL_WEIGHTS_MAX: Maximum number of weights that can be assigned to an entry of the \texttt{LC_COLLATE} order keyword in the locale definition file.

EXPR_NEST_MAX: Maximum number of expressions that can be nested within parentheses by the \texttt{expr(1)} utility.
EXPR_NEST_MAX

Maximum number of expressions that can be nested within parentheses by the expr utility.

LINE_MAX

Unless otherwise noted, the maximum length, in bytes, of a utility’s input line (either standard input or another file), when the utility is described as processing text files. The length includes room for the trailing <newline>.

NGROUPS_MAX

Maximum number of simultaneous supplementary group IDs per process.

RE_DUP_MAX

Maximum number of repeated occurrences of a regular expression permitted when using the interval notation \{m,n\}.

The symbolic constants in the following list are symbolic names for the most restrictive value for certain features on an implementation supporting the POSIX Timers option.

_POSIX_CLOCKRES_MIN

The resolution of the CLOCK_REALTIME clock, in nanoseconds.

The symbolic constants in the following list are symbolic names for the most restrictive value for certain features on an implementation conforming to various POSIX and Single Unix Specification requirements. See standards(5).

_POSIX_AIO_LISTIO_MAX

The number of I/O operations that can be specified in a list I/O call.

_POSIX_AIO_MAX

The number of outstanding asynchronous I/O operations.

_POSIX_ARG_MAX

Maximum length of argument to the exec(2) functions including environment data.

_POSIX_CHILD_MAX

Maximum number of simultaneous processes per real user ID.

_POSIX_DELAYTIMER_MAX

The number of timer expiration overruns.

_POSIX_HOST_NAME_MAX

Maximum length of a host name (not including the terminating null) as returned from the gethostname(3C) function.

_POSIX_LINK_MAX

Maximum number of links to a single file.

_POSIXLOGIN_NAME_MAX

The size of the storage required for a login name, in bytes, including the terminating null.

_POSIX_MAX_CANON

Maximum number of bytes in a terminal canonical input queue.

_POSIX_MAX_INPUT

Maximum number of bytes allowed in a terminal input queue.
<table>
<thead>
<tr>
<th><strong>_POSIX_MQ_OPEN_MAX</strong></th>
<th>The number of message queues that can be open for a single process.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>_POSIX_MQ_PRIO_MAX</strong></td>
<td>The maximum number of message priorities supported by the implementation.</td>
</tr>
<tr>
<td><strong>_POSIX_NAME_MAX</strong></td>
<td>Maximum number of bytes in a filename (not including terminating null).</td>
</tr>
<tr>
<td><strong>_POSIX_NGROUPS_MAX</strong></td>
<td>Maximum number of simultaneous supplementary group IDs per process.</td>
</tr>
<tr>
<td><strong>_POSIX_OPEN_MAX</strong></td>
<td>Maximum number of files that one process can have open at any one time.</td>
</tr>
<tr>
<td><strong>_POSIX_PATH_MAX</strong></td>
<td>Maximum number of bytes in a pathname.</td>
</tr>
<tr>
<td><strong>_POSIX_PIPE_BUF</strong></td>
<td>Maximum number of bytes that is guaranteed to be atomic when writing to a pipe.</td>
</tr>
<tr>
<td><strong>_POSIX_RE_DUP_MAX</strong></td>
<td>The number of repeated occurrences of a BRE permitted by the <code>regexec()</code> and <code>regcomp()</code> functions when using the interval notation <code>{\(m,n\)}</code></td>
</tr>
<tr>
<td><strong>_POSIX_RTSIG_MAX</strong></td>
<td>The number of realtime signal numbers reserved for application use.</td>
</tr>
<tr>
<td><strong>_POSIX_SEM_NSEMS_MAX</strong></td>
<td>The number of semaphores that a process can have.</td>
</tr>
<tr>
<td><strong>_POSIX_SEM_VALUE_MAX</strong></td>
<td>The maximum value a semaphore can have.</td>
</tr>
<tr>
<td><strong>_POSIX_SIGQUEUE_MAX</strong></td>
<td>The number of queued signals that a process can send and have pending at the receiver(s) at any time.</td>
</tr>
<tr>
<td><strong>_POSIX_SSIZE_MAX</strong></td>
<td>The value that can be stored in an object of type <code>ssize_t</code>.</td>
</tr>
<tr>
<td><strong>_POSIX_STREAM_MAX</strong></td>
<td>The number of streams that one process can have open at one time.</td>
</tr>
<tr>
<td><strong>_POSIX_SS_REPL_MAX</strong></td>
<td>The number of replenishment operations that can be simultaneously pending for a particular sporadic server scheduler.</td>
</tr>
<tr>
<td><strong>_POSIX_SYMLINK_MAX</strong></td>
<td>The number of bytes in a symbolic link.</td>
</tr>
<tr>
<td>Constant</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>_POSIX_SYMLOOP_MAX</td>
<td>The number of symbolic links that can be traversed in the resolution of a pathname in the absence of a loop.</td>
</tr>
<tr>
<td>_POSIX_THREAD_DESTRUCTOR_ITERATIONS</td>
<td>The number of attempts made to destroy a thread's thread-specific data values on thread exit.</td>
</tr>
<tr>
<td>_POSIX_THREAD_KEYS_MAX</td>
<td>The number of data keys per process.</td>
</tr>
<tr>
<td>_POSIX_THREAD_THREADS_MAX</td>
<td>The number of threads per process.</td>
</tr>
<tr>
<td>_POSIX_TIMER_MAX</td>
<td>The per-process number of timers.</td>
</tr>
<tr>
<td>_POSIX_TRACE_EVENT_NAME_MAX</td>
<td>The length in bytes of a trace event name.</td>
</tr>
<tr>
<td>_POSIX_TRACE_NAME_MAX</td>
<td>The length in bytes of a trace generation version string or a trace stream name.</td>
</tr>
<tr>
<td>_POSIX_TRACE_SYS_MAX</td>
<td>The number of trace streams that can simultaneously exist in the system.</td>
</tr>
<tr>
<td>_POSIX_TRACE_USER_EVENT_MAX</td>
<td>The number of user trace event type identifiers that may simultaneously exist in a traced process, including the predefined user trace event POSIX_TRACE_UNNAMED_USER_EVENT.</td>
</tr>
<tr>
<td>_POSIX_TTY_NAME_MAX</td>
<td>The size of the storage required for a terminal device name, in bytes, including the terminating null.</td>
</tr>
<tr>
<td>_POSIX_TZNAME_MAX</td>
<td>Maximum number of bytes supported for the name of a timezone (not of the TZ variable).</td>
</tr>
<tr>
<td>_POSIX2_BC_BASE_MAX</td>
<td>Maximum obase values allowed by the bc utility.</td>
</tr>
<tr>
<td>_POSIX2_BC_DIM_MAX</td>
<td>Maximum number of elements permitted in an array by the bc utility.</td>
</tr>
<tr>
<td>_POSIX2_BC_SCALE_MAX</td>
<td>Maximum scale value allowed by the bc utility.</td>
</tr>
<tr>
<td>_POSIX2_BC_STRING_MAX</td>
<td>Maximum length of a string constant accepted by the bc utility.</td>
</tr>
<tr>
<td>_POSIX2_CHARCLASS_NAME_MAX</td>
<td>Maximum number of bytes in a character class name.</td>
</tr>
<tr>
<td>_POSIX2_COLL_WEIGHTS_MAX</td>
<td>Maximum number of weights that can be assigned to an entry of the LC_COLLATE order keyword in the locale definition file.</td>
</tr>
</tbody>
</table>
Maximum number of expressions that can be nested within parentheses by the `expr` utility.

Unless otherwise noted, the maximum length, in bytes, of a utility's input line (either standard input or another file), when the utility is described as processing text files. The length includes room for the trailing `<newline>`.

Maximum number of repeated occurrences of a regular expression permitted when using the interval notation `{m,n}`.

Maximum number of `iovec` structures that one process has available for use with `read(2)` or `write(2)`.

Maximum number of bytes in a filename (not including the terminating null).

Maximum number of bytes in a pathname.

The values in the following lists shall be defined in `<limits.h>` and are constant expressions suitable for use in `#if` preprocessing directives. Moreover, except for `CHAR_BIT`, `DBL_DIG`, `DBL_MAX`, `FLT_DIG`, `FLT_MAX`, `LONG_BIT`, `WORD_BIT`, and `MB_LEN_MAX`, the symbolic names are defined as expressions of the correct type.

If the value of an object of type `char` is treated as a signed integer when used in an expression, the value of `CHAR_MIN` is the same as that of `SCHAR_MIN` and the value of `CHAR_MAX` is the same as that of `SCHAR_MAX`. Otherwise, the value of `CHAR_MIN` is 0 and the value of `CHAR_MAX` is the same as that of ` UCHAR_MAX`.

- **CHAR_BIT** Number of bits in a type `char`.
- **CHAR_MAX** Maximum value of type `char`.
- **CHAR_MIN** Minimum value of type `char`.
- **DBL_DIG** Digits of precision of type `double`.
- **DBL_MAX** Maximum decimal value of a `double`.
- **DBL_MIN** Minimum decimal value of a `double`.
- **FLT_DIG** Digits of precision of type `float`.
- **FLT_MAX** Maximum decimal value of a `float`.
- **FLT_MIN** Minimum decimal value of a `float`.
The following constants are defined in `<limits.h>`.

**INT_MIN**  Minimum value of type `int`.
**INT_MAX**  Maximum value of an `int`.
**LLONG_MIN** Minimum value of type `long long`.
**LLONG_MAX** Maximum value of type `long long`.
**LONG_BIT**  Number of bits in a `long`.
**LONG_MIN**  Minimum value of type `long`.
**LONG_MAX**  Maximum value of a `long`.
**MB_LEN_MAX** Maximum number of bytes in a character, for any supported locale.
**SCHAR_MIN** Minimum value of type `signed char`.
**SCHAR_MAX** Maximum value of type `signed char`.
**SHRT_MIN**  Minimum value of type `short`.
**SHRT_MAX**  Maximum value of type `short`.
**SSIZE_MAX**  Maximum value of an object of type `ssize_t`.
**TMP_MAX** Minimum number of unique filename generated by `tmpnam(3C)`. Maximum number of times an application can call `tmpnam()` reliably.
**UCHAR_MAX** Maximum value of type `unsigned char`.
**UINT_MAX**  Maximum value of type `unsigned`.
**ULLONG_MAX** Maximum value of type `unsigned long long`.
**ULONG_MAX**  Maximum value of type `unsigned long`.
**USHRT_MAX**  Maximum value for a type `unsigned short`.
**WORD_BIT**  Number of bits in a word or type `int`.

Other Invariant Values

**CHARCLASS_NAME_MAX**  Maximum number of bytes in a character class name.
**LOGNAME_MAX**  The maximum number of bytes supported in a user’s login name.
**NL_ARGMAX**  Maximum value of digit in calls to the printf(3C) and scanf(3C) functions.
**NL_LANGMAX**  Maximum number of bytes in a LANG name.
**NL_MSGMAX**  Maximum message number.
**NL_NMAX**  Maximum number of bytes in an N-to-1 collation mapping.
**NL_SETMAX**  Maximum set number.
NL_TEXTMAX  Maximum number of bytes in a message string.
NZERO       Default process priority.

See Also   fpathconf(2), pathconf(2), sysconf(3C), standards(5)
locale.h includes a definition for the `lconv` structure, which includes the following members. (See the definition of LC_MONETARY in `locale(5)`.)

- char *currency_symbol
- char *decimal_point
- char *frac_digits
- char *grouping
- char *int_curr_symbol
- char *int_fрак digits
- char *int_n_cs_precedes
- char *int_n_seپ_by_space
- char *int_n_sign_posn
- char *int_p_cs_precedes
- char *int_p_sep_by_space
- char *int_p_sign_posn
- char *mon_decimal_point
- char *mon_grouping
- char *monThousandsSep
- char *negative_sign
- char *n_cs_precedes
- char *nSep_by_space
- char *n_sign_posn
- char *positive_sign
- char *p_cs_precedes
- char *pSep_by_space
- char *p_sign_posn
- char *thousandsSep

The `locale.h` header defines NULL (as defined in `<stddef.h>`) and the following as macros:

- LC_ALL
- LC_COLLATE
- LC_CTYPE
- LC_MESSAGES
- LC_NUMERIC
- LC_MONETARY
- LC_TIME

The preceding expand to distinct integer constant expressions, for use as the first argument to the `setlocale()` function. See `setlocale(3C)`.

Additional macro definitions, beginning with the characters LC_ and an uppercase letter, can also be specified here.
Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTETYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also setlocale(3C), localeconv(3C), stddef.h(3HEAD), attributes(5), locale(5), standards(5)
**Name**
math.h, math – mathematical declarations

**Synopsis**
#include <math.h>

**Description**
The `<math.h>` header includes definitions for the following types:

- `float_t` A real-floating type at least as wide as `float`.
- `double_t` A real-floating type at least as wide as `double`, and at least as wide as `float_t`.

If `FLT_EVAL_METHOD` equals 0, `float_t` and `double_t` are `float` and `double`, respectively. If `FLT_EVAL_METHOD` equals 1, they are both `double`. If `FLT_EVAL_METHOD` equals 2, they are both `long double`. Other values of `FLT_EVAL_METHOD` are implementation-defined.

The `<math.h>` header provides the following constants. The values are of type `double` and are accurate within the precision of the `double` type.

- `M_E` The base of natural logarithms (e).
- `M_LOG2E` The base-2 logarithm of e.
- `M_LOG10E` The base-10 logarithm of e.
- `M_LN2` The natural logarithm of 2.
- `M_LN10` The natural logarithm of 10.
- `M_PI` π, the ratio of the circumference of a circle to its diameter.
- `M_PI_2` π/2.
- `M_PI_4` π/4.
- `M_1_PI` 1/π.
- `M_2_PI` 2/π.
- `M_2_SQRTPI` 2 over the square root of π.
- `M_SQRT2` The positive square root of 2.
- `M_SQRT1_2` The positive square root of 1/2.

The `<math.h>` header defines the following symbolic constants:

- `MAXFLOAT` The maximum value of a non-infinite single-precision floating point number.
- `HUGE_VAL` A positive `double` expression, not necessarily representable as a `float`. Used as an error value returned by the mathematics library. `HUGE_VAL` evaluates to +infinity on systems supporting IEEE Std 754-1985.
- `HUGE_VALF` A positive `float` constant expression. Used as an error value returned by the mathematics library. `HUGE_VALF` evaluates to +infinity on systems supporting IEEE Std 754-1985.
HUGE_VALL  A positive long double constant expression. Used as an error value returned by the mathematics library. HUGE_VALL evaluates to +infinity on systems supporting IEEE Std 754-1985.

INFINITY  A constant expression of type float representing positive or unsigned infinity, if available; else a positive constant of type float that overflows at translation time.

NAN  A constant expression of type float representing a quiet NaN. This symbolic constant is only defined if the implementation supports quiet NaNs for the float type.

The following macros are defined for number classification. They represent the mutually-exclusive kinds of floating-point values. They expand to integer constant expressions with distinct values

FP_INFINITE
FP_NAN
FP_NORMAL
FP_SUBNORMAL
FP_ZERO

The following optional macros indicate whether the fma() family of functions are fast compared with direct code:

FP_FAST_FMA
FP_FAST_FMAF
FP_FAST_FMAL

The FP_FAST_FMA macro is defined to indicate that the fma() function generally executes about as fast as, or faster than, a multiply and an add of double operands. The other macros have the equivalent meaning for the float and long double versions.

The following macros expand to integer constant expressions whose values are returned by ilogb(x) if x is zero or NaN, respectively. The value of FP_ILOGB0 is either {INT_MIN} or -{INT_MAX}. The value of FP_ILOGBNAN is either {INT_MAX} or {INT_MIN}.

FP_ILOGB0
FP_ILOGBNAN

The following macros expand to the integer constants 1 and 2, respectively:

MATH_ERRNO
MATH_ERREXCEPT

The following macro expands to an expression that has type int and the value MATH_ERREXCEPT:

math_errnohandling
The value of the macro `math_errhandling` is constant for the duration of the program. If a macro definition is suppressed or a program defines an identifier with the name `math_errhandling`, the behavior is undefined.

The `<math.h>` header defines the following external variable:

```c
extern int signgam;
```

The `<math.h>` header defines the structure and constants used by the `matherr(3M)` error-handling mechanisms.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also** `Intro(3), fenv.h(3HEAD), libm(3LIB), limits.h(3HEAD), matherr(3M), attributes(5), standards(5)`
mman.h(3HEAD)

Name    mman.h, mman – memory management declarations
Synopsis #include <sys/mman.h>
Description The <sys/mman.h> header supports the following options:

- the Memory Mapped Files option
- the Shared Memory Objects option
- the Process Memory Locking option
- the Memory Protection option
- the Synchronized Input and Output option

For Memory Mapped Files and Shared Memory Objects options, the following protection options are defined:

- PROT_READ Page can be read.
- PROT_WRITE Page can be written.
- PROT_EXEC Page can be executed.
- PROT_NONE Page cannot be accessed.

The following flag options are defined:

- MAP_SHARED Share changes.
- MAP_PRIVATE Changes are private.
- MAP_FIXED Interpret addr exactly.

The flags immediately following are defined for msync(). See msync(3).

- MS_ASYNC Perform asynchronous writes.
- MS_SYNC Perform synchronous writes.
- MS_INVALIDATE Invalidate mappings.

The symbolic constants immediately following are defined for the mlockall() function. See mlockall(3).

- MCL_CURRENT Lock currently mapped pages.
- MCL_FUTURE Lock pages that become mapped.

The symbolic constant MAP_FAILED is defined to indicate a failure from the mmap() function. See mmap(2).

The mode_t, off_t, and size_t types are be defined as described in <sys/types.h>. See types(3HEAD).
Attributes  See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also  mmap(2), mprotect(2), munmap(2), madvise(3C), mlock(3C), mlockall(3C), msync(3C), shm_open(3RT), shm_unlink(3RT), attributes(5), standards(5)
#include <monetary.h>

The `<monetary.h>` header defines the following types:

- `size_t` As described in `stddef.h`.
- `ssize_t` As described in `types.h`.

## Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

## See Also

`stddef.h(3HEAD), strfmon(3C), types.h(3HEAD), attributes(5), standards(5)`
Name    mqueue.h, mqueue – message queues

Synopsis    #include <mqueue.h>

Description    The <mqueue.h> header defines the mqd_t 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 OPEN_MAX file and message queues.

The <mqueue.h> header defines the sigevent structure (as described in <signal.h>, see signal.h(3HEAD)) and the mq_attr 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 mq_attr structure has the following members:

long mq_flags      message queue flags
long mq_maxmsg     maximum number of messages
long mq_msgsize    maximum message size
long mq_curmsgs    number of messages currently queued

Inclusion of the <mqueue.h> header may make visible symbols defined in the headers <fcntl.h>, <signal.h>, <sys/types.h>, and <time.h>.

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

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Committed</td>
</tr>
<tr>
<td>Standard</td>
<td>See standards(5).</td>
</tr>
</tbody>
</table>

See Also    fcntl.h(3HEAD), signal.h(3HEAD), time.h(3HEAD), types.h(3HEAD), attributes(5), standards(5)
msg.h (3HEAD)

**Name**
msg.h, msg – message queue structures

**Synopsis**
```c
#include <sys/msg.h>
```

**Description**
The `<sys/msg.h>` header defines the following data types through `typedef`:
- `msgqnum_t` used for the number of messages in the message queue
- `msglen_t` used for the number of bytes allowed in the message queue

These types are unsigned integer types that are able to store values at least as large as a type unsigned short.

The `<sys/msg.h>` header defines the following constant as a message operation flag:
- `MSG_NOERROR` no error if big message

The `msgid_ds` structure contains the following members:
- `struct ipc_perm msg_perm` Operation permission structure.
- `msgqnum_t msg_qnum` Number of messages currently on queue.
- `msglen_t msg_qbytes` Maximum number of bytes allowed on queue.
- `pid_t msg_lspid` Process ID of last msgsnd(2).
- `pid_t msg_lrpid` Process ID of last msgrcv(2).
- `time_t msg_stime` Time of last msgsnd().
- `time_t msg_rtime` Time of last msgrcv().
- `time_t msg_ctime` Time of last change.

The `pid_t`, `time_t`, `key_t`, `size_t`, and `ssize_t` types are defined as described in `<sys/types.h>`. See `types(3HEAD)`.

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

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**
`msgctl(2), msgget(2), msgrcv(2), msgsnd(2), ipc.h(3HEAD), types.h(3HEAD), attributes(5), standards(5)`
**Synopsis**  
```
#include <ndbm.h>
```

**Description**  
The `<ndbm.h>` header defines the `datum` type as a structure that includes at least the following members:

```c
void *dptr /* pointer to the application's data */
size_t dsize /* size of the object pointed to by dptr */
```

The `size_t` type is defined through `typedef` as described in `<stddef.h>`.

The `<ndbm.h>` header defines the `DBM` type through `typedef`.

The following constants are defined as possible values for the `store_mode` argument to `dbm_store()`:

- `DBM_INSERT`  Insertion of new entries only.
- `DBM_REPLACE` Allow replacing existing entries.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**  
`ndbm(3C), attributes(5), standards(5)`
#include <netdb.h>

The `<netdb.h>` header defines the type `in_port_t` and the type `in_addr_t` as described in `in.h(3HEAD)`.

The `<netdb.h>` header defines the `hostent` structure that includes the following members:

- `char *h_name /* official name of the host */`
- `char **h_aliases /* pointer to an array of pointers to alternative host names, terminated by a null pointer */`
- `int h_addrtype /* address type */`
- `int h_length /* length, in bytes, of the address */`
- `char **h_addr_list /* pointer to an array of pointers to network addresses (in network byte order) for the host, terminated by a null pointer */`

The `<netdb.h>` header defines the `netent` structure that includes the following members:

- `char *n_name /* official, fully-qualified (including the domain) name of the network */`
- `char **n_aliases /* pointer to an array of pointers to alternative network names, terminated by a null pointer */`
- `int n_addrtype /* the address type of the network */`
- `in_addr_t n_net /* the network number, in host byte order */`

The `<netdb.h>` header defines the `protoent` structure that includes the following members:

- `char *p_name /* official name of the protocol */`
- `char **p_aliases /* pointer to an array of pointers to alternative protocol names, terminated by a null pointer */`
- `int p_proto /* protocol number */`

The `<netdb.h>` header defines the `servent` structure that includes the following members:

- `char *s_name /* official name of the service */`
- `char **s_aliases /* pointer to an array of pointers to alternative service names, terminated by a null pointer */`
- `int s_port /* port number at which the service resides, in network byte order */`
- `char *s_proto /* 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`:

```c
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.h(3HEAD)`.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**  Intro(3), endhostent(3NSL), endhostent(3XNET), endnetent(3SOCKET), endnetent(3XNET), endprotoent(3SOCKET), endprotoent(3XNET), endservent(3SOCKET), endservent(3XNET), in.h(3HEAD), attributes(5), standards(5)
nl_types.h(3HEAD)

**Name**  nl_types.h, nl_types – native language data types

**Synopsis**  #include <nl_types.h>

**Description**  This header contains the following definitions:

- **nl_catd**: Used by the message catalog functions catopen, catgets and catclose to identify a catalog.
- **nl_item**: Used by nl_langinfo to identify items of langinfo data. Values for objects of type nl_item are defined in <langinfo.h>.
- **NL_SETD**: Used by gencat when no $set directive is specified in a message text source file. This constant can be used in subsequent calls to catgets as the value of the set identifier parameter.
- **NL_MGSMAX**: Maximum number of messages per set.
- **NL_SETMAX**: Maximum number of sets per catalog.
- **NL_TEXTMAX**: Maximum size of a message.

**See Also**  gencat(1), catgets(3C), catopen(3C), nl_langinfo(3C), langinfo.h(3HEAD)
poll.h(3HEAD)

Name poll.h, poll – definitions for the poll() function

Synopsis #include <poll.h>

Description The <poll.h> header defines the pollfd structure, which includes the following members:

- `int fd` — the following descriptor being polled
- `short events` — the input event flags (see below)
- `short revents` — the output event flags (see below)

The <poll.h> header defines the following type through typedef:

- `nfds_t` — an unsigned integer type used for the number of file descriptors

The implementation supports one or more programming environments in which the width of `nfds_t` is no greater than the width of type long. The names of these programming environments can be obtained using the `confstr()` function or the `getconf` utility. See `confstr(3C)` and `getconf(1)`.

The following symbolic constants are defined, zero or more of which can be OR’ed together to form the `events` or `revents` members in the pollfd structure:

- `POLLIN` — Data other than high-priority data can be read without blocking.
- `POLLRDNORM` — Normal data can be read without blocking.
- `POLLRDBAND` — Priority data can be read without blocking.
- `POLLPRI` — High priority data can be read without blocking.
- `POLLOUT` — Normal data can be written without blocking.
- `POLLWRNORM` — Equivalent to POLLOUT.
- `POLLWRBAND` — Priority data can be written.
- `POLLERR` — An error has occurred (revents only).
- `POLLYUP` — Device has been disconnected (revents only).
- `POLLNVAL` — Invalid fd member (revents only).

The significance and semantics of normal, priority, and high-priority data are file and device-specific.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>
See Also  getconf(1), poll(2), confstr(3C), attributes(5), standards(5)
**Name**  
pthread.h, pthread – threads

**Synopsis**  
#include <pthread.h>

**Description**  
The `<pthread.h>` header defines the following symbols:

- PTHREAD_BARRIER_SERIAL_THREAD
- PTHREAD_CANCEL_ASYNCHRONOUS
- PTHREAD_CANCEL_ENABLE
- PTHREAD_CANCEL_DEFERRED
- PTHREAD_CANCEL_DISABLE
- PTHREAD_CANCELED
- PTHREAD_COND_INITIALIZER
- PTHREAD_CREATE_DETACHED
- PTHREAD_CREATE_JOINABLE
- PTHREAD_EXPLICIT_SCHED
- PTHREAD_INHERIT_SCHED
- PTHREAD_MUTEX_DEFAULT
- PTHREAD_MUTEX_ERRORCHECK
- PTHREAD_MUTEX_INITIALIZER
- PTHREAD_MUTEX_NORMAL
- PTHREAD_MUTEX_RECURSIVE
- PTHREAD_ONCE_INIT
- PTHREAD_PRIO_INHERIT
- PTHREAD_PRIO_NONE
- PTHREAD_PRIO_PROTECT
- PTHREAD_PROCESS_SHARED
- PTHREAD_PROCESS_PRIVATE
- PTHREAD_SCOPE_PROCESS
- PTHREAD_SCOPE_SYSTEM

The types listed below are defined as described in `<sys/types.h>`. See `types.h(3HEAD)`.

- pthread_attr_t
- pthread_barrier_t
- pthread_barrierattr_t
- pthread_cond_t
- pthread_condattr_t
- pthread_key_t
- pthread_mutex_t
- pthread_mutexattr_t
- pthread_once_t
- pthread_rwlock_t
- pthread_rwlockattr_t
- pthread_spinlock_t
- pthread_t
Attributes  See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also  sched.h(3HEAD), time.h(3HEAD), types.h(3HEAD), pthread_attr_getguardsize(3C),
pthread_attr_init(3C), pthread_attr_setscope(3C), pthread_cancel(3C),
pthread_cleanup_pop(3C), pthread_cond_init(3C), pthread_cond_signal(3C),
pthread_cond_wait(3C), pthread_condattr_init(3C), pthread_create(3C),
pthread_detach(3C), pthread_equal(3C), pthread_exit(3C),
pthread_getconcurrency(3C), pthread_getschedparam(3C), pthread_join(3C),
pthread_key_create(3C), pthread_key_delete(3C), pthread_mutex_init(3C),
pthread_mutex_lock(3C), pthread_mutexattr_getpshared(3C),
pthread_mutexattr_gettype(3C), pthread_mutexattr_getprotocol(3C),
pthread_mutexattr_init(3C), pthread_once(3C), pthread_rwlock_init(3C),
pthread_rwlock_rdlock(3C), pthread_rwlock_unlock(3C), pthread_rwlock_wrlock(3C),
pthread_rwlockattr_getpshared(3C), pthread_rwlockattr_init(3C),
pthread_self(3C), pthread_setcancelstate(3C), pthread_setspecific(3C),
attributes(5), standards(5)
Name (pwd.h, pwd – password structure)

Synopsis  #include <pwd.h>

Description The <pwd.h> header provides a definition for struct passwd, which includes the following members:

char *pw_name user’s login name
uid_t pw_uid numerical user ID
gid_t pw_gid numerical group ID
char *pw_dir initial working directory
char *pw_shell program to use as shell

The gid_t and uid_t types are defined as described in <sys/types.h>. See types.h(3HEAD).

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also  getpwnam(3C), types.h(3HEAD), attributes(5), standards(5)
#include <regex.h>

Description

The `<regex.h>` header defines the structures and symbolic constants used by the `regcomp()`, `regexec()`, `regerror()`, and `regfree()` functions. See `regcomp(3C)`.

The structure type `regex_t` contains the following member:

```c
size_t re_nsub
```

number of parenthesized subexpressions

The type `size_t` is defined as described in `<sys/types.h>`. See `types.h(3HEAD)`.

The type `regoff_t` is defined as a signed integer type that can hold the largest value that can be stored in either a type `off_t` or type `ssize_t`. The structure type `regmatch_t` contains the following members:

```c
regoff_t rm_so
```

byte offset from start of string to start of substring

```c
regoff_t rm_eo
```

byte offset from start of string of the first character after the end of substring

Values for the `cflags` parameter to the `regcomp()` function are as follows:

- `REG_EXTENDED`: use extended regular expressions
- `REG_ICASE`: ignore case in match
- `REG_NOSUB`: report only success or fail in `regexec()`
- `REG_NEWLINE`: change the handling of `NEWLINE` character

Values for the `eflags` parameter to the `regexec()` function are as follows:

- `REG_NOTBOL`: The circumflex character (`^`), when taken as a special character, does not match the beginning of string.
- `REG_NOTEOL`: The dollar sign (`$`), when taken as a special character, does not match the end of string.

The following constants are defined as error return values:

- `REG_NOMATCH`: `regexec()` failed to match.
- `REG_BADPAT`: Invalid regular expression.
- `REG_ECOLLATE`: Invalid collating element referenced.
- `REG_ECTYPE`: Invalid character class type referenced.
- `REG_EESCAPE`: Trailing `\` in pattern.
- `REG_ESUBREG`: Number in `\digit` invalid or in error.
- `REG_EBRACK`: `"[" imbalance.
REG_EPAREN “\(\)” or “\)” imbalance.
REG_EBRACE “\{|\}” imbalance.
REG_BADBR Content of “\{|\}” invalid: not a number, number too large, more than two numbers, first larger than second.
REG_ERANGE Invalid endpoint in range expression.
REG_ESPACE Out of memory.
REG_BADRPT ‘?’, ‘*’, or ‘+’ not preceded by valid regular expression.
REG_ENOSYS Reserved.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also  regcomp(3C), types.h(3HEAD), attributes(5), standards(5)
#include <sys/resource.h>

The `<sys/resource.h>` header defines the symbolic constants listed below as possible values of the `which` argument of `getpriority()` and `setpriority()`. See `getpriority(3C)`.

- **PRIO_PROCESS**: identifies the `who` argument as a process ID
- **PRIO_PGRP**: identifies the `who` argument as a process group ID
- **PRIO_USER**: identifies the `who` argument as a user ID

The following type is defined through `typedef`:

```c
typedef unsigned int rlim_t;
```

The following symbolic constants are defined:

- **RLIM_INFINITY**: a value of `rlim_t` indicating no limit
- **RLIM_SAVED_MAX**: a value of type `rlim_t` indicating an unrepresentable saved hard limit
- **RLIM_SAVED_CUR**: a value of type `rlim_t` indicating an unrepresentable saved soft limit

The symbolic constants listed below are defined as possible values of the `who` parameter of `getrusage()`. See `getrusage(3C)`.

- **RUSAGE_SELF**: returns information about the current process
- **RUSAGE_CHILDREN**: returns information about children of the current process

The `<sys/resource.h>` header defines the `rlimit` structure, which includes the following members:

```c
typedef struct {
    rlim_t rlim_cur; /* the current (soft) limit */
    rlim_t rlim_max; /* the hard limit */
} rlimit_t;
```

The `<sys/resource.h>` header defines the `rusage` structure, which includes the following members:

```c
typedef struct {
    struct timeval ru_utime; /* user time used */
    struct timeval ru_stime; /* system time used */
} rusage_t;
```

The `timeval` structure is defined as described in `<sys/time.h>`.

The symbolic constants listed below are defined as possible values for the `resource` argument of `getrlimit()` and `setrlimit()`. See `getrlimit(2)`.

- **RLIMIT_CORE**: limit on size of core dump file
- **RLIMIT_CPU**: limit on CPU time per process
- **RLIMIT_DATA**: limit on data segment size
RLIMITFSIZE       limit on file size
RLIMIT_Nofile     limit on number of open files
RLIMIT_STACK      limit on stack size
RLIMIT_AS         limit on address space size

The id_t type is defined through typedef as described in <sys/types.h>. See types.h(3HEAD).

Inclusion of the <sys/resource.h> header can also make visible all symbols from <sys/time.h>. See time.h(3HEAD).

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also  getrlimit(2), getpriority(3C), time.h(3HEAD), types.h(3HEAD), attributes(5), standards(5)
The `sched.h` header defines the `sched_param` structure, which contains the scheduling parameters required for implementation of each supported scheduling policy. This structure contains the following member:

```
int sched_priority   process execution scheduling priority
```

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 polices.

The scheduling policies are indicated by the values of the following symbolic constants:

- `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 `sched_rr_get_interval(3RT)` or by the system.

- `SCHED_IDLE` Processes are scheduled according to the Inter-Active Class (IA) policy as described in `priocntl(2)`.

- `SCHED_OTHER` Processes are scheduled according to another policy not described above.

The values of these constants are distinct.

Inclusion of the `sched.h` header will make visible symbols defined in the header `time.h`.

**See Also** `priocntl(2), sched_rr_get_interval(3RT), time.h(3HEAD)`
#include <search.h>

The <search.h> header defines the ENTRY type for structure entry, which includes the following members:

```c
char *key
void *data
```

and defines ACTION and VISIT as enumeration data types through type definitions as follows:

```c
eenum { FIND, ENTER } ACTION;
eenum { preorder, postorder, endorder, leaf } VISIT;
```

The size_t type is defined as described in <sys/types.h>. See types.h(3HEAD).

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also** hsearch(3C), insque(3C), lsearch(3C), tsearch(3C), types.h(3HEAD), attributes(5), standards(5)
**Name**
select.h, select – select types

**Synopsis**
#include <sys/select.h>

**Description**
The <sys/select.h> header defines the timeval structure, which includes the following members:

```c
  time_t  tv_sec    /* seconds */
  suseconds_t tv_usec /* microseconds */
```

The `time_t` and `suseconds_t` types are defined as described in <sys/types.h>. See types.h(3HEAD).

The `sigset_t` type is defined as described in signal.h(3HEAD).

The timespec structure is defined as described in <time.h>. See time.h(3HEAD).

The <sys/select.h> header defines the `fd_set` type as a structure.

The following is defined as a macro:

```c
  FD_SETSIZE       Maximum number of file descriptors in an fd_set structure.
```

Inclusion of the <sys/select.h> header can make visible all symbols from the headers <signal.h>, <sys/time.h>, and <time.h>.

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

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**
select(3C), signal.h(3HEAD), time.h(3HEAD), types.h(3HEAD), attributes(5), standards(5)
Name  semaphore.h, semaphore – semaphores

Synopsis  #include <semaphore.h>

Description  The <semaphore.h> header defines the sem_t type, used in performing semaphore operations. The semaphore can be implemented using a file descriptor, in which case applications are able to open up at least a total of [OPEN_MAX] files and semaphores. The symbol SEM_FAILED is defined (see sem_open(3RT)).

Inclusion of the <semaphore.h> header can make visible symbols defined in the headers <fcntl.h> and <sys/types.h>. See fcntl.h(3HEAD) and types.h(3HEAD).

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also  fcntl.h(3HEAD), types.h(3HEAD), sem_destroy(3RT), sem_getvalue(3RT), sem_init(3RT), sem_open(3RT), sem_post(3RT), sem_timedwait(3RT), sem_unlink(3RT), sem_wait(3RT), attributes(5), standards(5)
# sem.h

## Synopsis
```
#include <sys/sem.h>
```

## Description
The `<sys/sem.h>` header defines the following constants and structures.

Semaphore operation flags:
```
SEM_UNDO  Set up adjust on exit entry.
```

Command definitions for the `semctl()` function are provided as listed below. See `semctl(2)`.
```
GETNCNT  Get `semncnt`
GETPID   Get `sempid`
GETVAL   Get `semval`
GETALL   Get all cases of `semval`
GETZCNT  Get `semzcnt`
SETVAL   Set `semval`
SETALL   Set all cases of `semval`
```

The `semid_ds` structure contains the following members:
```
struct ipc_perm sem_perm    /* operation permission structure */
unsigned short sem_nsems    /* number of semaphores in set */
time_t    sem_otime         /* last semop() time */
time_t    sem_ctime         /* last time changed by semctl() */
```

The `pid_t`, `time_t`, `key_t`, and `size_t` types are defined as described in `<sys/types.h>`. See `types.h(3HEAD)`.

A semaphore is represented by an anonymous structure containing the following members:
```
unsigned short semval   /* semaphore value */
pid_t      sempid       /* process ID of last operation */
unsigned short semncnt  /* number of processes waiting for semval to become greater than current value */
unsigned short semzcnt  /* number of processes waiting for semval to become 0 */
```

The `sembuf` structure contains the following members:
```
unsigned short sem_num   /* semaphore number */
short      sem_op        /* semaphore operation */
short      sem_flg       /* operation flags */
```

All of the symbols from `<sys/ipc.h>` are defined when this header is included. See `ipc.h(3HEAD)`.
Attributes  See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also  semctl(2), semget(2), semop(2), ipc.h(3HEAD), types.h(3HEAD), attributes(5), standards(5)
setjmp.h header defines the array types jmp_buf and sigjmp_buf. Applications must define the appropriate feature test macro to enable the visibility of the symbols in this header.

Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also

_longjmp(3C), setjmp(3C), attributes(5), standards(5)
#include <sys/shm.h>

The `<sys/shm.h>` header defines the following symbolic constants:

- **SHM_RDONLY** attach read-only (else read-write)
- **SHM_RND** round attach address to SHMLBA

The `<sys/shm.h>` header defines the following symbolic value:

- **SHMLBA** segment low boundary address multiple

The following data types are defined through `typedef`:

- **shmatt_t** Unsigned integer used for the number of current attaches that must be able to store values at least as large as a type unsigned short.

The `shmid_ds` structure contains the following members:

- `struct ipc_perm shm_perm /* operation permission structure */`
- `size_t shm_segsz /* size of segment in bytes */`
- `pid_t shm_lpid /* process ID of last shared memory operation */`
- `pid_t shm_cpid /* process ID of creator */`
- `shmatt_t shm_nattch /* number of current attaches */`
- `time_t shm_atime /* time of last shmat() */`
- `time_t shm_dtime /* time of last shmdt() */`
- `time_t shm_ctime /* time of last change by shmctl() */`

The `pid_t`, `time_t`, `key_t`, and `size_t` types are defined as described in `<sys/types.h>`. See `types.h(3HEAD)`.

In addition, all of the symbols from `<sys/ipc.h>` are defined when this header is included.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also** `shmctl(2), shmget(2), shmp(2), ipc(3HEAD), types.h(3HEAD), attributes(5), standards(5)`
siginfo.h, siginfo – signal generation information

**Synopsis**

```
#include <siginfo.h>
```

**Description**

If a process is catching a signal, it might request information that tells why the system generated that signal. See `sigaction(2)`. If a process is monitoring its children, it might receive information that tells why a child changed state. See `waitid(2)`. In either case, the system returns the information in a structure of type `siginfo_t`, which includes the following information:

```c
int si_signo /* signal number */
int si_errno /* error number */
int si_code /* signal code */
union sigval si_value /* signal value */
```

`si_signo` contains the system-generated signal number. For the `waitid(2)` function, `si_signo` is always `SIGCHLD`.

If `si_errno` is non-zero, it contains an error number associated with this signal, as defined in `<errno.h>`.

`si_code` contains a code identifying the cause of the signal.

If the value of the `si_code` member is `SI_NOINFO`, only the `si_signo` member of `siginfo_t` is meaningful, and the value of all other members is unspecified.

**User Signals**

If the value of `si_code` is less than or equal to 0, then the signal was generated by a user process (see `kill(2), _lwp_kill(2), sigqueue(3RT), sigsend(2), abort(3C), and raise(3C)`) and the `siginfo` structure contains the following additional information:

```c
pid_t si_pid /* sending process ID */
uid_t si_uid /* sending user ID */
ctid_t si_ctid /* sending contract ID */
zoneid_t si_zoneid /* sending zone ID */
```

If the signal was generated by a user process, the following values are defined for `si_code`:

- **SI_USER**
  - The implementation sets `si_code` to `SI_USER` if the signal was sent by `kill(2), sigsend(2), raise(3C) or abort(3C)`.

- **SI_LWP**
  - The signal was sent by `_lwp_kill(2)`.

- **SI_QUEUE**
  - The signal was sent by `sigqueue(3RT)`.

- **SI_TIMER**
  - The signal was generated by the expiration of a timer created by `timer_settime(3RT)`.

- **SI_ASYNCIO**
  - The signal was generated by the completion of an asynchronous I/O request.

- **SI_MESGQ**
  - The signal was generated by the arrival of a message on an empty message queue. See `mq_notify(3RT)`.
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:

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

<table>
<thead>
<tr>
<th>Signal</th>
<th>Code</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIGXRES</td>
<td>SI_RCTL</td>
<td>resource–control generated signal</td>
</tr>
<tr>
<td>SIGHUP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIGTERM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>Code</th>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SI_RCTL</td>
<td>hr_time</td>
<td>process-model entity of control</td>
</tr>
</tbody>
</table>

In addition, the following signal-dependent information is available for kernel-generated signals:
<table>
<thead>
<tr>
<th>Signal</th>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIGILL</td>
<td>caddr_t si_addr</td>
<td>address of faulting instruction</td>
</tr>
<tr>
<td>SIGFPE</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIGSEGV</td>
<td>caddr_t si_addr</td>
<td>address of faulting memory reference</td>
</tr>
<tr>
<td>SIGBUS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIGCHLD</td>
<td>pid_t si_pid</td>
<td>child process ID</td>
</tr>
<tr>
<td></td>
<td>int si_status</td>
<td>exit value or signal</td>
</tr>
<tr>
<td>SIGPOLL</td>
<td>long si_band</td>
<td>band event for POLL_IN, POLL_OUT, or POLL_MSG</td>
</tr>
</tbody>
</table>

**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.h(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.
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 kill(2) or sigsend(2) 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 siginfo.h(3HEAD).

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 SIGSEGV or SIGBUS; a floating point exception can generate SIGFPE; and the execution of an illegal instruction can generate SIGILL. 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.

Signals may also be generated by calling kill(), sigqueue(), or sigsend(). Events such as keyboard interrupts generate signals, such as SIGINT, 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:

- The thread is blocked in a call to sigwait(2) whose argument includes the type of signal generated.
- The thread has a signal mask that does not include the type of signal generated. See pthread_sigmask(3C). Each process can specify a system action to be taken in response to each signal sent to it, called the signal’s disposition. All threads 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 sigaction(), signal() or sigset(), or until the process execs(). See sigaction(2) and signal(3C). When a process execs, all signals whose disposition has been set to catch the signal will be set to SIG_DFL. Alternatively, a process may request that the system automatically reset the disposition of a signal to SIG_DFL after it has been caught. See sigaction(2) and signal(3C).

A signal is said to be delivered to a process when a thread 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 sigaction(2). The second method uses sigwait(2) to create a thread to handle the receipt of the signal. The sigaction() function can be used for both synchronously and asynchronously generated signals. The sigwait() function will work only for asynchronously generated signals, as synchronously generated signals are sent to the thread that caused the event. The sigwait() function is the recommended for use with a multithreaded application.
Each thread has a signal mask that defines the set of signals currently blocked from delivery to it. The signal mask of the main thread is inherited from the signal mask of the thread that created it in the parent process. The selection of the thread 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 in the receiving process. Signals that are generated by action of a particular thread such as hardware faults are delivered to the thread that caused the signal. See `<pthread_sigmask(3C)>` or `<sigprocmask(2)>`. See `<alarm(2)>` for current semantics of delivery of `SIGALRM`. Signals that are directed to a particular thread are delivered to the targeted thread. See `<pthread_kill(3C)>`. If the selected thread has blocked the signal, it remains pending on the thread 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) one of the threads that does not have the signal blocked is selected to process the signal. If all the threads within the process block the signal, it remains pending on the process until a thread 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 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:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Default</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIGHUP</td>
<td>1</td>
<td>Exit</td>
<td>Hangup (see <code>&lt;termio(7I)&gt;</code>)</td>
</tr>
<tr>
<td>SIGINT</td>
<td>2</td>
<td>Exit</td>
<td>Interrupt (see <code>&lt;termio(7I)&gt;</code>)</td>
</tr>
<tr>
<td>SIGQUIT</td>
<td>3</td>
<td>Core</td>
<td>Quit (see <code>&lt;termio(7I)&gt;</code>)</td>
</tr>
<tr>
<td>SIGILL</td>
<td>4</td>
<td>Core</td>
<td>Illegal Instruction</td>
</tr>
<tr>
<td>SIGTRAP</td>
<td>5</td>
<td>Core</td>
<td>Trace or Breakpoint Trap</td>
</tr>
<tr>
<td>SIGABRT</td>
<td>6</td>
<td>Core</td>
<td>Abort</td>
</tr>
<tr>
<td>SIGEMT</td>
<td>7</td>
<td>Core</td>
<td>Emulation Trap</td>
</tr>
<tr>
<td>SIGFPE</td>
<td>8</td>
<td>Core</td>
<td>Arithmetic Exception</td>
</tr>
<tr>
<td>SIGKILL</td>
<td>9</td>
<td>Exit</td>
<td>Killed</td>
</tr>
<tr>
<td>SIGBUS</td>
<td>10</td>
<td>Core</td>
<td>Bus Error</td>
</tr>
<tr>
<td>SIGSEGV</td>
<td>11</td>
<td>Core</td>
<td>Segmentation Fault</td>
</tr>
<tr>
<td>SIGSYS</td>
<td>12</td>
<td>Core</td>
<td>Bad System Call</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Default</td>
<td>Event</td>
</tr>
<tr>
<td>-------------</td>
<td>-------</td>
<td>---------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>SIGPIPE</td>
<td>13</td>
<td>Exit</td>
<td>Broken Pipe</td>
</tr>
<tr>
<td>SIGALRM</td>
<td>14</td>
<td>Exit</td>
<td>Alarm Clock</td>
</tr>
<tr>
<td>SIGTERM</td>
<td>15</td>
<td>Exit</td>
<td>Terminated</td>
</tr>
<tr>
<td>SIGUSR1</td>
<td>16</td>
<td>Exit</td>
<td>User Signal 1</td>
</tr>
<tr>
<td>SIGUSR2</td>
<td>17</td>
<td>Exit</td>
<td>User Signal 2</td>
</tr>
<tr>
<td>SIGCHLD</td>
<td>18</td>
<td>Ignore</td>
<td>Child Status Changed</td>
</tr>
<tr>
<td>SIGPWR</td>
<td>19</td>
<td>Ignore</td>
<td>Power Fail or Restart</td>
</tr>
<tr>
<td>SIGWINCH</td>
<td>20</td>
<td>Ignore</td>
<td>Window Size Change</td>
</tr>
<tr>
<td>SIGURG</td>
<td>21</td>
<td>Ignore</td>
<td>Urgent Socket Condition</td>
</tr>
<tr>
<td>SIGPOLL</td>
<td>22</td>
<td>Exit</td>
<td>Pollable Event (see streamio(7I))</td>
</tr>
<tr>
<td>SIGSTOP</td>
<td>23</td>
<td>Stop</td>
<td>Stopped (signal)</td>
</tr>
<tr>
<td>SIGTSTP</td>
<td>24</td>
<td>Stop</td>
<td>Stopped (user) (see termio(7I))</td>
</tr>
<tr>
<td>SIGCONT</td>
<td>25</td>
<td>Ignore</td>
<td>Continued</td>
</tr>
<tr>
<td>SIGTTIN</td>
<td>26</td>
<td>Stop</td>
<td>Stopped (tty input) (see termio(7I))</td>
</tr>
<tr>
<td>SIGTTOU</td>
<td>27</td>
<td>Stop</td>
<td>Stopped (tty output) (see termio(7I))</td>
</tr>
<tr>
<td>SIGVTALRM</td>
<td>28</td>
<td>Exit</td>
<td>Virtual Timer Expired</td>
</tr>
<tr>
<td>SIGPROF</td>
<td>29</td>
<td>Exit</td>
<td>Profiling Timer Expired</td>
</tr>
<tr>
<td>SIGXCPU</td>
<td>30</td>
<td>Core</td>
<td>CPU time limit exceeded (see getrlimit(2))</td>
</tr>
<tr>
<td>SIGXFSZ</td>
<td>31</td>
<td>Core</td>
<td>File size limit exceeded (see getrlimit(2))</td>
</tr>
<tr>
<td>SIGWAITING</td>
<td>32</td>
<td>Ignore</td>
<td>Reserved</td>
</tr>
<tr>
<td>SIGLWP</td>
<td>33</td>
<td>Ignore</td>
<td>Reserved</td>
</tr>
<tr>
<td>SIGFREEZE</td>
<td>34</td>
<td>Ignore</td>
<td>Check point Freeze</td>
</tr>
<tr>
<td>SIGTHAW</td>
<td>35</td>
<td>Ignore</td>
<td>Check point Thaw</td>
</tr>
<tr>
<td>SIGCANCEL</td>
<td>36</td>
<td>Ignore</td>
<td>Reserved for threading support</td>
</tr>
<tr>
<td>SIGLOST</td>
<td>37</td>
<td>Exit</td>
<td>Resource lost (for example, record–lock lost)</td>
</tr>
<tr>
<td>SIGXRES</td>
<td>38</td>
<td>Ignore</td>
<td>Resource control exceeded (see setrctl(2))</td>
</tr>
<tr>
<td>SIGJVM1</td>
<td>39</td>
<td>Ignore</td>
<td>Reserved for Java Virtual Machine 1</td>
</tr>
<tr>
<td>SIGJVM2</td>
<td>40</td>
<td>Ignore</td>
<td>Reserved for Java Virtual Machine 2</td>
</tr>
</tbody>
</table>
The symbols SIGRTMIN through SIGRTMAX are evaluated dynamically to permit future configurability.

Applications should not use any of the signals marked "reserved" in the above table for any purpose, to avoid interfering with their use by the system.

SIGNAL DISPOSITION
A process using a signal(3C), sigset(3C) or sigaction(2) system call can 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 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 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(2) function, however, additional arguments can be requested. 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.

If the disposition has been set with the sigset() or sigaction(), the signal is automatically blocked in the thread 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. If the SA_RESTART flag is set, however, certain function calls 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:

<table>
<thead>
<tr>
<th>Member</th>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>int</td>
<td>sigev_notify</td>
<td>Notification type</td>
</tr>
<tr>
<td>int</td>
<td>sigev_signo</td>
<td>Signal number</td>
</tr>
<tr>
<td>union sigval</td>
<td>sigev_value</td>
<td>Signal value</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Member</th>
<th>Member</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Name</td>
<td></td>
</tr>
<tr>
<td>int</td>
<td>sival_int</td>
<td>Integer signal value</td>
</tr>
<tr>
<td>void *</td>
<td>sival_ptr</td>
<td>Pointer signal value</td>
</tr>
</tbody>
</table>

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:

SIGEV_NONE     No asynchronous notification is delivered when the event of interest occurs.
SIGEV_SIGNAL   A queued signal, with its value application-defined, is generated when the event of interest occurs.
An asynchronous notification is delivered to an event port when the event of interest occurs. The sival_ptr member points to a port_notify_t structure (see port_associate(3C)). The event port identifier as well as an application-defined cookie are part of the port_notify_t structure.

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.

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**

lockd(1M), Intro(2), alarm(2), exit(2), fcntl(2), getrlimit(2), ioctl(2), kill(2), pause(2), setrlimit(2), sigaction(2), sigaltstack(2), sigprocmask(2), sigsend(2), sigsuspend(2), sigwait(2), port_associate(3C), pthread_create(3C), pthread_kill(3C), pthread_sigmask(3C), setjmp(3C), siginfo.h(3HEAD), signal(3C), sigqueue(3RT), sigsetops(3C), thr_create(3C), thr_kill(3C), thr_sigsetmask(3C), ucontext.h(3HEAD), wait(3C), attributes(5), standards(5)

**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, SIGSTOP, and SIGCANCEL signals cannot be blocked. The system silently enforces this restriction.

The SIGCANCEL signal cannot be directed to an individual thread using pthread_kill(3C), but it can be sent to a process using kill(2), sigsend(2), or sigqueue(3RT).
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_NOCLOSE 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.

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.
socket.h (3HEAD)

Name socket.h, socket – Internet Protocol family

Synopsis #include <sys/socket.h>

Description The <sys/socket.h> header defines the unsigned integral type sa_family_t through typedef.

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

- sa_family_t sa_family /* address family */
- char sa_data[ ] /* socket address (variable-length data) */

libxnet Interfaces The <sys/socket.h> header defines the msghdr structure for libxnet interfaces that includes the following members:

- void *msg_name /* optional address */
- socklen_t msg_namelen /* size of address */
- struct iovec *msg_iov /* scatter/gather array */
- int msg_iovlen /* members in msg_iov */
- void *msg_control /* ancillary data, see below */
- socklen_t msg_controllen /* ancillary data buffer len */
- int msg_flags /* flags on received message */

The <sys/socket.h> header defines the cmsghdr structure for libxnet that includes the following members:

- socklen_t cmsg_len /* data byte count, including hdr */
- int cmsg_level /* originating protocol */
- int cmsg_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 cmsg_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 macros for use as the cmsg_type values when cmsg_level is SOL_SOCKET.

- SCM_RIGHTS Indicates that the data array contains the access rights to be sent or received.
- SCM_UCREDS Indicates that the data array contains a ucred_t to be received. The ucred_t is the credential of the sending process at the time the message was sent. This is a Sun-specific, Evolving interface. See ucred_get(3C).
The IPv4 data formats generally use the same values for data passed back in cmsghdr as for setsockopt() to enable the feature. The IPv4 data formats are listed below with the associated payload for each.

IPPROTO_IP
IP_RECV DSTADDR
   ipaddr_t, IP address

IPPROTO_IP
IP_RECV OPTS
   variable-length IP options, up to 40 bytes

IPPROTO_IP
IP_RECV IF
   uint_t, ifIndex number

IPPROTO_IP
IP_RECV SLLA
   struct sockaddr_dl, link layer address

IPPROTO_IP
IP_RECV TTL
   uint8_t

SOL_SOCKET
SO_RECV UCRED
   ucred_t — cmsghdr.cmsg_type is SCM_UCRED, not SO_RECV UCRED

The IPv6 data formats use different values for enabling the option and for passing the value back to the application. The IPv6 data formats are listed below with the associated payload for each.

IPPROTO_IPV6
IPV6_RECV PKTINFO
   in_pktinfo, cmsg_type IPV6_PKTINFO

IPPROTO_IPV6
IPV6_RECV TCLASS
   uint_t, cmsg_type IPV6_TCLASS

IPPROTO_IPV6
IPV6_RECV PATHMTU
   ip6_mtuinfo, cmsg_type IPV6_PATHMTU

IPPROTO_IPV6
IPV6_RECV HOP LIMIT
   uint_t, cmsg_type IPV6_HOPLIMIT

IPPROTO_IPV6
IPV6_RECV HOPOPTS
variable-length IPv6 options, cmsg_type IPV6_HOPOPTS
IPPROTO_IPV6
IPV6_RECVDSTOPTS
variable-length IPv6 options, cmsg_type IPV6_DSTOPTS
IPPROTO_IPV6
IPV6_RECVRTHDR
variable-length IPv6 options, cmsg_type IPV6_RTHDR
IPPROTO_IPV6
IPV6_RECVRT HDRDSTOPTS
variable-length IPv6 options, cmsg_type IPV6_DSTOPTS

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:

**CMSG_DATA(cmsg)**
- 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.

**CMSG_NXTHDR(mhdr, cmsg)**
- If the first argument is a pointer to a msghdr 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 msghdr 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.

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

**CMSG_SPACE(len)**
- Given the length of an ancillary data object, CMSG_SPACE() returns the space required by the object and its cmsghdr structure, including any padding needed to satisfy alignment requirements. This macro can be used, for example, to allocate space dynamically for the ancillary data. This macro should not be used to initialize the cmsg_len member of a cmsghdr structure. Use the CMSG_LEN() macro instead.

**CMSG_LEN(len)**
- Given the length of an ancillary data object, CMSG_LEN() returns the value to store in the cmsg_len member of the cmsghdr structure, taking into account any padding needed to satisfy alignment requirements.

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

```c
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_KEEPALIVE** 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.
- **SO_RECVCRED** Request the reception of user credential ancillary data. This is a Sun-specific, Evolving interface. See `ucred_get(3C)`.
- **SO_MAC_EXEMPT** Mandatory Access Control (MAC) exemption for unlabeled peers. This option is available only if the system is configured with Trusted Extensions.
- **SO_ALLZONES** Bypass zone boundaries (privileged).

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

- **MSG_CTRUNC** 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.
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 <sys/socket.h> header defines the msghdr structure for libsocket interfaces that includes the following members:

void *msg_name /* optional address */
socklen_t msg_namelen /* size of address */
struct iovec *msg_iov /* scatter/gather array */
int msg_iovlen /* # elements in msg_iov */
caddr_t msg_accrights /* access rights sent/received */

The msg_name and msg_namelen parameters specify the destination address when the socket is unconnected. The msg_name can be specified as a NULL pointer if no names are desired or required. The msg_iov and msg_iovlen parameters describe the scatter-gather locations, as described in read(2). The msg_accrights parameter specifies the buffer in which access rights sent along with the message are received. The msg_accrightslen specifies the length of the buffer.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also accept(3SOCKET), accept(3XNET), bind(3SOCKET), bind(3XNET), connect(3SOCKET), connect(3XNET), getpeername(3SOCKET), getpeername(3XNET), getpeerucred(3C), getsockname(3SOCKET), getsockname(3XNET), getsockopt(3SOCKET), getsockopt(3XNET), libsocket(3LIB), 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),
ucred_get(3C) attributes(5), standards(5)
Name    spawn.h, spawn - spawn

Synopsis    #include <spawn.h>

Description    The <spawn.h> header defines the posix_spawnattr_t and posix_spawn_file_actions_t
types used in performing spawn operations.

    The <spawn.h> header defines the flags that can be set in a posix_spawnattr_t object using
the posix_spawnattr_setflags() function:

    POSIX_SPAWN_RESETIDS
    POSIX_SPAWN_SETPGROUP
    POSIX_SPAWN_SETSCHEDPARAM
    POSIX_SPAWN_SETSCHEDULER
    POSIX_SPAWN_SETSIGDEF
    POSIX_SPAWN_SETSIGMASK

    Inclusion of the <spawn.h> header can make visible symbols defined in the <sched.h>,
    <signal.h>, and <sys/types.h> headers.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also    sched.h(3HEAD), semaphore.h(3HEAD), signal.h(3HEAD), types.h(3HEAD),
            attributes(5), standards(5)
Name

stat.h, stat – data returned by stat system call

Synopsis

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

Description

The system calls stat(), lstat() and fstat() return data in a stat structure, which is defined in <stat.h>.

The constants used in the st_mode field are also defined in this file:

#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 */
#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) blockspecialfile
#define S_ISCHR(mode) characterspecialfile
#define S_ISDIR(mode) directoryfile
#define S_ISFIFO(mode) pipeorfifofile
#define S_ISREG(mode) regularfile
#define S_ISSOCK(mode) socketfile

The following symbolic constants are defined as distinct integer values outside of the range [0, 999 999 999], for use with the futimens() and utimensat() functions:

#define UTIME_NOW use the current time
#define UTIME_OMIT no time change

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Committed</td>
</tr>
<tr>
<td>Standard</td>
<td>See standards(5).</td>
</tr>
</tbody>
</table>

See Also  futimens(2), stat(2), types.h(3HEAD), attributes(5), standards(5)
# Name
statvfs.h, statvfs – VFS File System information structure

## Synopsis
```
#include <sys/statvfs.h>
```

## Description
The `<sys/statvfs.h>` header defines the `statvfs` structure, which includes the following members:

```c
unsigned long f_bsize /* file system block size */
unsigned long f_frsize /* fundamental file system block size */
fsblkcnt_t f_blocks /* total number of blocks on file system in units of f_frsize */
fsblkcnt_t f_bfree /* total number of free blocks */
fsblkcnt_t f_bavail /* number of free blocks available to non-privileged process */
fsfilcnt_t f_files /* total number of file serial numbers */
fsfilcnt_t f_ffree /* total number of free file serial numbers */
fsfilcnt_t f_favail /* number of file serial numbers available to non-privileged */
unsigned long f_fsid /* process file system ID */
unsigned long f_flag /* bit mask of f_flag values */
unsigned long f_namemax /* maximum filename length */
```

The `fsblkcnt_t` and `fsfilcnt_t` types are defined as described in `<sys/types.h>`. See `types.h(3HEAD)`.

The following flags for the `f_flag` member are defined:

- ST_RDONLY: read-only file system
- ST_NOSUID: does not support setuid() / setgid() semantics

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

<table>
<thead>
<tr>
<th>ATTRIBUTETYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

## See Also
`statvfs(2), types.h(3HEAD), attributes(5), standards(5)`
#include <stdbool.h>  

The `<stdbool.h>` header defines the following macros:

- `true` expands to the integer constant 1
- `false` expands to the integer constant 0
- `__bool_true_false_are_defined` expands to the integer constant 1

An application can undefine and then possibly redefine the macros `bool`, `true`, and `false`.

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**

attributes(5), standards(5)
**#include <stddef.h>**

The `<stddef.h>` header defines the following macros:

- `NULL` - Null pointer constant.
- `offsetof(type, member-designator)` - Integer constant expression of type `size_t`, the value of which is the offset in bytes to the structure member (member-designator), from the beginning of its structure (type).

The `<stddef.h>` header defines the following types:

- `ptrdiff_t` - Signed integer type of the result of subtracting two pointers.
- `wchar_t` - Integer type whose range of values can represent distinct wide-character codes for all members of the largest character set specified among the locales supported by the compilation environment: the null character has the code value 0 and each member of the portable character set has a code value equal to its value when used as the lone character in an integer character constant.
- `size_t` - Unsigned integer type of the result of the `sizeof` operator.

The implementation supports one or more programming environments in which the widths of `ptrdiff_t`, `size_t`, and `wchar_t` are no greater than the width of type `long`. The names of these programming environments can be obtained using the `confstr(3C)` function or the `getconf(1)` utility.

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**

`getconf(1), confstr(3C), types.h(3HEAD), wchar.h(3HEAD), attributes(5), standards(5)`
**Synopsis**

The `<stdint.h>` header declares sets of integer types having specified widths, and defines corresponding sets of macros. It also defines macros that specify limits of integer types corresponding to types defined in other standard headers.

Types are defined in the following categories:

- integer types having certain exact widths
- integer types having at least certain specified widths
- fastest integer types having at least certain specified widths
- integer types wide enough to hold pointers to objects
- integer types having greatest width

Some of these types may denote the same type.

Corresponding macros specify limits of the declared types and construct suitable constants.

For each type described herein that the implementation provides, the `<stdint.h>` header declares that `typedef` name and defines the associated macros. Conversely, for each type described herein that the implementation does not provide, the `<stdint.h>` header does not declare that `typedef` name, nor does it define the associated macros. An implementation provides those types described as required, but need not provide any of the others (described as optional).

**Exact-width integer types**

When `typedef` names differing only in the absence or presence of the initial `u` are defined, they denote corresponding signed and unsigned types as described in the ISO/IEC 9899: 1999 standard, Section 6.2.5; an implementation providing one of these corresponding types must also provide the other.

In the following descriptions, the symbol `N` represents an unsigned decimal integer with no leading zeros (for example, 8 or 24, but not 04 or 048).

The `typedef` name `intN_t` designates a signed integer type with width `N`, no padding bits, and a two’s-complement representation. Thus, `int8_t` denotes a signed integer type with a width of exactly 8 bits.

The `typedef` name `uintN_t` designates an unsigned integer type with width `N`. Thus, `uint24_t` denotes an unsigned integer type with a width of exactly 24 bits.

The following types are required:

<table>
<thead>
<tr>
<th>Name</th>
<th>stdint.h, stdint – integer types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synopsis</td>
<td><code>#include &lt;stdint.h&gt;</code></td>
</tr>
<tr>
<td>Description</td>
<td>The <code>&lt;stdint.h&gt;</code> header declares sets of integer types having specified widths, and defines corresponding sets of macros. It also defines macros that specify limits of integer types corresponding to types defined in other standard headers. The “width” of an integer type is the number of bits used to store its value in a pure binary system; the actual type can use more bits than that (for example, a 28-bit type could be stored in 32 bits of actual storage). An N-bit signed type has values in the range $-2^{N-1}$ or $1 - 2^{N-1}$ to $2^{N-1} - 1$, while an N-bit unsigned type has values in the range 0 to $2^{N-1}$. Types are defined in the following categories:</td>
</tr>
<tr>
<td></td>
<td>- integer types having certain exact widths</td>
</tr>
<tr>
<td></td>
<td>- integer types having at least certain specified widths</td>
</tr>
<tr>
<td></td>
<td>- fastest integer types having at least certain specified widths</td>
</tr>
<tr>
<td></td>
<td>- integer types wide enough to hold pointers to objects</td>
</tr>
<tr>
<td></td>
<td>- integer types having greatest width</td>
</tr>
<tr>
<td></td>
<td>Some of these types may denote the same type. Corresponding macros specify limits of the declared types and construct suitable constants. For each type described herein that the implementation provides, the <code>&lt;stdint.h&gt;</code> header declares that <code>typedef</code> name and defines the associated macros. Conversely, for each type described herein that the implementation does not provide, the <code>&lt;stdint.h&gt;</code> header does not declare that <code>typedef</code> name, nor does it define the associated macros. An implementation provides those types described as required, but need not provide any of the others (described as optional).</td>
</tr>
</tbody>
</table>

When `typedef` names differing only in the absence or presence of the initial `u` are defined, they denote corresponding signed and unsigned types as described in the ISO/IEC 9899: 1999 standard, Section 6.2.5; an implementation providing one of these corresponding types must also provide the other. In the following descriptions, the symbol `N` represents an unsigned decimal integer with no leading zeros (for example, 8 or 24, but not 04 or 048). Exact-width integer types

The `typedef` name `intN_t` designates a signed integer type with width `N`, no padding bits, and a two’s-complement representation. Thus, `int8_t` denotes a signed integer type with a width of exactly 8 bits.

The `typedef` name `uintN_t` designates an unsigned integer type with width `N`. Thus, `uint24_t` denotes an unsigned integer type with a width of exactly 24 bits.

The following types are required:
If an implementation provides integer types with width 64 that meet these requirements, then the following types are required:

```c
int64_t
uint64_t
```

In particular, this is the case if any of the following are true:

- The implementation supports the `_POSIX_V6_ILP32_OFFBIG` programming environment and the application is being built in the `_POSIX_V6_ILP32_OFFBIG` programming environment (see the Shell and Utilities volume of IEEE Std 1003.1-200x, c99, Programming Environments).
- The implementation supports the `_POSIX_V6_LP64_OFF64` programming environment and the application is being built in the `_POSIX_V6_LP64_OFF64` programming environment.
- The implementation supports the `_POSIX_V6_LPBIG_OFFBIG` programming environment and the application is being built in the `_POSIX_V6_LPBIG_OFFBIG` programming environment.

All other types of this form are optional.

Minimum-width integer types

The typedef name `int_leastN_t` designates a signed integer type with a width of at least \( N \), such that no signed integer type with lesser size has at least the specified width. Thus, `int_least32_t` denotes a signed integer type with a width of at least 32 bits.

The typedef name `uint_leastN_t` designates an unsigned integer type with a width of at least \( N \), such that no unsigned integer type with lesser size has at least the specified width. Thus, `uint_least16_t` denotes an unsigned integer type with a width of at least 16 bits.

The following types are required:

```c
int_least8_t
int_least16_t
int_least32_t
int_least64_t
uint_least8_t
uint_least16_t
uint_least32_t
uint_least64_t
```

All other types of this form are optional.
Fastest minimum-width integer types
Each of the following types designates an integer type that is usually fastest to operate with among all integer types that have at least the specified width.

The designated type is not guaranteed to be fastest for all purposes; if the implementation has no clear grounds for choosing one type over another, it will simply pick some integer type satisfying the signedness and width requirements.

The typedef name int_fastN_t designates the fastest signed integer type with a width of at least N. The typedef name uint_fastN_t designates the fastest unsigned integer type with a width of at least N.

The following types are required:

- int_fast8_t
- int_fast16_t
- int_fast32_t
- int_fast64_t
- uint_fast8_t
- uint_fast16_t
- uint_fast32_t
- uint_fast64_t

All other types of this form are optional.

Integer types capable of holding object pointers

- intptr_t Designates a signed integer type with the property that any valid pointer to void can be converted to this type, then converted back to a pointer to void, and the result will compare equal to the original pointer.

- uintptr_t Designates an unsigned integer type with the property that any valid pointer to void can be converted to this type, then converted back to a pointer to void, and the result will compare equal to the original pointer.

On standard-conforming systems, the intptr_t and uintptr_t types are required; otherwise, they are optional.

Greatest-width integer types

- intmax_t Designates a signed integer type capable of representing any value of any signed integer type.

- uintmax_t Designates an unsigned integer type capable of representing any value of any unsigned integer type.

These types are required.

Applications can test for optional types by using the corresponding limit macro from Limits of Specified-Width Integer Types.
The following macros specify the minimum and maximum limits of the types declared in the `<stdint.h>` header. Each macro name corresponds to a similar type name in Integer Types.

Each instance of any defined macro is replaced by a constant expression suitable for use in `#if` preprocessing directives. This expression has the same type as would an expression that is an object of the corresponding type converted according to the integer promotions. Its implementation-defined value is equal to or greater in magnitude (absolute value) than the corresponding value given below, with the same sign, except where stated to be exactly the given value.

Limits of exact-width integer types

- Minimum values of exact-width signed integer types:
  \( \{ \text{INT}_N\_MIN \} \) Exactly \(-(2^{N-1})\)

- Maximum values of exact-width signed integer types:
  \( \{ \text{INT}_N\_MAX \} \) Exactly \(2^{N-1}-1\)

- Maximum values of exact-width unsigned integer types:
  \( \{ \text{UINT}_N\_MAX \} \) Exactly \(2^N-1\)

Limits of minimum-width integer types

- Minimum values of minimum-width signed integer types:
  \( \{ \text{INT}_\text{LEAST}_N\_MIN \} \) \(-(2^{N-1}-1)\)

- Maximum values of minimum-width signed integer types:
  \( \{ \text{INT}_\text{LEAST}_N\_MAX \} \) \(2^{N-1}-1\)

- Maximum values of minimum-width unsigned integer types:
  \( \{ \text{UINT}_\text{LEAST}_N\_MAX \} \) \(2^N-1\)

Limits of fastest minimum-width integer types

- Minimum values of fastest minimum-width signed integer types:
  \( \{ \text{INT}_\text{FAST}_N\_MIN \} \) \(-(2^{N-1}-1)\)

- Maximum values of fastest minimum-width signed integer types:
  \( \{ \text{INT}_\text{FAST}_N\_MAX \} \) \(2^{N-1}-1\)

- Maximum values of fastest minimum-width unsigned integer types:
  \( \{ \text{UINT}_\text{FAST}_N\_MAX \} \) \(2^{N-1}-1\)

Limits of integer types capable of holding object pointers

- Minimum value of pointer-holding signed integer type:
  \( \{ \text{INTPTR}_\text{MIN} \} \) \(-(2^{15}-1)\)
- Maximum value of pointer-holding signed integer type:
  \[ \text{INTPTR\_MAX} \quad 2^{15} - 1 \]
- Minimum value of pointer-holding signed integer type:
  \[ \text{UINTPTR\_MAX} \quad 2^{16} - 1 \]

Limits of greatest-width integer types
- Minimum value of greatest-width signed integer type:
  \[ \text{INTMAX\_MIN} \quad -(2^{63} - 1) \]
- Maximum value of greatest-width signed integer type:
  \[ \text{INTMAX\_MIN} \quad 2^{63} - 1 \]
- Maximum value of greatest-width unsigned integer type:
  \[ \text{UINTMAX\_MIN} \quad 2^{64} - 1 \]

The following macros specify the minimum and maximum limits of integer types corresponding to types defined in other standard headers.

Each instance of these macros is replaced by a constant expression suitable for use in \#if preprocessing directives. This expression has the same type as would an expression that is an object of the corresponding type converted according to the integer promotions. Its implementation-defined value is equal to or greater in magnitude (absolute value) than the corresponding value given below, with the same sign.

Limits of \texttt{ptrdiff\_t}:
- \[ \text{PTRDIFF\_MIN} \quad -65535 \]
- \[ \text{PTRDIFF\_MAX} \quad +65535 \]

Limits of \texttt{sig\_atomic\_t}:
- \[ \text{SIG\_ATOMIC\_MIN} \quad \text{See below.} \]
- \[ \text{SIG\_ATOMIC\_MAX} \quad \text{See below.} \]

Limits of \texttt{size\_t}:
- \[ \text{SIZE\_MAX} \quad 65535 \]

Limits of \texttt{wchar\_t}:
- \[ \text{WCHAR\_MIN} \quad \text{See below.} \]
- \[ \text{WCHAR\_MAX} \quad \text{See below.} \]
Limits of wint_t:

{WINT_MIN} See below.
{WINT_MAX} See below.

If sig_atomic_t (see the <signal.h> header) is defined as a signed integer type, the value of {SIG_ATOMIC_MIN} is no greater than -127 and the value of {SIG_ATOMIC_MAX} is no less than 127. Otherwise, sig_atomic_t is defined as an unsigned integer type, the value of {SIG_ATOMIC_MIN} is 0 and the value of {SIG_ATOMIC_MAX} is no less than 255.

If wchar_t (see the <stddef.h> header) is defined as a signed integer type, the value of {WCHAR_MIN} is no greater than -127 and the value of {WCHAR_MAX} is no less than 127. Otherwise, wchar_t is defined as an unsigned integer type, and the value of {WCHAR_MIN} is 0 and the value of {WCHAR_MAX} is no less than 255.

If wint_t (see the <wchar.h> header) is defined as a signed integer type, the value of {WINT_MIN} is no greater than -32767 and the value of {WINT_MAX} is no less than 32767. Otherwise, wint_t is defined as an unsigned integer type, and the value of {WINT_MIN} is 0 and the value of {WINT_MAX} is no less than 65535.

The following macros expand to integer constant expressions suitable for initializing objects that have integer types corresponding to types defined in the <stdint.h> header. Each macro name corresponds to a similar type name listed under minimum-width integer types and greatest-width integer types.

Each invocation of one of these macros expands to an integer constant expression suitable for use in #if preprocessing directives. The type of the expression has the same type as would an expression that is an object of the corresponding type converted according to the integer promotions. The value of the expression is that of the argument. The argument in any instance of these macros is a decimal, octal, or hexadecimal constant with a value that does not exceed the limits for the corresponding type.

Macros for minimum-width integer constant expressions

The macro INTN_C(value) expands to an integer constant expression corresponding to the type int_leastN_t. The macro UINTN_C(value) expands to an integer constant expression corresponding to the type uint_leastN_t. For example, if uint_least64_t is a name for the type unsigned long long, then UINT64_C(0x123) might expand to the integer constant 0x123ULL.

Macros for greatest-width integer constant expressions

The following macro expands to an integer constant expression having the value specified by its argument and the type intmax_t:

INTMAX_C(value)

The following macro expands to an integer constant expression having the value specified by its argument and the type uintmax_t:
UINTMAX_C(value)

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also  inttypes.h(3HEAD), signal.h(3HEAD), stddef.h(3HEAD), wchar.h(3HEAD), attributes(5), standards(5)
The `<stdio.h>` header defines the following macros as positive integer constant expressions:

- `BUFSIZ`: Size of `<stdio.h>` buffers
- `_IOFBF`: Input/output fully buffered
- `_IOLBF`: Input/output line buffered
- `_IONBF`: Input/output unbuffered
- `L_ctermid`: Maximum size of character array to hold `ctermid()` output
- `L_tmpnam`: Maximum size of character array to hold `tmpnam()` output
- `SEEK_CUR`: Seek relative to current position
- `SEEK_END`: Seek relative to end-of-file
- `SEEK_SET`: Seek relative to start-of-file

The following macros are defined as positive integer constant expressions that denote implementation limits:

- `{FILENAME_MAX}`: Maximum size in bytes of the longest filename string that the implementation guarantees can be opened.
- `{OPEN_MAX}`: Number of streams that the implementation guarantees can be open simultaneously. The value is at least eight.
- `{TMP_MAX}`: Minimum number of unique filenames generated by `tmpnam()`. Maximum number of times an application can call `tmpnam()` reliably. The value of `{TMP_MAX}` is at least 25. On XSI-conformant systems, the value of `{TMP_MAX}` is at least 10000.

The following macro name is defined as a negative integer constant expression:

- `EOF`: End-of-file return value

The following macro name is defined as a null pointer constant:

- `NULL`: Null pointer

The following macro name is defined as a string constant:

- `P_tmpdir`: Default directory prefix for `tmpnam()`

The following is defined as expressions of type “pointer to FILE” point to the FILE objects associated, respectively, with the standard error, input, and output streams:

- `stderr`: Standard error output stream
The following data types are defined through typedef:

- **FILE**: structure containing information about a file
- **fpos_t**: non-array type containing all information needed to specify uniquely every position within a file
- **va_list**: as described in `<stdarg.h>`
- **size_t**: as described in `<stddef.h>`

### Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

### See Also

rename(2), ctermid(3C), fclose(3C), fdopen(3C), fflush(3C), fgetc(3C), fgetpos(3C), fgets(3C), flockfile(3C), fopen(3C), fputc(3C), fputs(3C), fputwc(3C), fread(3C), freopen(3C), fseek(3C), fsetpos(3C), ftell(3C), fwrite(3C), getwchar(3C), getopt(3C), perror(3C), popen(3C), printf(3C), remove(3C), rewind(3C), scanf(3C), setbuf(3C), stdio(3C), system(3C), tmpfile(3C), tmpnam(3C), ungetc(3C), vprintf(3C), attributes(5), standards(5)
Name  stdlib.h, stdlib – standard library definitions

Synopsis  #include <stdlib.h>

Description  The <stdlib.h> header defines the following macros:

<table>
<thead>
<tr>
<th>Macro</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXIT_FAILURE</td>
<td>Unsuccessful termination for exit(); evaluates to a non-zero value. See exit(3C).</td>
</tr>
<tr>
<td>EXIT_SUCCESS</td>
<td>Successful termination for exit(); evaluates to 0.</td>
</tr>
<tr>
<td>NULL</td>
<td>Null pointer.</td>
</tr>
<tr>
<td>{RAND_MAX}</td>
<td>Maximum value returned by rand(); at least 32767. See rand(3C).</td>
</tr>
<tr>
<td>{MB_CUR_MAX}</td>
<td>Integer expression whose value is the maximum number of bytes in a character specified by the current locale.</td>
</tr>
</tbody>
</table>

The following data types are defined through typedef:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>div_t</td>
<td>structure type returned by the div() function</td>
</tr>
<tr>
<td>ldiv_t</td>
<td>structure type returned by the ldiv() function</td>
</tr>
<tr>
<td>lldiv_t</td>
<td>structure type returned by the lldiv() function</td>
</tr>
<tr>
<td>size_t</td>
<td>as described in &lt;stddef.h&gt;</td>
</tr>
<tr>
<td>wchar_t</td>
<td>as described in &lt;stddef.h&gt;</td>
</tr>
</tbody>
</table>

See div(3C), which covers div(), ldiv(), and lldiv(), and stddef.h(3HEAD).

In addition, the symbolic names and macros listed below are defined as in <sys/wait.h>, for use in decoding the return value from system(). See wait.h(3HEAD) and system(3C).

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WNOHANG</td>
<td></td>
</tr>
<tr>
<td>WUNTRACED</td>
<td></td>
</tr>
<tr>
<td>WEXITSTATUS</td>
<td></td>
</tr>
<tr>
<td>WIFEXITED</td>
<td></td>
</tr>
<tr>
<td>WIFSIGNALED</td>
<td></td>
</tr>
<tr>
<td>WIFSTOPPED</td>
<td></td>
</tr>
<tr>
<td>WSTOPSIG</td>
<td></td>
</tr>
<tr>
<td>WTERMSIG</td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>
See Also  a64l(3C), abort(3C), abs(3C), atexit(3C), bsearch(3C), div(3C), drand48(3C), exit(3C),
getenv(3C), getsubopt(3C), grantpt(3C), malloc(3C), mblen(3C), mbstowcs(3C),
mbtowc(3C), mkstemp(3C), ptsname(3C), putenv(3C), qsort(3C), random(3C), realpath(3C),
strtol(3C), strtol(3C), strtof(3C), unlockpt(3C), wcstombs(3C), wctomb(3C),
limits.h(3HEAD), math.h(3HEAD), stddef.h(3HEAD), types.h(3HEAD),
wait.h(3HEAD), attributes(5), standards(5)
The `<string.h>` header defines the following:

- **NULL** null pointer constant
- **size_t** as described in `<stddef.h>`

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also memory(3C), strcoll(3C), string(3C), strxfrm(3C), stddef.h(3HEAD), types.h(3HEAD), attributes(5), standards(5)
#include <strings.h>

The `size_t` type specified in `<strings.h>` is defined through `typedef` as described in `<stddef.h>`.

Attributes

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also `ffs(3C), string(3C), stddef.h(3HEAD), attributes(5), standards(5)`
Name  stropts.h, stropts – STREAMS interface (STREAMS)

Synopsis  #include <stropts.h>

Description  The <stropts.h> header defines the bandinfo structure, which includes the following members:

    unsigned char bi_pri /* priority band */
    int bi_flag /* flushing type */

The <stropts.h> header defines the strpeek structure that includes the following members:

    struct strbuf ctlbuf /* control portion of the message */
    struct strbuf databuf /* data portion of the message */
    t_uscalar_t flags /* RS_HIPRI or 0 */

The <stropts.h> header defines the strbuf structure that includes the following members:

    int maxlen /* maximum buffer length */
    int len /* length of data */
    char *buf /* pointer to buffer */

The <stropts.h> header defines the strfdinsert structure that includes the following members:

    struct strbuf ctlbuf /* control portion of the message */
    struct strbuf databuf /* data portion of the message */
    t_uscalar_t flags /* RS_HIPRI or 0 */
    int fildes /* file descriptor of the other STREAM */
    int offset /* relative location of the stored value */

The <stropts.h> header defines the strioctrl structure that includes the following members:

    int ic_cmd /* ioctl() command */
    int ic_timeout /* timeout for response */
    int ic_len /* length of data */
    char *ic_dp /* pointer to buffer */

The <stropts.h> header defines the strrecvfd structure that includes the following members:

    int fda /* received file descriptor */
    uid_t uid /* UID of sender */
    gid_t gid /* GID of sender */

The uid_t and gid_t types are defined through typedef as described in <sys/types.h>. See types.h(3HEAD).

The <stropts.h> header defines the t Scalar_t and t Uscalar_t types, respectively, as signed and unsigned opaque types of equal length of at least 32 bits.
The `<stropts.h>` header defines the `str_list` structure that includes the following members:

```c
int sl_nmods /* number of STREAMS module names */
struct str_mlist *sl_modlist /* STREAMS module names */
```

The `<stropts.h>` header defines the `str_mlist` structure that includes the following member:

```c
char l_name[FMNAMESZ+1] /* a STREAMS module name */
```

The following macros are defined for use as the request argument to `ioctl()`:

- `I_PUSH` Push a STREAMS module.
- `I_POP` Pop a STREAMS module.
- `I_LOOK` Get the top module name.
- `I_FLUSH` Flush a STREAM.
- `I_FLUSHBAND` Flush one band of a STREAM.
- `I_SETSIG` Ask for notification signals.
- `I_GETSIG` Retrieve current notification signals.
- `I_FIND` Look for a STREAMS module.
- `I_PEEK` Peek at the top message on a STREAM.
- `I_SRDOPT` Set the read mode.
- `I_GRDOPT` Get the read mode.
- `I_NREAD` Size the top message.
- `I_FDINSERT` Send implementation-defined information about another STREAM.
- `I_STR` Send a STREAMS `ioctl()`.
- `I_SWROPT` Set the write mode.
- `I_GRWROPT` Get the write mode.
- `I_SENDFD` Pass a file descriptor through a STREAMS pipe.
- `I_RECVFD` Get a file descriptor sent via `I_SENDFD`.
- `I_LIST` Get all the module names on a STREAM.
- `I_ATMARK` Is the top message "marked"?
- `I_CKBAND` See if any messages exist in a band.
- `I_GETBAND` Get the band of the top message on a STREAM.
I_CANPUT  Is a band writable?
I_SETCLTIME  Set close time delay.
I_GETCLTIME  Get close time delay.
I_LINK  Connect two STREAMs.
I_UNLINK  Disconnect two STREAMs.
I_PLINK  Persistently connect two STREAMs.
I_PUNLINK  Dismantle a persistent STREAMS link.

The following macro is defined for use with I_LOOK:
FMNAMESZ  minimum size in bytes of the buffer referred to by the arg argument

The following macros are defined for use with I_FLUSH:
FLUSHR  flush read queues
FLUSHW  flush write queues
FLUSHRW  flush read and write queues

The following macros are defined for use with I_SETSIG:
S_RDNORM  A normal (priority band set to 0) message has arrived at the head of a STREAM head read queue.
S_RDBAND  A message with a non-zero priority band has arrived at the head of a STREAM head read queue.
S_INPUT  A message, other than a high-priority message, has arrived at the head of a STREAM head read queue.
S_HIPRI  A high-priority message is present on a STREAM head read queue.
S_OUTPUT  The write queue for normal data (priority band 0) just below the STREAM head is no longer full. This notifies the process that there is room on the queue for sending (or writing) normal data downstream.
S_WRNORM  Equivalent to S_OUTPUT.
S_WRBAND  The write queue for a non-zero priority band just below the STREAM head is no longer full.
S_MSG  A STREAMS signal message that contains the SIGPOLL signal reaches the front of the STREAM head read queue.
S_ERROR  Notification of an error condition reaches the STREAM head.
S_HANGUP  Notification of a hangup reaches the STREAM head.
S_BANDURG  When used in conjunction with S_RDBAND, SIGURG is generated instead of SIGPOLL when a priority message reaches the front of the STREAM head read queue.

The following macro is defined for use with I_PEEK:
RS_HIPRI  Only look for high-priority messages.

The following macros are defined for use with I_SRDOPT:
RNORM  Byte-STREAM mode, the default.
RMSGD  Message-discard mode.
RMSGN  Message-non-discard mode.
RPROTNORM  Fail read() with [EBADMSG] if a message containing a control part is at the front of the STREAM head read queue.
RPROTDAT  Deliver the control part of a message as data when a process issues a read()
RPROTDIS  Discard the control part of a message, delivering any data part, when a process issues a read()

The following macro is defined for use with I_SWOPT:
SNDZERO  Send a zero-length message downstream when a write() of 0 bytes occurs.

The following macros are defined for use with I_ATMARK:
ANYMARK  Check if the message is marked.
LASTMARK  Check if the message is the last one marked on the queue.

The following macro is defined for use with I_UNLINK:
MUXID_ALL  Unlink all STREAMs linked to the STREAM associated with fildes.

The following macros are defined for getmsg(), getpmsg(), putmsg(), and putpmsg():
MSG_ANY  Receive any message.
MSG_BAND  Receive message from specified band.
MSG_HIPRI  Send/receive high-priority message.
MORECTL  More control information is left in message.
MOREDATA  More data is left in message.

The <stropts.h> header can make visible all of the symbols from <unistd.h>.
Attributes  See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also  close(2), fcntl(2), getmsg(2), ioctl(2), open(2), pipe(2), poll(2), putmsg(2), read(2), write(2), signal(3C), types.h(3HEAD), unistd.h(3HEAD), attributes(5), standards(5)
syslog.h—definitions for system error logging

#include <syslog.h>

The <syslog.h> header defines the following symbolic constants, zero or more of which can be OR’ed together to form the logopt option of openlog():

LOG_PID Log the process ID with each message.
LOG_CONS Log to the system console on error.
LOG_NDELAY Connect to syslog daemon immediately.
LOG_ODELAY Delay open until syslog() is called.
LOG_NOWAIT Do not wait for child processes.

The following symbolic constants are defined as possible values of the facility argument to openlog():

LOG_KERN reserved for message generated by the system
LOG_USER message generated by a process
LOG_MAIL reserved for message generated by mail system
LOG_NWS reserved for message generated by news system
LOG_UUCP reserved for message generated by UUCP system
LOG_DAEMON reserved for message generated by system daemon
LOG_AUTH reserved for message generated by authorization daemon
LOG_CRON reserved for message generated by clock daemon
LOG_LPR reserved for message generated by printer system
LOG_LOCAL0 reserved for local use
LOG_LOCAL1 reserved for local use
LOG_LOCAL2 reserved for local use
LOG_LOCAL3 reserved for local use
LOG_LOCAL4 reserved for local use
LOG_LOCAL5 reserved for local use
LOG_LOCAL6 reserved for local use
LOG_LOCAL7 reserved for local use
The following is declared as a macro for constructing the `maskpri` argument to `setlogmask()`. The following macro expands to an expression of type `int` when the argument `pri` is an expression of type `int`:

```c
LOG_MASK(pri)  a mask for priority `pri`
```

The following constants are defined as possible values for the `priority` argument of `syslog()`:

- `LOG_EMERG`: A panic condition was reported to all processes.
- `LOG_ALERT`: A condition that should be corrected immediately.
- `LOG_CRIT`: A critical condition.
- `LOG_ERR`: An error message.
- `LOG_WARNING`: A warning message.
- `LOG_NOTICE`: A condition requiring special handling.
- `LOG_INFO`: A general information message.
- `LOG_DEBUG`: A message useful for debugging programs.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also** `syslog(3C), attributes(5), standards(5)`
Name: tar.h, tar – extended tar definitions

Synopsis: #include <tar.h>

Description: The `<tar.h>` header defines header block definitions as follows.

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TMAGIC</td>
<td>&quot;ustar&quot;</td>
<td>ustar plus null byte</td>
</tr>
<tr>
<td>TMAGLEN</td>
<td>6</td>
<td>length of the above</td>
</tr>
<tr>
<td>TVERSION</td>
<td>&quot;00&quot;</td>
<td>00 without a null byte</td>
</tr>
<tr>
<td>TVERSLEN</td>
<td>2</td>
<td>length of the above</td>
</tr>
</tbody>
</table>

Type flag field definitions:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGTYPE</td>
<td>'0'</td>
<td>regular file</td>
</tr>
<tr>
<td>AREGTYPE</td>
<td>'\0'</td>
<td>regular file</td>
</tr>
<tr>
<td>LNKTYPE</td>
<td>'1'</td>
<td>link</td>
</tr>
<tr>
<td>SYMTYPE</td>
<td>'2'</td>
<td>symbolic link</td>
</tr>
<tr>
<td>CHRTYPE</td>
<td>'3'</td>
<td>character special</td>
</tr>
<tr>
<td>BLKTYPE</td>
<td>'4'</td>
<td>block special</td>
</tr>
<tr>
<td>DIRTYPE</td>
<td>'5'</td>
<td>directory</td>
</tr>
<tr>
<td>FIFOTYPE</td>
<td>'6'</td>
<td>FIFO special</td>
</tr>
<tr>
<td>CONTTYPE</td>
<td>'7'</td>
<td>reserved</td>
</tr>
</tbody>
</table>

Mode field bit definitions (octal):

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSUID</td>
<td>04000</td>
<td>set UID on execution</td>
</tr>
<tr>
<td>TSGID</td>
<td>02000</td>
<td>set GID on execution</td>
</tr>
<tr>
<td>TSVTX</td>
<td>01000</td>
<td>on directories, restricted deletion flag</td>
</tr>
<tr>
<td>Name</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>------------------------------</td>
</tr>
<tr>
<td>TUREAD</td>
<td>00400</td>
<td>read by owner</td>
</tr>
<tr>
<td>TUWRITE</td>
<td>00200</td>
<td>write by owner special</td>
</tr>
<tr>
<td>TUEXEC</td>
<td>00100</td>
<td>execute/search by owner</td>
</tr>
<tr>
<td>TGREAD</td>
<td>00040</td>
<td>read by group</td>
</tr>
<tr>
<td>TGWRITE</td>
<td>00020</td>
<td>write by group</td>
</tr>
<tr>
<td>TGEXEC</td>
<td>00010</td>
<td>execute/search by group</td>
</tr>
<tr>
<td>TOREAD</td>
<td>00004</td>
<td>read by other</td>
</tr>
<tr>
<td>TOWRITE</td>
<td>00002</td>
<td>write by other</td>
</tr>
<tr>
<td>TOEXEC</td>
<td>00001</td>
<td>execute/search by other</td>
</tr>
</tbody>
</table>

Types used in ancillary files:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACL_HDR</td>
<td>'A'</td>
<td>Access Control List</td>
</tr>
<tr>
<td>LBL_TYPE</td>
<td>'L'</td>
<td>Trusted Extensions file label</td>
</tr>
<tr>
<td>DIR_TYPE</td>
<td>'D'</td>
<td>Trusted Extensions directory label</td>
</tr>
</tbody>
</table>

Attribute types used in Trusted Solaris ancillary files that are interpreted by Trusted Extensions for backward compatibility:

<table>
<thead>
<tr>
<th>Name</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLD_TYPE</td>
<td>'S'</td>
<td>Single-level directory component</td>
</tr>
<tr>
<td>PATH_TYPE</td>
<td>'P'</td>
<td>Path component</td>
</tr>
<tr>
<td>MLD_TYPE</td>
<td>'M'</td>
<td>Multi-level directory component</td>
</tr>
<tr>
<td>FILE_TYPE</td>
<td>'F'</td>
<td>Must handle files differently</td>
</tr>
<tr>
<td>APRIV_TYPE</td>
<td>'P'</td>
<td>Allowed privileges data type in file</td>
</tr>
<tr>
<td>FPRIV_TYPE</td>
<td>'p'</td>
<td>Forced privileges data type in file</td>
</tr>
<tr>
<td>COMP_TYPE</td>
<td>'C'</td>
<td>Path components, use for MLD</td>
</tr>
<tr>
<td>ATTR_FLAG_TYPE</td>
<td>'F'</td>
<td>File attribute flag bytes data type</td>
</tr>
<tr>
<td>LK_COMP_TYPE</td>
<td>'K'</td>
<td>Link data path component</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>See below.</td>
</tr>
</tbody>
</table>

The general definitions, the type flag field definitions, and the mode field bit definitions are Standard. The types used in ancillary files and the attribute types used in Trusted Solaris ancillary files are Evolving.

See Also  pax(1), attributes(5), standards(5)
Name tcp.h, tcp – definitions for the Internet Transmission Control Protocol (TCP)

Synopsis

```
#include <netinet/tcp.h>
```

Description

The <netinet/tcp.h> header defines the following macro for use as a socket option at the IPPROTO_TCP level:

```
TCP_NODELAY  Avoid coalescing of small segments.
```

The macro is defined in the header. The implementation need not allow the value of the option to be set with `setsockopt()` or retrieved with `getsockopt()`.

Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also

`getsockopt(3XNET), socket.h(3HEAD), attributes(5), standards(5)`
The `<termios.h>` header contains the definitions used by the terminal I/O interfaces. See `termios(3C)` and `termio(7I)` for an overview of the terminal interface.

The following data types are defined through `typedef`:

- `cc_t` used for terminal special characters
- `speed_t` used for terminal baud rates
- `tcflag_t` used for terminal modes

The above types are all unsigned integer types.

The implementation supports one or more programming environments in which the widths of `cc_t`, `speed_t`, and `tcflag_t` are no greater than the width of type `long`. The names of these programming environments can be obtained using the `confstr(3C)` function or the `getconf(1)` utility.

The `termios` structure is defined and includes the following members:

- `tcflag_t c_iflag /* input modes */`
- `tcflag_t c_oflag /* output modes */`
- `tcflag_t c_cflag /* control modes */`
- `tcflag_t c_lflag /* local modes */`
- `cc_t c_cc[NCCS] /* control characters */`

A definition is provided for:

- `NCCS` size of the array `c_cc` for control characters

The following subscript names for the array `c_cc` are defined:

<table>
<thead>
<tr>
<th>Subscript Usage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Canonical Mode</td>
<td>Non-Canonical Mode</td>
</tr>
<tr>
<td>VEOF</td>
<td>EOF character</td>
</tr>
<tr>
<td>VEOL</td>
<td>EOL character</td>
</tr>
<tr>
<td>VERASE</td>
<td>ERASE character</td>
</tr>
<tr>
<td>VINTR</td>
<td>INTR character</td>
</tr>
<tr>
<td>VKILL</td>
<td>KILL character</td>
</tr>
<tr>
<td></td>
<td>VMIN</td>
</tr>
</tbody>
</table>
The subscript values are unique, except that the `VMIN` and `VTIME` subscripts can have the same values as the `VEOF` and `VEOL` subscripts, respectively.

The header file provides the flags described below.

**Input Modes**

The `c_iflag` field describes the basic terminal input control:

- **BRKINT** Signal interrupt on break.
- **ICRNL** Map CR to NL on input.
- **IGNBRK** Ignore break condition.
- **IGNCR** Ignore CR.
- **IGNPAR** Ignore characters with parity errors.
- **INLCR** Map NL to CR on input.
- **INPCK** Enable input parity check.
- **ISTRIP** Strip character.
- **IXANY** Enable any character to restart output.
- **IXOFF** Enable start/stop input control.
- **IXON** Enable start/stop output control.
- **PARMRK** Mark parity errors.

**Output Modes**

The `c_oflag` field specifies the system treatment of output:

- **OPOST** Post-process output.
- **ONLCR** Map NL to CR-NL on output.
- **OCRNL** Map CR to NL on output.
- **ONOCR** No CR output at column 0.
- **ONLRET** NL performs CR function.
The input and output baud rates are stored in the `termios` structure. These are the valid values for objects of type `speed_t`. The following values are defined, but not all baud rates need be supported by the underlying hardware.

- **B0** Hang up
- **B50** 50 baud
- **B75** 75 baud
- **B110** 110 baud
### Control Modes

The `c_cflag` field describes the hardware control of the terminal; not all values specified are required to be supported by the underlying hardware:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSIZE</td>
<td>Character size:</td>
</tr>
<tr>
<td>CS5</td>
<td>5 bits</td>
</tr>
<tr>
<td>CS6</td>
<td>6 bits</td>
</tr>
<tr>
<td>CS7</td>
<td>7 bits</td>
</tr>
<tr>
<td>CS8</td>
<td>8 bits</td>
</tr>
<tr>
<td>CSTOPB</td>
<td>Send two stop bits, else one.</td>
</tr>
<tr>
<td>CREAD</td>
<td>Enable receiver.</td>
</tr>
<tr>
<td>PARENB</td>
<td>Parity enable.</td>
</tr>
<tr>
<td>PARODD</td>
<td>Odd parity, else even.</td>
</tr>
<tr>
<td>HUPCL</td>
<td>Hang up on last close.</td>
</tr>
<tr>
<td>CLOCAL</td>
<td>Ignore modem status lines.</td>
</tr>
</tbody>
</table>

The implementation supports the functionality associated with the symbols CS7, CS8, CSTOPB, PARODD, and PARENB.

### Local Modes

The `c_lflag` field of the argument structure is used to control various terminal functions:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECHO</td>
<td>Enable echo.</td>
</tr>
</tbody>
</table>
ECHOE  Echo erase character as error-correcting backspace.
ECHOK  Echo KILL.
ECHONL Echo NL.
ICANON Canonical input (erase and kill processing).
IEXTEN Enable extended input character processing.
ISIG  Enable signals.
NOFLSH Disable flush after interrupt or quit.
TOSTOP Send SIGTTOU for background output.

The following symbolic constants for use with `tcgetattr()` are defined:

<table>
<thead>
<tr>
<th>Symbolic Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCSANOW</td>
<td>Change attributes immediately.</td>
</tr>
<tr>
<td>TCSADRAIN</td>
<td>Change attributes when output has drained.</td>
</tr>
<tr>
<td>TCSAFLUSH</td>
<td>Change attributes when output has drained; also flush pending input.</td>
</tr>
</tbody>
</table>

The following symbolic constants for use with `tcflush()` are defined:

<table>
<thead>
<tr>
<th>Symbolic Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCIFLUSH</td>
<td>Flush pending input.</td>
</tr>
<tr>
<td>TIOFLUSH</td>
<td>Flush both pending input and untransmitted output.</td>
</tr>
<tr>
<td>TCOFLUSH</td>
<td>Flush untransmitted output.</td>
</tr>
</tbody>
</table>

The following symbolic constants for use with `tcflow()` are defined:

<table>
<thead>
<tr>
<th>Symbolic Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCIOFF</td>
<td>Transmit a STOP character, intended to suspend input data.</td>
</tr>
<tr>
<td>TCION</td>
<td>Transmit a START character, intended to restart input data.</td>
</tr>
<tr>
<td>TCOFF</td>
<td>Suspend output.</td>
</tr>
<tr>
<td>TCOON</td>
<td>Restart output.</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Attribute Type</th>
<th>Attribute Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**  getconf(1), cfgetispeed(3C), cfsetispeed(3C), confstr(3C), tcdrain(3C), tcflow(3C), tcflush(3C), tcgetattr(3C), tcgetattrs(3C), tcgetsid(3C), tcsetbreak(3C), tcsetattr(3C), attributes(5), standards(5)
The `<tgmath.h>` header includes the headers `<math.h>` and `<complex.h>` and defines several type-generic macros.

Of the functions contained within the `<math.h>` and `<complex.h>` headers without an f (float) or l (long double) suffix, several have one or more parameters whose corresponding real type is double. For each such function except `modf(3M)`, there is a corresponding type-generic macro. The parameters whose corresponding real type is double in the function synopsis are generic parameters. Use of the macro invokes a function whose corresponding real type and type domain are determined by the arguments for the generic parameters.

Use of the macro invokes a function whose generic parameters have the corresponding real type determined as follows:

- First, if any argument for generic parameters has type `long double`, the type determined is `long double`.
- Otherwise, if any argument for generic parameters has type `double` or is of integer type, the type determined is `double`.
- Otherwise, the type determined is `float`.

For each unsuffixed function in the `<math.h>` header for which there is a function in the `<complex.h>` header with the same name except for a c prefix, the corresponding type-generic macro (for both functions) has the same name as the function in the `<math.h>` header. The corresponding type-generic macro for `fabs()` and `cabs()` is `fabs()`.

<table>
<thead>
<tr>
<th><code>&lt;math.h&gt;</code> Function</th>
<th><code>&lt;complex.h&gt;</code> Function</th>
<th>Type-Generic Macro</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>acos()</code></td>
<td><code>cacos()</code></td>
<td><code>acos()</code></td>
</tr>
<tr>
<td><code>asin()</code></td>
<td><code>casin()</code></td>
<td><code>asin()</code></td>
</tr>
<tr>
<td><code>atan()</code></td>
<td><code>catan()</code></td>
<td><code>atan()</code></td>
</tr>
<tr>
<td><code>acosh()</code></td>
<td><code>cacosh()</code></td>
<td><code>acosh()</code></td>
</tr>
<tr>
<td><code>asinh()</code></td>
<td><code>casinh()</code></td>
<td><code>asinh()</code></td>
</tr>
<tr>
<td><code>atanh()</code></td>
<td><code>catanh()</code></td>
<td><code>atanh()</code></td>
</tr>
<tr>
<td><code>cos()</code></td>
<td><code>ccos()</code></td>
<td><code>cos()</code></td>
</tr>
<tr>
<td><code>sin()</code></td>
<td><code>csin()</code></td>
<td><code>sin()</code></td>
</tr>
<tr>
<td><code>tan()</code></td>
<td><code>ctan()</code></td>
<td><code>tan()</code></td>
</tr>
<tr>
<td><code>cosh()</code></td>
<td><code>ccosh()</code></td>
<td><code>cosh()</code></td>
</tr>
<tr>
<td><code>&lt;math.h&gt;</code> Function</td>
<td><code>&lt;complex.h&gt;</code> Function</td>
<td>Type-Generic Macro</td>
</tr>
<tr>
<td>---------------------</td>
<td>------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>sinh()</td>
<td>csinh()</td>
<td>sinh()</td>
</tr>
<tr>
<td>tanh()</td>
<td>ctanh()</td>
<td>tanh()</td>
</tr>
<tr>
<td>exp()</td>
<td>cexp()</td>
<td>exp()</td>
</tr>
<tr>
<td>log()</td>
<td>clog()</td>
<td>log()</td>
</tr>
<tr>
<td>pow()</td>
<td>cpow()</td>
<td>pow()</td>
</tr>
<tr>
<td>sqrt()</td>
<td>csqrt()</td>
<td>sqrt()</td>
</tr>
<tr>
<td>fabs()</td>
<td>cfabs()</td>
<td>fabs()</td>
</tr>
</tbody>
</table>

If at least one argument for a generic parameter is complex, then use of the macro invokes a complex function; otherwise, use of the macro invokes a real function.

For each unsuffixed function in the `<math.h>` header without a c-prefixed counterpart in the `<complex.h>` header, the corresponding type-generic macro has the same name as the function. These type-generic macros are:

- atan2()
- fma()
- llround()
- remainder()
- cbrt()
- fmax()
- log10()
- remquo()
- ceil()
- fmin()
- log1p()
- rint()
- copysign()
- fmod()
- log2()
- round()
- erf()
- frexp()
- logb()
- scalbn()
- erfc()
- hypot()
- lrint()
- scalbln()
- exp2()
- ilogb()
- lround()
- tgamma()
- expm1()
- ldexp()
- nearbyint()
- trunc()
- fdim()
- lgamma()
- nextafter()
- floor()
- lldiv()
- nexttoward()

If all arguments for generic parameters are real, then use of the macro invokes a real function; otherwise, use of the macro results in undefined behavior.

For each unsuffixed function in the `<complex.h>` header that is not a c-prefixed counterpart to a function in the `<math.h>` header, the corresponding type-generic macro has the same name as the function. These type-generic macros are:

- carg()
- cimag()
- conj()
- cproj()
- creal()

Use of the macro with any real or complex argument invokes a complex function.
Functions invoked by use of type-generic macros are invoked with the declarations listed below.

```c
#include <tgmath.h>
int n;
float f;
double d;
long double ld;
float complex fc;
double complex dc;
long double complex ldc;
```

The following are the type-generic macros that invoke the functions that are invoked with the preceding declarations.

<table>
<thead>
<tr>
<th>Macro</th>
<th>Use Invokes</th>
</tr>
</thead>
<tbody>
<tr>
<td>exp(n)</td>
<td>exp(n), the function</td>
</tr>
<tr>
<td>acosh(f)</td>
<td>acoshf(f)</td>
</tr>
<tr>
<td>sin(d)</td>
<td>sin(d), the function</td>
</tr>
<tr>
<td>atan(ld)</td>
<td>atanl(ld)</td>
</tr>
<tr>
<td>log(fc)</td>
<td>clogf(fc)</td>
</tr>
<tr>
<td>sqrt(dc)</td>
<td>csqrt(dc)</td>
</tr>
<tr>
<td>pow(ldc,f)</td>
<td>cpowl(ldc, f)</td>
</tr>
<tr>
<td>remainder(n,n)</td>
<td>remainder(n, n), the function</td>
</tr>
<tr>
<td>nextafter(d,f)</td>
<td>nextafter(d, f), the function</td>
</tr>
<tr>
<td>nexttoward(f,ld)</td>
<td>nexttoward(f, ld)</td>
</tr>
<tr>
<td>copysign(n,ld)</td>
<td>copysignl(n, ld)</td>
</tr>
<tr>
<td>ceil(fc)</td>
<td>undefined behavior</td>
</tr>
<tr>
<td>rint(dc)</td>
<td>undefined behavior</td>
</tr>
<tr>
<td>fmax(ldc,ld)</td>
<td>undefined behavior</td>
</tr>
<tr>
<td>carg(n)</td>
<td>carg(n), the function</td>
</tr>
<tr>
<td>cproj(f)</td>
<td>cprojf(f)</td>
</tr>
<tr>
<td>creal(d)</td>
<td>creald(d), the function</td>
</tr>
<tr>
<td>cimag(ld)</td>
<td>cimagl(ld)</td>
</tr>
<tr>
<td>cabs(fc)</td>
<td>cabsf(fc)</td>
</tr>
</tbody>
</table>
Macro Use Invokes
---
carg(dc) carg(dc), the function
cproj(ldc) cproj(ldc)

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also modf(3M), complex.h(3HEAD), math.h(3HEAD), cabs(3M), fabs(3M), attributes(5), standards(5)
timeb.h

Name timeb.h, timeb – additional definitions for date and time

Synopsis #include <sys/timeb.h>

Description The <sys/timeb.h> header defines the timeb structure, which includes the following members:

    time_t time /* the seconds portion of the current time */
    unsigned short millitm /* the milliseconds portion of the current time */
    short timezone /* the local timezone in minutes west of Greenwich */
    short dstflag /* TRUE if Daylight Savings Time is in effect */

The time_t type is defined as described in <sys/types.h>.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also time.h(3HEAD), types.h(3HEAD), attributes(5), standards(5)
The `<time.h>` header declares the structure `tm`, which includes the following members:

```c
int tm_sec /* seconds [0,60] */
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 */
```

The value of `tm_isdst` 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.

The `<time.h>` header defines the following symbolic names:

- **NULL**: Null pointer constant.
- **CLOCKS_PER_SEC**: A number used to convert the value returned by the `clock()` function into seconds. See `clock(3C)`.
- **CLOCK_PROCESS_CPUTIME_ID**: The identifier of the CPU-time clock associated with the process making a `clock()` or `timer*()` function call.
- **CLOCK_THREAD_CPUTIME_ID**: The identifier of the CPU-time clock associated with the thread making a `clock()` or `timer*()` function call.

The `<time.h>` header declares the `timespec` structure, which has the following members:

```c
time_t tv_sec /* seconds */
long tv_nsec /* nanoseconds */
```

The `<time.h>` header declares the `itimerspec` structure, which has the following members:

```c
struct timespec it_interval /* timer period */
struct timespec it_value /* timer expiration */
```

The following manifest constants are defined:

- **CLOCK_REALTIME**: The identifier of the system-wide realtime clock.
- **TIMER_ABSTIME**: Flag indicating time is absolute. For functions taking timer objects, this refers to the clock associated with the timer.
- **CLOCK_MONOTONIC**: The identifier for the system-wide monotonic clock, which is defined as a clock whose value cannot be set with `clock_settime()` and that cannot have backward clock jumps. The maximum possible clock jump
is implementation-defined. See `clock_gettime(3RT)`.

The `clock_t, size_t, time_t, clockid_t, and timer_t` types are defined as described in `<sys/types.h>`. See `types.h(3HEAD)`.

Although the value of `CLOCKS_PER_SEC` is required to be 1 million on all standard-conforming systems, it can be variable on other systems, and it should not be assumed that `CLOCKS_PER_SEC` is a compile-time constant.

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

The following are declared as variables:

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

Inclusion of the `<time.h>` header can make visible all symbols from the `<signal.h>` header.

**Usage**

The range [0,60] for `tm_sec` allows for the occasional leap second.

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

To obtain the number of clock ticks per second returned by the `times()` function, applications should call `sysconf(_SC_CLK_TCK)`. See `times(2)` and `sysconf(3C)`.

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**

`time(2), utime(2), clock(3C), ctime(3C), difftime(3C), getdate(3C), mktime(3C), strftime(3C), strptime(3C), types.h(3HEAD), clock_settime(3RT), nanosleep(3RT), timer_create(3RT), timer_delete(3RT), timer_settime(3RT), attributes(5), standards(5)`
Name times.h, times – file access and modification times structure

Synopsis #include <sys/times.h>

Description The <sys/times.h> header defines the structure tms, which is returned by times() and includes the following members:

- `clock_t tms_utime /* user CPU time */`
- `clock_t tms_stime /* system CPU time */`
- `clock_t tms_cutime /* user CPU time of terminated child processes */`
- `clock_t tms_cstime /* system CPU time of terminated child processes */`

The `clock_t` type is defined as described in <sys/types.h>.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also times(2), types.h(3HEAD), attributes(5), standards(5)
The following fixed-width data types defined in `<sys/types32.h>` 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.

```c
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 nlink32_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
```
The data types defined in `<sys/types.h>` are as follows:

### 32-bit Solaris

```c
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;
```

### 64-bit Solaris

```c
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 ptdiff_t intptr_t
typedef ulong_t rlim_t
typedef ulong_t size_t
typedef uint_t speed_t
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

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 be compiled only 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 be compiled only as part of a 64-bit version of a given C program. See EXAMPLES.

This header incorporates definitions of other preprocessor symbols that can be useful when keeping code portable between different instruction set architectures.

__LITTLE_ENDIAN _BIG_ENDIAN
The natural byte order of the processor. A pointer to an int points to the least/most significant byte of that int.

__STACK_GROWS_UPWARD _STACK_GROWS_DOWNWARD
The processor specific direction of stack growth. A push onto the stack increases/decreases the stack pointer, so it stores data at successively higher/lower addresses.

__CHAR_IS_UNSIGNED _CHAR_IS_SIGNED
The C Compiler implements objects of type char as unsigned or signed respectively. This is really an implementation choice of the compiler, but it is specified in the ABI and tends to be uniform across compilers for an instruction set architecture.
The ABI defines alignment requirements of each of the primitive object types. Some, if not all, might be hardware requirements as well. The values are expressed in bytes.

The most stringent alignment requirement as specified by the ABI. Equal to the maximum of all the above _XXX_ALIGNMENT values.

The 32-bit ABI supported by a 64-bit kernel may have different alignment requirements for primitive object types. The value of this identifier is expressed in bytes.

The addr_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.

In the following example, the preprocessor symbol _LP64 defines sections of code that will be compiled only as part of a 64-bit version of the given C program.

```c
#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
```

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Stable</td>
</tr>
</tbody>
</table>
See Also  types32.h(3HEAD), attributes(5), standards(5)
#include <ucontext.h>

The `ucontext.h` header defines the `ucontext_t` type as a structure that includes at least the following members:

- `ucontext_t uc_link`
- `sigset_t uc_sigmask`
- `stack_t uc_stack`
- `mcontext_t uc_mcontext`

The `uc_link` member is a pointer to the context that to be resumed when this context returns. If `uc_link` is equal to 0, this context is the main context and the process exits when this context returns.

The `uc_sigmask` member defines the set of signals that are blocked when this context is active. See `sigprocmask(2)`.

The `uc_stack` member defines the stack used by this context. See `sigaltstack(2)`.

The `uc_mcontext` member contains the saved set of machine registers and any implementation-specific context data. Portable applications should not modify or access `uc_mcontext`.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

## See Also
`getcontext(2), sigaction(2), sigaltstack(2), sigprocmask(2), makecontext(3C), attributes(5), standards(5)`
#include <sys/uio.h>

The <sys/uio.h> header defines the iovec structure, which includes the following members:

```c
void  *iov_base    /* base address of a memory region
            for input or output */
size_t  iov_len    /* size of the memory pointed to by
                    iov_base */
```

The <sys/uio.h> header uses the iovec structure for scatter/gather I/O.

The ssize_t and size_t types are defined as described in <sys/types.h>.

The symbol {IOV_MAX} defined in <limits.h> should always be used to learn about the limits on the number of scatter/gather elements that can be processed in one call, instead of assuming a fixed value.

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**

read(2), write(2), limits.h(3HEAD), types.h(3HEAD), attributes(5), standards(5)
ulimit.h, ulimit — ulimit commands

Synopsis

```
#include <ulimit.h>
```

Description

The `<ulimit.h>` header defines the following symbolic constants used by the `ulimit()` function.

- **UL_GETFSIZE**
  - Get maximum file size.
- **UL_SETFSIZE**
  - Set maximum file size.

Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also

`ulimit(2), attributes(5), standards(5)`
**Name**  un.h, un – definitions for UNIX-domain sockets

**Synopsis**  #include <sys/un.h>

**Description**  The <sys/un.h> header defines the sockaddr_un structure that includes the following members:

```c
sa_family_t sun_family  /* address family */
char sun_path[]  /* socket pathname */
```

The sockaddr_un structure is used to store addresses for UNIX domain sockets. Values of this type must be cast to struct sockaddr for use with the socket interfaces.

The <sys/un.h> header defines the type sa_family_t as described in socket.h(3HEAD).

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**  bind(3SOCKET), bind(3XNET), socket.h(3HEAD), socket(3SOCKET), socket(3XNET), socketpair(3SOCKET), socketpair(3XNET), attributes(5), standards(5)
#include <unistd.h>

The `<unistd.h>` 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.

The following symbolic constants are defined (with fixed values):

- `_POSIX_VERSION` Integer value indicating version of the POSIX standard (C language binding). See `standards(5)`.
- `_POSIX2_VERSION` Integer value indicating version of the POSIX.2 standard (Commands).
- `_POSIX2_C_VERSION` Integer value indicating version of the POSIX.2 standard (C language binding).
- `_XOPEN_VERSION` Integer value indicating version of the XPG to which system conforms.
- `_XOPEN_XCU_VERSION` Integer value indicating the version of the XCU specification to which the implementation conforms. If this constant is not defined, use the `sysconf(3C)` function to determine which features are supported. This constant is not defined for the SUSv3 environment.

The following symbolic constants, if defined in `<unistd.h>`, have a value of -1, 0, or greater, unless otherwise specified below. If these are undefined, the `fpathconf(2)`, `pathconf(2)`, or `sysconf(3C)` functions can be used to determine whether the option is provided for a particular invocation of the application.

If a symbolic constant is defined with the value -1, the option is not supported. Headers, data types, and function interfaces required only for the option need not be supplied. An application that attempts to use anything associated only with the option is considered to be requiring an extension.

If a symbolic constant is defined with a value greater than zero, the option is always supported when the application is executed. All headers, data types, and functions are present and operate as specified.

If a symbolic constant is defined with the value zero, all headers, data types, and functions are present. The application can check at runtime to see whether the option is supported by calling `fpathconf()`, `pathconf()`, or `sysconf()` with the indicated `name` parameter.

Unless explicitly specified otherwise, the behavior of functions associated with an unsupported option is unspecified, and an application that uses such functions without first checking `fpathconf()`, `pathconf()`, or `sysconf()` is considered to be requiring an extension.
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_POSIX_ADVISORY_INFO</td>
<td>Implementation supports the Advisory Information option.</td>
</tr>
<tr>
<td>_POSIX_ASYNCIONIOUS_IO</td>
<td>Implementation supports the Asynchronous Input and Output option.</td>
</tr>
<tr>
<td>_POSIX_BARRIERS</td>
<td>Implementation supports the Barriers option.</td>
</tr>
<tr>
<td>_POSIX_CLOCK_SELECTION</td>
<td>Implementation supports the Clock Selection option.</td>
</tr>
<tr>
<td>_POSIX_CPU_TIME</td>
<td>Implementation supports the Process CPU-Time Clocks option.</td>
</tr>
<tr>
<td>_POSIX_FSYNC</td>
<td>Implementation supports the File Synchronisation option.</td>
</tr>
<tr>
<td>_POSIX_IPV6</td>
<td>Implementation supports the IPv6 option.</td>
</tr>
<tr>
<td>_POSIX_JOB_CONTROL</td>
<td>Implementation supports job control.</td>
</tr>
<tr>
<td>_POSIX_MAPPED_FILES</td>
<td>Implementation supports the Memory Mapped Files option.</td>
</tr>
<tr>
<td>_POSIX_MEMLOCK</td>
<td>Implementation supports the Process Memory Locking option.</td>
</tr>
<tr>
<td>_POSIX_MEMLOCK_RANGE</td>
<td>Implementation supports the Range Memory Locking option.</td>
</tr>
<tr>
<td>_POSIX_MEMORY_PROTECTION</td>
<td>Implementation supports the Memory Protection option.</td>
</tr>
<tr>
<td>_POSIX_MESSAGE_PASSING</td>
<td>Implementation supports the Message Passing option.</td>
</tr>
<tr>
<td>_POSIX_MONOTONIC_CLOCK</td>
<td>Implementation supports the Monotonic Clock option.</td>
</tr>
<tr>
<td>_POSIX_PRIORITY_SCHEDULING</td>
<td>Implementation supports the Process Scheduling option.</td>
</tr>
<tr>
<td>_POSIX_RAW_SOCKETS</td>
<td>Implementation supports the Raw Sockets option.</td>
</tr>
<tr>
<td>_POSIX_READER_WRITER_LOCKS</td>
<td>Implementation supports the Read-Write Locks option.</td>
</tr>
<tr>
<td>_POSIX_REALTIME_SIGNALS</td>
<td>Implementation supports the Realtime Signals Extension option.</td>
</tr>
<tr>
<td>_POSIX_REGEXP</td>
<td>Implementation supports the Regular Expression Handling option.</td>
</tr>
<tr>
<td>_POSIX_SAVED_IDS</td>
<td>The exec functions (see exec(2)) save the effective user and group.</td>
</tr>
<tr>
<td>Symbol</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>_POSIX_SEMAPHORES</td>
<td>Implementation supports the Semaphores option.</td>
</tr>
<tr>
<td>_POSIX_SHARED_MEMORY_OBJECTS</td>
<td>Implementation supports the Shared Memory Objects option.</td>
</tr>
<tr>
<td>_POSIX_SHELL</td>
<td>Implementation supports the POSIX shell.</td>
</tr>
<tr>
<td>_POSIX_SPAWN</td>
<td>Implementation supports the Spawn option.</td>
</tr>
<tr>
<td>_POSIX_SPIN_LOCKS</td>
<td>Implementation supports the Spin Locks option.</td>
</tr>
<tr>
<td>_POSIX_SPORADIC_SERVER</td>
<td>Implementation supports the Process Sporadic Server option.</td>
</tr>
<tr>
<td>_POSIX_SYNCHRONIZED_IO</td>
<td>Implementation supports the Synchronized Input and Output option.</td>
</tr>
<tr>
<td>_POSIX_THREAD_ATTR_STACKADDR</td>
<td>Implementation supports the thread stack address attribute option.</td>
</tr>
<tr>
<td>_POSIX_THREAD_ATTR_STACKSIZE</td>
<td>Implementation supports the thread stack size attribute option.</td>
</tr>
<tr>
<td>_POSIX_THREAD_CPUTIME</td>
<td>Implementation supports the Thread CPU-Time Clocks option.</td>
</tr>
<tr>
<td>_POSIX_THREAD_PROCESS_SHARED</td>
<td>Implementation supports the process-shared synchronization option.</td>
</tr>
<tr>
<td>_POSIX_THREAD_SAFE_FUNCTIONS</td>
<td>Implementation supports the thread-safe functions option.</td>
</tr>
<tr>
<td>_POSIX_THREAD_SPORADIC_SERVER</td>
<td>Implementation supports the Thread Sporadic Server option.</td>
</tr>
<tr>
<td>_POSIX_THREADS</td>
<td>Implementation supports the threads option.</td>
</tr>
<tr>
<td>_POSIX_TIMERS</td>
<td>Implementation supports the Timers option.</td>
</tr>
<tr>
<td>_POSIX_TIMEOUTS</td>
<td>Implementation supports the Timeouts option.</td>
</tr>
<tr>
<td>_POSIX_TRACE</td>
<td>Implementation supports the Trace option.</td>
</tr>
<tr>
<td>_POSIX_TRACE_EVENT_FILTER</td>
<td>Implementation supports the Trace Event Filter option.</td>
</tr>
<tr>
<td>_POSIX_TRACE_INHERIT</td>
<td>Implementation supports the Trace Inherit option.</td>
</tr>
<tr>
<td>_POSIX_TRACE_LOG</td>
<td>Implementation supports the Trace Log option.</td>
</tr>
<tr>
<td>_POSIX_TYPED_MEMORY_OBJECTS</td>
<td>Implementation supports the Typed Memory Objects option.</td>
</tr>
</tbody>
</table>
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.

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.

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

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.

Implementation supports the XSI STREAMS Option Group.

Implementation supports the C Language Binding option.

Implementation supports the C Language Development Utilities option.

Implementation supports at least one terminal type.

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

Implementation supports the Batch Environment Services and Utilities option.

Implementation supports the Batch Accounting option.

Implementation supports the Batch Checkpoint/Restart option.

Implementation supports the Locate Batch Job Request option.

Implementation supports the Batch Job Message Request option.

Implementation supports the Track Batch Job Request option.

Implementation supports the Software Development Utilities option.
<table>
<thead>
<tr>
<th>Symbolic Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>_POSIX2_UPE</code></td>
<td>Implementation supports the User Portability Utilities option.</td>
</tr>
<tr>
<td><code>_XBS5_ILP32_OFF32</code></td>
<td>Implementation provides a C-language compilation environment with 32-bit int, long, pointer and off_t types.</td>
</tr>
<tr>
<td><code>_XBS5_ILP32_OFFBIG</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>_XBS5_LP64_OFF64</code></td>
<td>Implementation provides a C-language compilation environment with 32-bit int and 64-bit long, pointer and off_t types.</td>
</tr>
<tr>
<td><code>_XBS5_LPBIG_OFFBIG</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>_XOPEN_ENH_I18N</code></td>
<td>Implementation supports the Issue 4, Version 2 Enhanced Internationalization Feature Group.</td>
</tr>
<tr>
<td><code>_XOPEN_LEGACY</code></td>
<td>Implementation supports the Legacy Feature Group.</td>
</tr>
<tr>
<td><code>_XOPEN_REALTIME</code></td>
<td>Implementation supports the X/Open Realtime Feature Group.</td>
</tr>
<tr>
<td><code>_XOPEN_SHM</code></td>
<td>Implementation supports the Issue 4, Version 2 Shared Memory Feature Group.</td>
</tr>
</tbody>
</table>

If any of the following constants are not defined in the header `<unistd.h>`, the value varies depending on the file to which it is applied.
If any of the following constants are defined to have value −1 in the header <unistd.h>, the implementation will not provide the option on any file; if any are defined to have a value other than −1 in the header <unistd.h>, the implementation will provide the option on all applicable files.

All of the following constants, whether defined in <unistd.h> or not, can be queried with respect to a specific file using the pathconf() or fpathconf() functions.

_POSIX_ASYNC_IO  Asynchronous input or output operations can be performed for the associated file.

_POSIX_PRIO_IO  Prioritized input or output operations can be performed for the associated file.

_POSIX_SYNC_IO  Synchronized input or output operations can be performed for the associated file.

The following constant is defined:

NULL  Null pointer.

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

R_OK  Test for read permission.

W_OK  Test for write permission.

X_OK  Test for execute (search) permission.

F_OK  Test for existence of file.

The constants F_OK, R_OK, W_OK, and X_OK, and the expressions R_OK | W_OK, R_OK | X_OK, and R_OK | W_OK | X_OK all have distinct values.

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

F_ULOCK  Unlock a previously locked region.

F_LOCK  Lock a region for exclusive use.

F_TLOCK  Test and lock a region for exclusive use.

F_TEST  Test a region for other processes locks.

The following symbolic constants are defined for the lseek(2) and fcntl(2) functions (they have distinct values):

SEEK_SET  Set file offset to offset.

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_POSIX_V6_ILP32_OFF32_CFLAGS
_CS_POSIX_V6_ILP32_OFF32_LDFLAGS
_CS_POSIX_V6_ILP32_OFF32_LIBS
_CS_POSIX_V6_ILP32_OFF32_LINTFLAGS
_CS_POSIX_V6_ILP32_OFFBIG_CFLAGS
_CS_POSIX_V6_ILP32_OFFBIG_LDFLAGS
_CS_POSIX_V6_ILP32_OFFBIG_LIBS
_CS_POSIX_V6_ILP32_OFFBIG_LINTFLAGS
_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_POSIX_V6_LP64_OFF64_CFLAGS    _CS_POSIX_V6_LP64_OFF64_LDFLAGS
_CS_POSIX_V6_LP64_OFF64_LIBS      _CS_POSIX_V6_LP64_OFF64_LINTFLAGS
_CS_POSIX_V6_LP64_OFF64_CFLAGS    _CS_POSIX_V6_LP64_OFF64_LDFLAGS
_CS_POSIX_V6_LP64_OFF64_LIBS      _CS_POSIX_V6_LP64_OFF64_LINTFLAGS
_CS_XBS5_LP64_OFF64_CFLAGS        _CS_XBS5_LP64_OFF64_LDFLAGS
_CS_XBS5_LP64_OFF64_LIBS          _CS_XBS5_LP64_OFF64_LINTFLAGS
_CS_XBS5_LP64_OFF64_CFLAGS        _CS_XBS5_LP64_OFF64_LDFLAGS
_CS_XBS5_LP64_OFF64_LIBS          _CS_XBS5_LP64_OFF64_LINTFLAGS
_CS_XBS5_LP64_OFF64_CFLAGS        _CS_XBS5_LP64_OFF64_LDFLAGS
_CS_XBS5_LP64_OFF64_LIBS          _CS_XBS5_LP64_OFF64_LINTFLAGS
```

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

```
_SC_2_C_BIND
_SC_2_C_DEV
```
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>_SC_2_C_VERSION</td>
<td>_SC_2_FORT_DEV</td>
</tr>
<tr>
<td>_SC_2_FORT_RUN</td>
<td>_SC_2_LOCALEDEF</td>
</tr>
<tr>
<td>_SC_2_PBS</td>
<td>_SC_2_PBS_ACCOUNTING</td>
</tr>
<tr>
<td>_SC_2_PBS_CHECKPOINT</td>
<td>_SC_2_PBS_LOCATE</td>
</tr>
<tr>
<td>_SC_2_PBS_MESSAGE</td>
<td>_SC_2_PBS_TRACK</td>
</tr>
<tr>
<td>_SC_2_SW_DEV</td>
<td>_SC_2_UPE</td>
</tr>
<tr>
<td>_SC_2_VERSION</td>
<td>_SC_ADVISORY_INFO</td>
</tr>
<tr>
<td>_SC_AIO_LISTIO_MAX</td>
<td>_SC_AIO_MAX</td>
</tr>
<tr>
<td>_SC_AIO_PRIO_DELTA_MAX</td>
<td>_SC_ARG_MAX</td>
</tr>
<tr>
<td>_SC_ASYNCQUALIFIED_ARG</td>
<td>_SC_ATEXIT_MAX</td>
</tr>
<tr>
<td>_SC_AVPHYS_PAGES</td>
<td>_SC_BARRIERS</td>
</tr>
<tr>
<td>_SC_BC_BASE_MAX</td>
<td>_SC_BC_DIM_MAX</td>
</tr>
<tr>
<td>_SC_BC_SCALE_MAX</td>
<td>_SC_BC_STRING_MAX</td>
</tr>
<tr>
<td>_SC_CHILD_MAX</td>
<td>_SC_CLK_TCK</td>
</tr>
<tr>
<td>_SC_CLOCK_SELECTION</td>
<td>_SC_COLL_WEIGHTS_MAX</td>
</tr>
<tr>
<td>_SC_CPUTIME</td>
<td>_SC_DELAYTIMER_MAX</td>
</tr>
<tr>
<td>_SC_EXPR_NEST_MAX</td>
<td>_SC_FSYNC</td>
</tr>
<tr>
<td>_SC_GETGR_R_SIZE_MAX</td>
<td>_SC_GETPW_R_SIZE_MAX</td>
</tr>
<tr>
<td>_SC_HOST_NAME_MAX</td>
<td>_SC_IOV_MAX</td>
</tr>
<tr>
<td>_SC_IPV6</td>
<td>_SC_JOB_CONTROL</td>
</tr>
<tr>
<td>_SC_LINE_MAX</td>
<td>_SC_LOGIN_NAME_MAX</td>
</tr>
<tr>
<td>_SC_LOGNAME_MAX</td>
<td>_SC_MAPPED_FILES</td>
</tr>
<tr>
<td>_SC_MEMLOCK</td>
<td>_SC_MEMLOCK_RANGE</td>
</tr>
<tr>
<td>_SC_MEMORY_PROTECTION</td>
<td>_SC_MESSAGE_PASSING</td>
</tr>
<tr>
<td>_SC_MONOTONIC_CLOCK</td>
<td>_SC_MQ_OPEN_MAX</td>
</tr>
<tr>
<td>_SC_MQ_PRIO_MAX</td>
<td>_SC_NGROUPS_MAX</td>
</tr>
<tr>
<td>_SC_NPROCEESSORS_CONF</td>
<td>_SC_NPROCEESSORS_ONLN</td>
</tr>
<tr>
<td>_SC_OPEN_MAX</td>
<td>_SC_PAGESIZE</td>
</tr>
<tr>
<td>_SC_PAGE_SIZE</td>
<td>_SC_PAGESIZE</td>
</tr>
</tbody>
</table>

Compiler Directive Symbols
<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>_SC_PHYS_PAGES</code></td>
<td>Physical Memory Pages</td>
</tr>
<tr>
<td><code>_SC_PRIORITY_SCHEDULING</code></td>
<td>Priority Scheduling</td>
</tr>
<tr>
<td><code>_SC_READER_WRITER_LOCKS</code></td>
<td>Reader-Writer Locks</td>
</tr>
<tr>
<td><code>_SC_REEXP</code></td>
<td>Regular Expressions</td>
</tr>
<tr>
<td><code>_SC_RTSIG_MAX</code></td>
<td>RTSIG Max</td>
</tr>
<tr>
<td><code>_SC_SEMAPHORES</code></td>
<td>Semaphores</td>
</tr>
<tr>
<td><code>_SC_SEM_VALUE_MAX</code></td>
<td>Sem Value Max</td>
</tr>
<tr>
<td><code>_SC_SHELL</code></td>
<td>Shell</td>
</tr>
<tr>
<td><code>_SC_SPAWN</code></td>
<td>Spawn</td>
</tr>
<tr>
<td><code>_SC_SPORADIC_SERVER</code></td>
<td>Sporadic Server</td>
</tr>
<tr>
<td><code>_SC_STREAM_MAX</code></td>
<td>Stream Max</td>
</tr>
<tr>
<td><code>_SC_SYNCHRONIZED_IO</code></td>
<td>Synchronized IO</td>
</tr>
<tr>
<td><code>_SC_THREAD_ATTR_STACKADDR</code></td>
<td>Thread Attr Stackaddr</td>
</tr>
<tr>
<td><code>_SC_THREAD_ATTR_STACKSIZE</code></td>
<td>Thread Attr Stacksize</td>
</tr>
<tr>
<td><code>_SC_THREAD_CPUTIME</code></td>
<td>Thread Cputime</td>
</tr>
<tr>
<td><code>_SC_THREAD_DESTRUCTOR_ITERATIONS</code></td>
<td>Thread Destructor Iterations</td>
</tr>
<tr>
<td><code>_SC_THREAD_KEYS_MAX</code></td>
<td>Thread Keys Max</td>
</tr>
<tr>
<td><code>_SC_THREAD_PRIORITY_SCHEDULING</code></td>
<td>Thread Priority Scheduling</td>
</tr>
<tr>
<td><code>_SC_THREAD_PROCESS_SHARED</code></td>
<td>Thread Process Shared</td>
</tr>
<tr>
<td><code>_SC_THREAD_SAFE_FUNCTIONS</code></td>
<td>Thread Safe Functions</td>
</tr>
<tr>
<td><code>_SC_THREAD_STACKADDR</code></td>
<td>Thread Stackaddr</td>
</tr>
<tr>
<td><code>_SC_THREAD_STACK_MIN</code></td>
<td>Thread Stack Min</td>
</tr>
<tr>
<td><code>_SC_THREAD_STACK_MAX</code></td>
<td>Thread Stack Max</td>
</tr>
<tr>
<td><code>_SC_THREAD祉TIMEOUTS</code></td>
<td>Thread Timers</td>
</tr>
<tr>
<td><code>_SC_THREAD祉TIMERS</code></td>
<td>Thread Timers</td>
</tr>
<tr>
<td><code>_SC_TRACE</code></td>
<td>Trace</td>
</tr>
<tr>
<td><code>_SC_TRACE_EVENT_FILTER</code></td>
<td>Trace Event Filter</td>
</tr>
<tr>
<td><code>_SC_TRACE_EVENT_NAME_MAX</code></td>
<td>Trace Event Name Max</td>
</tr>
<tr>
<td><code>_SC_TRACE_INHERIT</code></td>
<td>Trace Inherit</td>
</tr>
<tr>
<td><code>_SC_TRACE_LOG</code></td>
<td>Trace Log</td>
</tr>
<tr>
<td><code>_SC_TRACE_NAME_MAX</code></td>
<td>Trace Name Max</td>
</tr>
<tr>
<td><code>_SC_TRACE_SYS_MAX</code></td>
<td>Trace Sys Max</td>
</tr>
<tr>
<td><code>_SC_TTY_NAME_MAX</code></td>
<td>TTY Name Max</td>
</tr>
<tr>
<td><code>_SC_TZNAME_MAX</code></td>
<td>TZName Max</td>
</tr>
<tr>
<td><code>_SC_V6_ILP32_OFF32</code></td>
<td>V6 Ilp32 Off32</td>
</tr>
<tr>
<td><code>_SC_V6_ILP32_OFFBIG</code></td>
<td>V6 Ilp32 Offbig</td>
</tr>
<tr>
<td><code>_SC_V6_LP64_OFF64</code></td>
<td>V6 Lp64 Off64</td>
</tr>
<tr>
<td><code>_SC_V6_LPBIG_OFFBIG</code></td>
<td>V6 Lpbig Offbig</td>
</tr>
<tr>
<td><code>_SC_XBS5_ILP32_OFF32</code></td>
<td>Xbs5 Ilp32 Off32</td>
</tr>
<tr>
<td><code>_SC_XBS5_ILP32_OFFBIG</code></td>
<td>Xbs5 Ilp32 Offbig</td>
</tr>
</tbody>
</table>
The constants \_SC_PAGESIZE and \_SC_PAGE_SIZE can be defined to have the same value.

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

\_PC_2_SYMLINKS \_PC_ALLOC_SIZE_MIN
\_PC_ASYNC_IO \_PC_CHOWN_RESTRICTED
\_PC_FILESIZEBITS \_PC_LINK_MAX
\_PC_MAX_CANON \_PC_MAX_INPUT
\_PC_NAME_MAX \_PC_NO_TRUNC
\_PC_PATH_MAX \_PC_PIPE_BUF
\_PC_PRIO_IO \_PC_REC_INCR_XFER_SIZE
\_PC_REC_MAX_XFER_SIZE \_PC_REC_MIN_XFER_SIZE
\_PC_REC_XFER_ALIGN \_PC_SYMLINK_MAX
\_PC_SYNC_IO \_PC_TIMESTAMP_RESOLUTION
\_PC_VDISABLE \_PC_XATTR_ENABLED
\_PC_XATTR_EXISTS

The following symbolic constants are defined for file streams:

STDIN_FILENO File number (0) of stdin.
STDOUT_FILENO File number (1) of stdout.
STDERR_FILENO File number (2) of stderr.

The following pathnames are defined:

GF_PATH Pathname of the group file.
PF_PATH Pathname of the passwd file.
Attributes  See `attributes(5)` for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Committed</td>
</tr>
</tbody>
</table>
| Standard         | See `standards(5)`.

See Also  `access(2), exec(2), fcntl(2), fpathconf(2), lseek(2), confstr(3C), lockf(3C), sysconf(3C), termios(3C), group(4), passwd(4), attributes(5), standards(5), termio(7I)`
utime.h, utime – access and modification times structure

#include <utime.h>

The <utime.h> header declares the structure utimbuf, which includes the following members:

```c
    time_t actime /* access time */
    time_t modtime /* modification time */
```

The times are measured in seconds since the Epoch.

The type `time_t` is defined as described in `<sys/types.h>`.

**Attributes**

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

**See Also**

utime(2), types.h(3HEAD), attributes(5), standards(5)
#include <utmpx.h>

The `<utmpx.h>` header defines the `utmpx` structure, which includes the following members:

- `char ut_user[]; /* user login name */`
- `char ut_id[]; /* unspecified initialization */ /* process identifier */`
- `char ut_line[]; /* device name */`
- `pid_t ut_pid; /* process ID */`
- `short ut_type; /* type of entry */`

For X/Open compilation environments:

```c
struct ut_exit_status ut_exit; /* process termination/exit status*/
```

For all other compilation environments:

```c
struct exit_status ut_exit; /* process termination/exit status*/
```

The `pid_t` type is defined through `typedef` as described in `<sys/types.h>`.

The `timeval` structure is defined as described in `<sys/time.h>`.

Inclusion of the `<utmpx.h>` header can also make visible all symbols from `<sys/time.h>`.

The following symbolic constants are defined as possible values for the `ut_type` member of the `utmpx` structure:

- `EMPTY` No valid user accounting information.
- `BOOT_TIME` Identifies time of system boot.
- `OLD_TIME` Identifies time when system clock changed.
- `NEW_TIME` Identifies time after system clock changed.
- `USER_PROCESS` Identifies a process.
- `INIT_PROCESS` Identifies a process spawned by the `init` process.
- `LOGIN_PROCESS` Identifies the session leader of a logged-in user.
- `DEAD_PROCESS` Identifies a session leader who has exited.
Attributes  See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTETYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also  enduxtent(3C), time.h(3HEAD), types.h(3HEAD), attributes(5), standards(5)
utsname.h, utsname – system name structure

Synopsis

#include <sys/utsname.h>

Description

The <sys/utsname.h> header defines the structure utsname, which includes the following members:

```c
char sysname[]  /* name of this implementation of the operating system */
char nodename[] /* name of this node within an implementation-defined communications network */
char release[]  /* current release level of this implementation */
char version[]  /* current version level of this release */
char machine[]  /* name of the hardware type on which the system is running */
```

The character arrays are of unspecified size, but the data stored in them is terminated by a null byte.

Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also

uname(2), attributes(5), standards(5)
values.h(3HEAD)

Name values.h, values – machine-dependent values

Synopsis #include <values.h>

Description This file contains a set of manifest constants, conditionally defined for particular processor architectures.

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.

BITS(type) The number of bits in a specified type (for example, int).
HIBITS The value of a short integer with only the high-order bit set.
HIBITL The value of a long integer with only the high-order bit set.
HIBITI The value of a regular integer with only the high-order bit set.
MAXSHORT The maximum value of a signed short integer.
MAXLONG The maximum value of a signed long integer.
MAXINT The maximum value of a signed regular integer.
MAXFLOAT, LN_MAXFLOAT The maximum value of a single-precision floating-point number, and its natural logarithm.
MAXDOUBLE, LN_MAXDOUBLE The maximum value of a double-precision floating-point number, and its natural logarithm.
MINFLOAT, LN_MINFLOAT The minimum positive value of a single-precision floating-point number, and its natural logarithm.
MINDOUBLE, LN_MINDOUBLE The minimum positive value of a double-precision floating-point number, and its natural logarithm.
FSIGNIF The number of significant bits in the mantissa of a single-precision floating-point number.
DSIGNIF The number of significant bits in the mantissa of a double-precision floating-point number.

See Also Intro(3) math.h(3HEAD)
wait.h, wait – wait status

#include <sys/wait.h>

When a process waits for status from its children using either the `wait(3C)` or `waitpid(3C)` function, the status returned can be evaluated with the following macros, defined in `<sys/wait.h>`. These macros evaluate to integral expressions. The `stat` argument to these macros is the integer value returned from `wait()` or `waitpid()`.

- **WCOREDUMP(stat)**: If the value of `WIFSIGNALED(stat)` is non-zero, this macro evaluates to a non-zero value if a core image of the terminated child was created.
- **WEXITSTATUS(stat)**: If the value of `WIFEXITED(stat)` is non-zero, this macro evaluates to the exit code that the child process passed to `_exit()` (see `exit(2)`) or `exit(3C)`, or the value that the child process returned from `main`.
- **WIFCONTINUED(stat)**: Evaluates to a non-zero value if status was returned for a child process that has continued.
- **WIFEXITED(stat)**: Evaluates to a non-zero value if status was returned for a child process that terminated normally.
- **WIFSIGNALED(stat)**: Evaluates to a non-zero value if status was returned for a child process that terminated due to the receipt of a signal.
- **WIFSTOPPED(stat)**: Evaluates to a non-zero value if status was returned for a child process that is currently stopped.
- **WSTOPSIG(stat)**: If the value of `WIFSTOPPED(stat)` is non-zero, this macro evaluates to the number of the signal that caused the child process to stop.
- **WTERMSIG(stat)**: If the value of `WIFSIGNALED(stat)` is non-zero, this macro evaluates to the number of the signal that caused the termination of the child process.

The `<sys/wait.h>` header defines the symbolic constants listed below for use with `waitpid(3C)`.

- **WNOHANG**: Do not hang if no status is available; return immediately.
- **WUNTRACED**: Report status of stopped child process.

The symbolic constants listed below are defined as possible values for the `options` argument to `waitid(2)`.

- **WEXITED**: Wait for processes that have exited.
- **WSTOPPED**: Status is returned for any child that has stopped upon receipt of a signal.
- **WCONTINUED**: Status is returned for any child that was stopped and has been continued.
- **WNOHANG**: Return immediately if there are no children to wait for.
WNOWAIT

Keep the process whose status is returned in infop in a waitable state.

The type idtype_t is defined as an enumeration type whose possible values include the following:

P. ALL
P. PID
P. PGID

The id_t and pid_t types are defined as described in <sys/types.h>.

The siginfo_t type is defined as described in <signal.h>.

The rusage structure is defined as described in <sys/resource.h>.

Inclusion of the <sys/wait.h> header can also make visible all symbols from <signal.h>
and <sys/resource.h>.

Attributes

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also

exit(2), waitid(2), exit(3C), wait(3C), waitpid(3C), attributes(5), standards(5)


**Name**  wchar.h, wchar – wide-character handling

**Synopsis**  #include <wchar.h>

**Description**  The `<wchar.h>` header defines the following types:

- `wchar_t`: As described in `<stddef.h>`.
- `wint_t`: An integer type capable of storing any valid value of `wchar_t` or `WEOF`.
- `wctype_t`: A scalar type of a data object that can hold values which represent locale-specific character classification.
- `mbstate_t`: An object type other than an array type that can hold the conversion state information necessary to convert between sequences of (possibly multi-byte) characters and wide characters. If a codeset is being used such that an `mbstate_t` needs to preserve more than two levels of reserved state, the results are unspecified.

- `FILE`: As described in `<stdio.h>`.
- `size_t`: As described in `<stddef.h>`.
- `va_list`: As described in `<stdarg.h>`.

The implementation supports one or more programming environments in which the width of `wint_t` is no greater than the width of type `long`. The names of these programming environments can be obtained using the `confstr(3C)` function or the `getconf(1)` utility.

The `<wchar.h>` header defines the following macros:

- `WCHAR_MAX`: The maximum value representable by an object of type `wchar_t`.
- `WCHAR_MIN`: The minimum value representable by an object of type `wchar_t`.
- `WEOF`: Constant expression of type `wint_t` that is returned by several WP functions to indicate end-of-file.
- `NULL`: As described in `<stddef.h>`.

The tag `tm` is declared as naming an incomplete structure type, the contents of which are described in the header `<time.h>`.

Inclusion of the `<wchar.h>` header can make visible all symbols from the headers `<ctype.h>`, `<string.h>`, `<stdarg.h>`, `<stddef.h>`, `<stdio.h>`, `<stdlib.h>`, and `<time.h>`.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>
See Also  getconf(1), btowc(3C), confstr(3C), fgetwc(3C), getws(3C), fputwc(3C), fputws(3C), fwide(3C), fwprintf(3C), fwscanf(3C), getwc(3C), getwchar(3C), iswalpha(3C), iswctype(3C), mbsinit(3C), mbrlen(3C), mbtowc(3C), mbsrtowcs(3C), towlower(3C), towupper(3C), ungetwc(3C), vfwpprintf(3C), wcrtomb(3C), wcsrtombs(3C), wcstring(3C), wcswrchr(3C), wcscoll(3C), wcssort(3C), wctob(3C), wcsftime(3C), wcstold(3C), wcstoul(3C), wcswidth(3C), wcscxfrm(3C), wctob(3C), wctype(3C), wcwidth(3C), wmemchr(3C), wmemcmp(3C), wmemcpy(3C), wmemmove(3C), wmemset(3C), stdarg(3EXT), stddef.h(3HEAD), stdio.h(3HEAD), stdlib.h(3HEAD), string.h(3HEAD), time.h(3HEAD), wchar.h(3HEAD), wctype.h(3HEAD), attributes(5), standards(5)
Name  wctype.h, wctype – wide-character classification and mapping utilities

Synopsis  #include <wctype.h>

Description  The <wctype.h> header defines the following types:

- wint_t  As described in <wchar.h>.
- wctrans_t  A scalar type that can hold values that represent locale-specific character mappings.
- wctype_t  As described in <wchar.h>.

The <wctype.h> header defines the following macro name:

- WEOF  Constant expression of type wint_t that is returned by several MSE functions to indicate end-of-file.

For all functions described in this header that accept an argument of type wint_t, the value is representable as a wchar_t or equals the value of WEOF. If this argument has any other value, the behavior is undefined.

The behavior of these functions is affected by the LC_CTYPE category of the current locale.

Inclusion of the <wctype.h> header can make visible all symbols from the headers <ctype.h>, <stdarg.h>, <stddef.h>, <stdio.h>, <stdlib.h>, <string.h>, <time.h>, and <wchar.h>.

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

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also  iswalpha(3C), iswctype(3C), locale.h(3HEAD), setlocale(3C), stdarg(3EXT), stddef.h(3HEAD), stdio.h(3HEAD), stdlib.h(3HEAD), string.h(3HEAD), time.h(3HEAD), towctrans(3C), tolower(3C), towupper(3C), wctrans(3C), wctype(3C), attributes(5), standards(5)
The `<wordexp.h>` header defines the structures and symbolic constants used by the `wordexp()` and `wordfree()` functions. See `wordexp(3C)`.

The structure type `wordexp_t` contains the following members:

- `size_t we_wordc /* count of words matched by words */`
- `char **we_wordv /* pointer to list of expanded words */`
- `size_t we_offs /* slots to reserve at the beginning of we_wordv */`

The `flags` argument to the `wordexp()` function is the bitwise-inclusive OR of the following flags:

- `WRDE_APPEND` Append words to those previously generated.
- `WRDE_DOOFFS` Number of null pointers to prepend to `we_wordv`.
- `WRDE_NOCMD` Fail if command substitution is requested.
- `WRDE_REUSE` The `pwordexp` argument was passed to a previous successful call to `wordexp()`, and has not been passed to `wordfree()`. The result is the same as if the application had called `wordfree()` and then called `wordexp()` without `WRDE_REUSE`.
- `WRDE_SHOWERR` Do not redirect `stderr` to `/dev/null`.
- `WRDE_UNDEF` Report error on an attempt to expand an undefined shell variable.

The following constants are defined as error return values:

- `WRDE_BADCHAR` One of the unquoted characters—`<newline>`, `]` `,` `&` `>` `(` `)` `]`—appears in words in an inappropriate context.
- `WRDE_BADVAL` Reference to undefined shell variable when `WRDE_UNDEF` is set in `flags`.
- `WRDE_CMDSUB` Command substitution requested when `WRDE_NOCMD` was set in `flags`.
- `WRDE_NOSPACE` Attempt to allocate memory failed.
- `WRDE_NOSYS` Reserved.
- `WRDE_Syntax` Shell syntax error, such as unbalanced parentheses or unterminated string.

The `<wordexp.h>` header defines the following type:

- `size_t` As described in `<stdbool.h>`.
Attributes  See attributes(5) for descriptions of the following attributes:

<table>
<thead>
<tr>
<th>ATTRIBUTE TYPE</th>
<th>ATTRIBUTE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface Stability</td>
<td>Standard</td>
</tr>
</tbody>
</table>

See Also  wordexp(3C), attributes(5), standards(5)