```
**********************************************************
   32494 Sun Sep 16 19:22:52 2018
new/usr/src/man/man1/crle.1
9842 man page typos and spelling
**********************************************************
   1 '\" te
   2 .\"  Copyright (c) 2008, Sun Microsystems, Inc. All Rights Reserved
   3 .\" The contents of this file are subject to the terms of the Common Development
   4 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
   5 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
   6 .TH CRLE 1 "Oct 6, 2008"
   7 .SH NAME
   8 crle \- configure runtime linking environment
   9 .SH SYNOPSIS
  10 .LP
  11 .nf
  12 \fBcrle\fR [\fB-64\fR] [\fB-a\fR \fIname\fR] [\fB-A\fR \fIname\fR] [\fB-c\fR \fI
  13      [\fB-f\fR \fIflags\fR] [\fB-i\fR \fIname\fR] [\fB-I\fR \fIname\fR] [\fB-g\f
  14      [\fB-l\fR \fIdir\fR] [\fB-o\fR \fIdir\fR] [\fB-s\fR \fIdir\fR] [\fB-t\fR [
  15 .fi

  17 .SH DESCRIPTION
  18 .sp
  18 .LP
  19 The \fBcrle\fR utility provides for the creation and display of a runtime
  20 linking configuration file. The configuration file is read and interpreted by
  21 the runtime linker, \fBld.so.1\fR(1), during process startup. The runtime
  22 linker attempts to read a default configuration file for all processes. For
  23 32-bit processes, the default configuration file is \fB/var/ld/ld.config\fR.
  24 For 64-bit processes, the default configuration file is
  25 \fB/var/ld/64/ld.config\fR.
  26 .sp
  27 .LP
  28 Without any arguments, or with just the \fB-c\fR option, \fBcrle\fR displays
  29 configuration information. This information includes the contents of a
  30 configuration file, any system defaults and the command-line required to
  31 regenerate the configuration file. When used with any other options, a new
  32 configuration file is created or updated.
  33 .sp
  34 .LP
  35 The runtime linker can also be directed to an alternative configuration file by
  36 setting one of the \fBLD_CONFIG\fR family of environment variable.
  37 \fBLD_CONFIG\fR applies to both 32-bit and 64-bit programs. Since 32-bit and
  38 64-bit configuration files differ, a single configuration file cannot be used
  39 for both class of object. Hence, \fBLD_CONFIG\fR can adversely affect program
  40 execution in cases where a program of one class executes a program of the other
  41 class. In particular, it is common practice for the 32-bit version of standard
  42 Solaris utilities to execute their 64-bit counterpart. \fBLD_CONFIG\fR cannot
  43 be successfully used in this case. Therefore, the use of the \fBLD_CONFIG_32\fR
  44 and \fBLD_CONFIG_64\fR environment variables, that precisely target the
  45 appropriate class of process, is recommended.
  46 .sp
  47 .LP
  48 Creating an incorrect configuration file in the standard location,
  49 \fB/var/ld\fR, can prevent programs from running, and can therefore be
  50 difficult to recover from. To guard against this situation, it is recommended
  51 difficult to recover from. To guard against this situation, it is recommented
  51 that new configuration files first be created in a temporary location. Then set
  52 the appropriate \fBLD_CONFIG\fR environment variable to this new configuration
  53 file. This setting causes the new configuration file to be used by the runtime
  54 linker instead of any default. After verification, the new configuration file
  55 can be moved to the default location if desired. At any time, the environment
  56 variable \fBLD_NOCONFIG\fR can be set to any value to instruct the runtime
  57 linker to ignore any configuration files. This setting can prove useful during
  58 experimentation.
  59 .sp
```



```
  60 .LP
  61 A configuration file can contain the following information.
  62 .sp
  63 .ne 2
  64 .na
  65 \fBDefault Search Paths\fR
  66 .ad
  67 .sp .6
  68 .RS 4n
  69 The runtime linker uses a prescribed search path for locating the dynamic
  70 dependencies of an object. This search path starts with the components of any
  71 \fBLD_LIBRARY_PATH\fR definition, followed by the components of an object's
  72 \fBrunpath\fR. Finally, any default search paths specific to the object's class
  73 are used. This last component of the search path can be expressed within the
  74 configuration file. Typically, use of this facility should be augmented with
  75 any system default. See the \fB-l\fR and \fB-u\fR options.
  76 .RE

  78 .sp
  79 .ne 2
  80 .na
  81 \fBTrusted Directories\fR
  82 .ad
  83 .sp .6
  84 .RS 4n
  85 When processing a secure application, the runtime linker restricts the use of
  86 \fBLD_LIBRARY_PATH\fR searches, and \fB$ORIGIN\fR token expansion. See
  87 \fISecurity\fR in \fILinker and Libraries Guide\fR. In addition, the
  88 directories from which preload and audit libraries can be located are also
  89 restricted. The path names that are associated with preload and audit libraries
  90 are restricted to known trusted directories. Trusted directories can be
  91 expressed within the configuration file. Typically, use of this facility should
  92 be augmented with any system defaults. See the \fB-s\fR and \fB-u\fR options.
  93 .RE

  95 .sp
  96 .ne 2
  97 .na
  98 \fBEnvironment Variables\fR
  99 .ad
 100 .sp .6
 101 .RS 4n
 102 Any environment variable interpreted by the runtime linker can be specified
 103 within the configuration file.
 104 .RE

 106 .sp
 107 .ne 2
 108 .na
 109 \fBDirectory Cache\fR
 110 .ad
 111 .sp .6
 112 .RS 4n
 113 The location of shared objects within defined directories can be maintained as
 114 a cache within the configuration file. This directory cache can reduce the
 115 overhead of searching for application dependencies.
 116 .RE

 118 .sp
 119 .ne 2
 120 .na
 121 \fBAlternative Objects\fR
 122 .ad
 123 .sp .6
 124 .RS 4n
 125 In conjunction with the directory cache, shared objects can have alternative
```

objects specified for use at runtime. These alternate objects, can be supplied by the user. Alternative objects can also be created by \fBcrle\fR as copies of shared objects fixed to known memory locations. These fixed alternative objects can require less processing at runtime than their original shared object counterpart.
.RE

.sp
.LP
Defining additional default search paths, or additional trusted directories can be useful for administrators who wish to install third party software in a central location, or otherwise alter the search path of applications that might not have been coded with a suitable runpath.
.sp
.LP
The declaration of alternative objects provides a means of replacing dependencies other than by using symbolic links or requiring \fBLD_LIBRARY_PATH\fR settings.
.sp
.LP
The declaration of environment variables that are interpreted by the runtime linker provides a means of centralizing their definition for all applications.
.sp
.LP
The directory cache, and \fBcrle\fR generated alternate objects, can provide a means of reducing the runtime startup overhead of applications. Alternative objects can be useful for applications that require many dependencies, or whose dependencies are expensive to relocate. Shared objects that contain \fBposition-dependent\fR code are often expensive to relocate. Note, the system has many caching facilities that help mitigate expenses such as negative path lookups, and thus employing \fBcrle\fR to create a directory cache may have minimal effect other than for some very specific cases.
.sp
.LP
When alternate objects that are generated by \fBcrle\fR are specified within a configuration file, the runtime linker performs some minimal consistency verification. The alternative objects are verified against their originating objects. This verification is intended to avert application failure should an applications configuration information become out-of-sync with the underlying system components. When this situation arises the flexibility offered by dynamic linking system components can be compromised. This type of application failure can be very difficult to diagnose. No verification of directory cache information is performed. Any changes to the directory structure are not seen by a process until the cache is rebuilt.
.sp
.LP
System shared objects are often well tuned, and can show little benefit from being cached. The directory cache and alternative object features are typically applicable to user applications and shared objects, and may only show improvement in some very specific cases.
.sp
.LP
\fBcrle\fR creates alternate objects for the shared objects that are discovered when using the \fB-I\fR and \fB-G\fR options, using \fBdldump\fR(3C). The alternate object is created in the directory specified by the preceding \fB-o\fR option, or defaults to the directory in which the configuration file is created. The flags used by \fBdldump()\fR are specified using the \fB-f\fR option, or default to \fBRTLD_REL_RELATIVE\fR.
.SH OPTIONS
.*sp*
.LP
The following options are supported.
.sp
.ne 2
.na
\fB\fB-64\fR\fR

.ad
.sp .6
.RS 4n
Specify to process 64-bit objects, the default is 32-bit. Use \fB-64\fR to create a 64-bit specific configuration file.
.RE

.sp
.ne 2
.na
\fB\fB-a\fR \fIname\fR\fR
.ad
.sp .6
.RS 4n
Create an alternative path name for \fIname\fR. The alternative path name is added to the configuration file.
.sp
The actual alternative file must be supplied by the user. Multiple occurrences of this option are permitted. If \fIname\fR is a directory, each shared object within the directory is added to the cache. If \fIname\fR does not exist, then \fIname\fR is marked in the cache as a nonexistent file.
.sp
Typically, this option is used with the \fB-o\fR option.
.RE

.sp
.ne 2
.na
\fB\fB-A\fR \fIname\fR\fR
.ad
.sp .6
.RS 4n
Create an optional alternative path name for \fIname\fR. This alternative path name is added to the configuration file.
.sp
This option mimics the \fB-a\fR option, except that if the alternative is unavailable at runtime, the original object \fIname\fR is used. This model mimics the use of auxiliary filters. See \fIGenerating Auxiliary Filters\fR in \fILinker and Libraries Guide\fR.
.sp
Typically, this option is used with the \fB-o\fR option.
.RE

.sp
.ne 2
.na
\fB\fB-c\fR \fIconf\fR\fR
.ad
.sp .6
.RS 4n
Specify to use the configuration file name \fIconf\fR. If this option is not supplied, the default configuration file is used.
.RE

.sp
.ne 2
.na
\fB\fB-e\fR \fIenv\fR\fR
.ad
.sp .6
.RS 4n
Specify a \fBreplaceable\fR environment variable, \fIenv\fR. Only environment variables that are applicable to the runtime linker are meaningful. Multiple occurrences of this option are permitted. This option is similar to the \fB-E\fR option. However, the options differs in how configuration file definitions, and process environment definitions of the same name are resolved

```
257 at runtime.
258 .sp
259 A definition established in a configuration file can be \fBoverridden\fR by a
260 process environment definition, or be \fBsuppressed\fR by a null-value process
261 environment definition.
262 .sp
263 In other words, these configuration file definitions can be replaced, or
264 removed by the process environment at runtime.
265 .RE

267 .sp
268 .ne 2
269 .na
270 \fB\fB-E\fR \fIenv\fR\fR
271 .ad
272 .sp .6
273 .RS 4n
274 Specify a \fBpermanent\fR environment variable, \fIenv\fR. Only environment
275 variables that are applicable to the runtime linker are meaningful. Multiple
276 occurrences of this option are permitted. This option is similar to the
277 \fB-e\fR option. However, the option differs in how configuration file
278 definitions, and process environment definitions of the same name are resolved
279 at runtime.
280 .sp
281 Environment variable definitions that are meaningful to the runtime linker fall
282 into one of two categories. Singular definitions are definitions such as
283 \fBLD_NOLAZYLOAD=1\fR and \fBLD_DEBUG_OUTPUT=\fR\fIfile\fR. List definitions,
284 which can take one or more values, are definitions such as
285 \fBLD_LIBRARY_PATH=\fR\fIpath\fR, and \fBLD_DEBUG=\fR\fIfiles\fR,\fIdetails\fR.
286 .sp
287 A singular definition that is established in a configuration file takes
288 precedence over a process environment definition. A list definition that is
289 established in a configuration file is \fBappended\fR to a process environment
290 definition. Any definition that is established in a configuration file can
291 \fBnot\fR be suppressed by a null-value process environment definition.
292 .sp
293 In other words, these configuration file definitions can \fBnot\fR be replaced,
294 or removed by the process environment at runtime.
295 .RE

297 .sp
298 .ne 2
299 .na
300 \fB\fB-f\fR \fIflags\fR\fR
301 .ad
302 .sp .6
303 .RS 4n
304 Provide the symbolic \fIflags\fR argument to the \fBdldump\fR(3C) calls used to
305 generate alternate objects. Any of the \fBRTLD_REL\fR flags that are defined in
306 \fB/usr/include/dlfcn.h\fR can be used. Multiple flags can be \fBor\fR'ed
307 together using the "\fB|\fR" character. In this case, the string should be
308 quoted to avoid expansion by the shell. If no \fIflags\fR values are provided
309 the default flag is \fBRTLD_REL_RELATIVE\fR.
310 .RE

312 .sp
313 .ne 2
314 .na
315 \fB\fB-i\fR \fIname\fR\fR
316 .ad
317 .sp .6
318 .RS 4n
319 Add an individual \fIname\fR to the configuration cache. Multiple occurrences
320 of this option are permitted. \fIname\fR can be a shared object or a directory.
321 If \fIname\fR is a directory, each shared object within the directory is added
322 to the cache. If \fIname\fR does not exist, the \fIname\fR is marked in the
```



```
323 cache as a nonexistent directory.
324 .RE

326 .sp
327 .ne 2
328 .na
329 \fB\fB-I\fR \fIname\fR\fR
330 .ad
331 .sp .6
332 .RS 4n
333 Mimic the \fB-i\fR, and in addition any shared object that is processed has an
334 alternative created using \fBdldump\fR(3C). If the \fB-f\fR flag contains
335 \fBRTLD_REL_EXEC\fR, then \fIname\fR can be a dynamic executable, for which an
336 alternative is created. Only one dynamic executable can be specified in this
337 manner, as the cache that is created is specific to this application.
338 .RE

340 .sp
341 .ne 2
342 .na
343 \fB\fB-g\fR \fIname\fR\fR
344 .ad
345 .sp .6
346 .RS 4n
347 Add the group \fIname\fR to the configuration cache. Each object is expanded to
348 determine its dependencies. Multiple occurrences of this option are permitted.
349 \fIname\fR can be a dynamic executable, shared object or a directory. If
350 \fIname\fR is a shared object, the shared object and its dependencies are added
351 to the cache. If \fIname\fR is a directory, each shared object within the
352 directory, and its dependencies, are added to the cache.
353 .RE

355 .sp
356 .ne 2
357 .na
358 \fB\fB-G\fR \fIname\fR\fR
359 .ad
360 .sp .6
361 .RS 4n
362 Mimic the \fB-g\fR option, and in addition any shared object that is processed
363 has an alternative created using \fBdldump\fR(3C). If \fIname\fR is a dynamic
364 executable, and the \fB-f\fR flag contains \fBRTLD_REL_EXEC\fR, then an
365 alternative for the dynamic executable is also created. Only one dynamic
366 executable can be specified in this manner as the cache that is created is
367 specific to this application.
368 .RE

370 .sp
371 .ne 2
372 .na
373 \fB\fB-l\fR \fIdir\fR\fR
374 .ad
375 .sp .6
376 .RS 4n
377 Specify a new default search directory \fIdir\fR for \fBELF\fR or \fBAOUT\fR
378 objects. Multiple occurrences of this option are permitted. The type of object
379 that is applicable to the search, is specified by the preceding \fB-t\fR
380 option, or defaults to \fBELF\fR.
381 .sp
382 The default search paths for 32-bit \fBELF\fR objects are \fB/lib\fR followed
383 by \fB/usr/lib\fR. For 64-bit \fBELF\fR objects, the default search paths are
384 \fB/lib/64\fR followed by \fB/usr/lib/64\fR.
385 .sp
386 The default search paths for \fBAOUT\fR objects are \fB/usr/4lib\fR, followed
387 by \fB/usr/lib\fR and finally \fB/usr/local/lib\fR.
388 .sp
```

Use of this option \fBreplaces\fR the default search path. Therefore, a \fB-l\fR option is normally required to specify the original system default in relation to any new paths that are being applied. However, if the \fB-u\fR option is in effect, and a configuration file does \fBnot\fR exist, the system defaults are added to the new configuration file. These defaults are added before the new paths specified with the \fB-l\fR option.
.RE

.sp
.ne 2
.na
\fB\fB-o\fR \fIdir\fR\fR
.ad
.sp .6
.RS 4n
When used with either the \fB-a\fR or \fB-A\fR options, specifies the directory \fIdir\fR in which any alternate objects exist. When alternative objects are created by \fBcrle\fR, this option specified where the alternative are created. Without this option, alternate objects exist in the directory in which the configuration file is created. Multiple occurrences of this option are permitted, the directory \fIdir\fR being used to locate alternatives for any following command-line options. Alternative objects are not permitted to override their associated originals.
.sp
Typically, this option is used with the \fB-a\fR or \fB-A\fR options.
.RE

.sp
.ne 2
.na
\fB\fB-s\fR \fIdir\fR\fR
.ad
.sp .6
.RS 4n
Specify a new trusted directory \fIdir\fR for \fIsecure\fR \fBELF\fR or \fBAOUT\fR objects. See \fBSECURITY\fR in \fBld.so.1\fR(1) for a definition of secure objects. See \fISecurity\fR in \fILinker and Libraries Guide\fR for a discussion of runtime restrictions imposed on secure applications.
.sp
Multiple occurrences of this option are permitted. The type of object that is applicable to the search is specified by the preceding \fB-t\fR option, or defaults to \fBELF\fR.
.sp
The default trusted directories for secure 32-bit \fBELF\fR objects, and \fBAOUT\fR objects, are \fB/lib/secure\fR followed by \fB/usr/lib/secure\fR. For 64-bit secure \fBELF\fR objects, the default trusted directories are \fB/lib/secure/64\fR followed by \fB/usr/lib/secure/64\fR.
.sp
Use of this option \fBreplaces\fR the default trusted directories. Therefore, a \fB-s\fR option is normally required to specify the original system default in relation to any new directories that are being applied. However, if the \fB-u\fR option is in effect, and a configuration file does \fBnot\fR exist, the system defaults are added to the new configuration file. These defaults are added before the new directories specified with the \fB-l\fR option.
.RE

.sp
.ne 2
.na
\fB\fB-t\fR \fBELF\fR | \fBAOUT\fR\fR
.ad
.sp .6
.RS 4n
Toggle the object type that is applicable to any \fB-l\fR or \fB-s\fR options that follow. The default object type is \fBELF\fR.
.RE

.sp
.ne 2
.na
\fB\fB-u\fR\fR
.ad
.sp .6
.RS 4n
Request that a configuration file be updated, possibly with the addition of new information. Without other options, any existing configuration file is inspected and its contents recomputed. Additional arguments allow information to be appended to the recomputed contents. See NOTES.
.sp
If a configuration file does not exist, the configuration file is created as directed by the other arguments. In the case of the \fB-l\fR and \fB-s\fR options, any system defaults are first applied to the configuration file before the directories specified with these options.
.sp
The configuration file can be in the older format that lacks the system identification information that is normally written at the beginning of the file. In this case, \fBcrle\fR does not place system identification information into the resulting file, preserving compatibility of the file with older versions of Solaris. See NOTES.
.RE

.sp
.ne 2
.na
\fB\fB-v\fR\fR
.ad
.sp .6
.RS 4n
Specify verbose mode. When creating a configuration file, a trace of the files that are being processed is written to the standard out. When printing the contents of a configuration file, more extensive directory and file information is provided.
.RE

.sp
.LP
By default, the runtime linker attempts to read the configuration file \fB/var/ld/ld.config\fR for each 32-bit application processed. \fB/var/ld/64/ld.config\fR is read for each 64-bit application. When processing an alternative application, the runtime linker uses a \fB$ORIGIN/ld.config.\fIapp-name\fR\fR configuration file if present. See NOTES. Applications can reference an alternative configuration file by setting the \fBLD_CONFIG\fR environment variable. An alternative configuration file can also be specified by recording the configuration file name in the application at the time the application is built. See the \fB-c\fR option of \fBld\fR(1).
.SH EXAMPLES
.LP
\fBExample 1 \fRExperimenting With a Temporary Configuration File
.sp
.LP
The following example creates a temporary configuration file with a new default search path for ELF objects. The environment variable \fBLD_CONFIG_32\fR is used to instruct the runtime linker to use this configuration file for all 32-bit processes.

.sp
.in +2
.nf
$ \fBcrle -c /tmp/ld.config -u -l /local/lib\fR
$ \fBcrle -c /tmp/ld.config\fR

Configuration file [version 4]: /tmp/ld.config

```
   Platform:     32-bit MSB SPARC
   Default Library Path (ELF):  /lib:/usr/lib:/local/lib
   Trusted Directories (ELF):   /lib/secure:/usr/lib/secure  \e
                                (system default)

Command line:
  crle -c /tmp/ld.config -l /lib:/usr/lib:/local/lib

$ \fBLD_CONFIG_32=/tmp/ld.config date\fR
Thu May 29 17:42:00 PDT 2008
.fi
.in -2
.sp

.LP
\fBExample 2 \fRUpdating and Displaying a New Default Search Path for ELF
Objects
.sp
.LP
The following example updates and displays a new default search path for ELF
objects.

.sp
.in +2
.nf
# \fBcrle -u -l /local/lib\fR
# \fBcrle\fR

Configuration file [version 4]: /var/ld/ld.config
  Platform:     32-bit MSB SPARC
  Default Library Path (ELF):  /lib:/usr/lib:/local/lib
  Trusted Directories (ELF):   /lib/secure:/usr/lib/secure  \e
                               (system default)

Command line:
  crle -l /lib:/usr/lib:/local/lib

# \fBcrle -u -l /ISV/lib\fR
# \fBcrle\fR

Configuration file [version 4]: /var/ld/ld.config
  Platform      32-bit MSB SPARC
  Default Library Path (ELF):  /lib:/usr/lib:/local/lib:/ISV/lib
  Trusted Directories (ELF):   /lib/secure:/usr/lib/secure  \e
                               (system default)

Command line:
  crle -l /lib:/usr/lib:/local/lib:/usr/local/lib
.fi
.in -2
.sp

.sp
.LP
In this example, the default configuration file initially did not exist.
Therefore, the new search path \fB/local/lib\fR is appended to the system
default. The next update appends the search path \fB/ISV/lib\fR to those paths
already established in the configuration file.

.LP
\fBExample 3 \fRRecovering From a Bad Configuration File
.sp
.LP
The following example creates a bad configuration file in the default location.
The file can be removed by instructing the runtime linker to ignore any
configuration file with the \fBLD_NOCONFIG\fR environment variable. Note, it is
```



```
recommended that temporary configuration files be created and the environment
variable \fBLD_CONFIG\fR used to experiment with these files.

.sp
.in +2
.nf
# \fBcrle -l /local/lib\fR
# \fBdate\fR
ld.so.1: date: fatal: libc.so.1: open failed:  \e
   No such file or directory
Killed
# \fBLD_NOCONFIG=yes rm /var/ld/ld.config\fR
# \fBdate\fR
Thu May 29 17:52:00 PDT 2008
.fi
.in -2
.sp

.sp
.LP
Note, the reason the configuration file is bad is because the system default
search paths are not present. Hence, the \fBdate\fR utility is not able to
locate the system dependencies that it required. In this case, the \fB-u\fR
option should have been used.
.LP
\fBExample 4 \fRCreating and Displaying a New Default Search Path and New
Trusted Directory for ELF Objects
.sp
.LP
The following example creates and displays a new default search path and new
trusted directory for ELF objects.

.sp
.in +2
.nf
# \fBcrle -l /local/lib -l /lib -l /usr/lib -s /local/lib\fR
# \fBcrle\fR

Configuration file [version 4]: /var/ld/ld.config
  Platform:     32-bit MSB SPARC
  Default Library Path (ELF):  /local/lib:/lib:/usr/lib
  Trusted Directories (ELF):   /local/lib

Command line:
  crle -l /local/lib:/lib:/usr/lib -s /local/lib
.fi
.in -2
.sp

.sp
.LP
With this configuration file, third party applications could be installed in
\fB/local/bin\fR and their associated dependencies in \fB/local/lib\fR. The
default search path allows the applications to locate their dependencies
without the need to set \fBLD_LIBRARY_PATH\fR. The default trusted directories
have also been replaced with this example.

.LP
\fBExample 5 \fRCreating a Directory Cache for ELF Objects
.sp
.LP
The following example creates a directory cache for ELF objects.

.sp
.in +2
.nf
```

```
$ \fBcrle -i /usr/dt/lib -i /usr/openwin/lib -i /lib -i /usr/lib  \e
        -c config\fR
$ \fBldd -s ./main\fR
\&....
   find object=libc.so.1; required by ./main
    search path=/usr/dt/lib:/usr/openwin/lib  (RUNPATH/RPATH ./main)
    trying path=/usr/dt/lib/libc.so.1
    trying path=/usr/openwin/lib/libc.so.1
    search path=/lib  (default)
    trying path=/lib/libc.so.1
        libc.so.1 =>     /lib/libc.so.1

$ \fBLD_CONFIG=config ldd -s ./main\fR
\&....
   find object=libc.so.1; required by ./main
    search path=/usr/dt/lib:/usr/openwin/lib  (RUNPATH/RPATH ./main)
    search path=/lib  (default)
    trying path=/lib/libc.so.1
        libc.so.1 =>     /lib/libc.so.1
.fi
.in -2
.sp

.sp
.LP
With this configuration, the cache reflects that the system library
\fBlibc.so.1\fR does not exist in the directories \fB/usr/dt/lib\fR or
\fB/usr/openwin/lib\fR. Therefore, the search for this system file ignores
these directories even though the application's runpath indicates these paths
should be searched.

.LP
\fBExample 6 \fRCreating an Alternative Object Cache for an ELF Executable
.sp
.LP
The following example creates an alternative object cache for an ELF
executable.

.sp
.in +2
.nf
$ \fBcrle -c /local/$HOST/.xterm/ld.config.xterm  \e
        -f RTLD_REL_ALL  -G /usr/openwin/bin/xterm\fR
$ \fBln  -s /local/$HOST/.xterm/xterm  /local/$HOST/xterm\fR
$ \fBldd /usr/local/$HOST/xterm\fR
    libXaw.so.5 =>  /local/$HOST/.xterm/libWaw.so.5  (alternate)
    libXmu.so.4 =>  /local/$HOST/.xterm/libXmu.so.4  (alternate)
    ....
    libc.so.1 =>    /local/$HOST/.xterm/libc.so.1  (alternate)
    ....
.fi
.in -2
.sp

.sp
.LP
With this configuration, a new \fBxterm\fR and its dependencies are created.
These new objects are fully relocated to each other, and result in faster
startup than the originating objects. The execution of this application uses
its own specific configuration file. This model is generally more flexible than
using the environment variable \fBLD_CONFIG\fR, as the configuration file can
not be erroneously used by other applications such as \fBldd\fR(1) or
\fBtruss\fR(1).

.LP
\fBExample 7 \fRCreating an Alternative Object Cache to Replace an ELF Shared
```



```
Object
.sp
.LP
The following example creates an alternative object cache to replace an ELF
shared object.

.sp
.in +2
.nf
$ \fBldd /usr/bin/vi\fR
    libcurses.so.1 =>  /lib/libcurses.so.1
    ....

# \fBcrle -a /lib/libcurses.so.1 -o /usr/ucblib\fR
# \fBcrle\fR

Configuration file [version 4]: /var/ld/ld.config
  Platform:     32-bit MSB SPARC
  Default Library Path (ELF):  /lib:/usr/lib  (system default)
  Trusted Directories (ELF):   /lib/secure:/usr/lib/secure  \e
                               (system default)

Directory: /lib
  libcurses.so.1  (alternate: /usr/ucblib/libcurses.so.1)
\&....

$ \fBldd /usr/bin/vi\fR
    libcurses.so.1 => /usr/ucblib/libcurses.so.1 (alternate)
    ....
.fi
.in -2
.sp

.sp
.LP
With this configuration, any dependency that would normally resolve to
\fB/usr/lib/libcurses.so.1\fR instead resolves to
\fB/usr/ucblib/libcurses.so.1\fR.

.LP
\fBExample 8 \fRSetting Replaceable and Permanent Environment Variables
.sp
.LP
The following example sets replaceable and permanent environment variables.

.sp
.in +2
.nf
# \fBcrle -e LD_LIBRARY_PATH=/local/lib  \e
        -E LD_PRELOAD=preload.so.1\fR
# \fBcrle\fR
\&.....
Environment Variables:
  LD_LIBRARY_PATH=/local/lib  (replaceable)
  LD_PRELOAD=preload.so.1  (permanent)

\&.....
$ \fBLD_DEBUG=files LD_PRELOAD=preload.so.2 ./main\fR
\&.....
18764: file=preload.so.2;  preloaded
18764: file=/local/lib/preload.so.2  [ ELF ]; generating link map
\&.....
18764: file=preload.so.1;  preloaded
18764: file=/local/lib/preload.so.1  [ ELF ]; generating link map
\&.....
.fi
```

```
.in -2
.sp

.sp
.LP
With this configuration file, a replaceable search path has been specified
together with a permanent preload object which becomes appended to the process
environment definition.

.SH EXIT STATUS
.sp
.LP
The creation or display of a configuration file results in a \fB0\fR being
returned. Otherwise, any error condition is accompanied with a diagnostic
message and a non-zero value being returned.
.SH NOTES
.sp
.LP
The ability to tag an alternative application to use an application-specific
configuration file, is possible if the original application contains one of the
\fI\&.dynamic\fR tags \fBDT_FLAGS_1\fR or \fBDT_FEATURE_1\fR. Without these
entries, a configuration file must be specified using the \fBLD_CONFIG\fR
environment variable. Care should be exercised with this latter method as this
environment variable is visible to any forked applications.
.sp
.LP
The use of the \fB-u\fR option requires at least version 2 of \fBcrle\fR. This
version level is evident from displaying the contents of a configuration file.
.sp
.in +2
.nf
$ \fBcrle\fR

Configuration file [2]: /var/ld/ld.config
  ......
.fi
.in -2
.sp

.sp
.LP
With a version 2 configuration file, \fBcrle\fR is capable of constructing the
command-line arguments required to regenerate the configuration file. This
command-line construction, provides full update capabilities using the \fB-u\fR
option. Although a version 1 configuration file update is possible, the
configuration file contents might be insufficient for \fBcrle\fR to compute the
entire update requirements.
.sp
.LP
Configuration files contain platform specific binary data. A given
configuration file can only be interpreted by software with the same machine
class and byte ordering. However, the information necessary to enforce this
restriction was not included in configuration files until \fBSXCE\fR build
\fB41\fR. As of this \fBSXCE\fR build, configuration files have system
identification information at the beginning of the file. This additional
information is used by \fBcrle\fR and the runtime to check their compatibility
with configuration files. This information also allows the \fBfile\fR(1)
command to properly identify  configuration files. For backward compatibility,
older files that are missing this information are still accepted, although
without the identification and error checking that would otherwise be possible.
When processing an update (\fB-u\fR) operation for an older file that lacks
system information, \fBcrle\fR does not add system identification information
to the result.
.SH FILES
.sp
.ne 2
```



```
.na
\fB\fB/var/ld/ld.config\fR\fR
.ad
.sp .6
.RS 4n
Default configuration file for 32-bit applications.
.RE

.sp
.ne 2
.na
\fB\fB/var/ld/64/ld.config\fR\fR
.ad
.sp .6
.RS 4n
Default configuration file for 64-bit applications.
.RE

.sp
.ne 2
.na
\fB\fB/var/tmp\fR\fR
.ad
.sp .6
.RS 4n
Default location for temporary configuration file. See \fBtempnam\fR(3C).
.RE

.sp
.ne 2
.na
\fB\fB/usr/lib/lddstub\fR\fR
.ad
.sp .6
.RS 4n
Stub application that is employed to \fBdldump\fR(3C) 32-bit objects.
.RE

.sp
.ne 2
.na
\fB\fB/usr/lib/64/lddstub\fR\fR
.ad
.sp .6
.RS 4n
Stub application that is employed to \fBdldump\fR(3C) 64-bit objects.
.RE

.sp
.ne 2
.na
\fB\fB/usr/lib/libcrle.so.1\fR\fR
.ad
.sp .6
.RS 4n
Audit library that is employed to \fBdldump\fR(3C) 32-bit objects.
.RE

.sp
.ne 2
.na
\fB\fB/usr/lib/64/libcrle.so.1\fR\fR
.ad
.sp .6
.RS 4n
Audit library that is employed to \fBdldump\fR(3C) 64-bit objects.
```

```
914 .RE

916 .SH ENVIRONMENT VARIABLES
922 .sp
917 .LP
918 There are no environment variables that are referenced by \fBcrle\fR. However,
919 several environment variables affect the runtime linkers behavior in regard to
920 the processing of configuration files that are created by \fBcrle\fR.
921 .sp
922 .ne 2
923 .na
924 \fB\fBLD_CONFIG\fR, \fBLD_CONFIG_32\fR and \fBLD_CONFIG_64\fR\fR
925 .ad
926 .sp .6
927 .RS 4n
928 Provide an alternative configuration file.
929 .RE

931 .sp
932 .ne 2
933 .na
934 \fB\fBLD_NOCONFIG\fR, \fBLD_NOCONFIG_32\fR and \fBLD_NOCONFIG_64\fR\fR
935 .ad
936 .sp .6
937 .RS 4n
938 Disable configuration file processing.
939 .RE

941 .sp
942 .ne 2
943 .na
944 \fB\fBLD_NODIRCONFIG\fR, \fBLD_NODIRCONFIG_32\fR and \fBLD_NODIRCONFIG_64\fR\fR
945 .ad
946 .sp .6
947 .RS 4n
948 Disable directory cache processing from a configuration file.
949 .RE

951 .sp
952 .ne 2
953 .na
954 \fB\fBLD_NOENVCONFIG\fR, \fBLD_NOENVCONFIG_32\fR and \fBLD_NOENVCONFIG_64\fR\fR
955 .ad
956 .sp .6
957 .RS 4n
958 Disable environment variable processing from a configuration file.
959 .RE

961 .sp
962 .ne 2
963 .na
964 \fB\fBLD_NOOBJALTER\fR, \fBLD_NOOBJALTER_32\fR and \fBLD_NOOBJALTER_64\fR\fR
965 .ad
966 .sp .6
967 .RS 4n
968 Disable alternative object processing from a configuration file.
969 .RE

971 .SH ATTRIBUTES
978 .sp
972 .LP
973 See \fBattributes\fR(5) for descriptions of the following attributes.
974 .sp

976 .sp
977 .TS
```



```
978 box;
979 c | c
980 l | l .
981 ATTRIBUTE TYPE  ATTRIBUTE VALUE
982 _
983 Interface Stability     Committed
984 .TE

986 .SH SEE ALSO
994 .sp
987 .LP
988 \fBfile\fR(1), \fBld\fR(1), \fBld.so.1\fR(1), \fBdldump\fR(3C),
989 \fBtempnam\fR(3C), \fBattributes\fR(5)
990 .sp
991 .LP
992 \fILinker and Libraries Guide\fR
```

```
**********************************************************
   24019 Sun Sep 16 19:22:52 2018
new/usr/src/man/man1/find.1
9842 man page typos and spelling
**********************************************************
```

```
.\"
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for
.\" permission to reproduce portions of its copyrighted documentation.
.\" Original documentation from The Open Group can be obtained online at
.\" http://www.opengroup.org/bookstore/.
.\"
.\" The Institute of Electrical and Electronics Engineers and The Open
.\" Group, have given us permission to reprint portions of their
.\" documentation.
.\"
.\" In the following statement, the phrase ‘‘this text’’ refers to portions
.\" of the system documentation.
.\"
.\" Portions of this text are reprinted and reproduced in electronic form
.\" in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition,
.\" Standard for Information Technology -- Portable Operating System
.\" Interface (POSIX), The Open Group Base Specifications Issue 6,
.\" Copyright (C) 2001-2004 by the Institute of Electrical and Electronics
.\" Engineers, Inc and The Open Group.  In the event of any discrepancy
.\" between these versions and the original IEEE and The Open Group
.\" Standard, the original IEEE and The Open Group Standard is the referee
.\" document.  The original Standard can be obtained online at
.\" http://www.opengroup.org/unix/online.html.
.\"
.\" This notice shall appear on any product containing this material.
.\"
.\" The contents of this file are subject to the terms of the
.\" Common Development and Distribution License (the "License").
.\" You may not use this file except in compliance with the License.
.\"
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
.\" or http://www.opensolaris.org/os/licensing.
.\" See the License for the specific language governing permissions
.\" and limitations under the License.
.\"
.\" When distributing Covered Code, include this CDDL HEADER in each
.\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
.\" If applicable, add the following below this CDDL HEADER, with the
.\" fields enclosed by brackets "[]" replaced with your own identifying
.\" information: Portions Copyright [yyyy] [name of copyright owner]
.\"
.\"
.\" Copyright 1989 AT&T
.\" Portions Copyright (c) 1992, X/Open Company Limited All Rights Reserved
.\" Copyright (c) 2008, Sun Microsystems, Inc. All Rights Reserved
.\" Copyright 2011 Nexenta Systems, Inc. All rights reserved.
.\" Copyright (c) 2013 Andrew Stormont.  All rights reserved.
.\"
.TH FIND 1 "Sep 5, 2011"
.SH NAME
find \- find files
.SH SYNOPSIS
.LP
.nf
\fB/usr/bin/find\fR [\fB-E\fR] [\fB-H\fR | \fB-L\fR] \fIpath\fR... \fIexpression
.fi

.LP
.nf
\fB/usr/xpg4/bin/find\fR [\fB-H\fR | \fB-L\fR] \fIpath\fR... \fIexpression\fR
.fi
```



```
.SH DESCRIPTION
.sp
.LP
The \fBfind\fR utility recursively descends the directory hierarchy for each
\fIpath\fR seeking files that match a Boolean \fIexpression\fR written in the
primaries specified below.
.sp
.LP
\fBfind\fR is able to descend to arbitrary depths in a file hierarchy and does
not fail due to path length limitations (unless a \fIpath\fR operand specified
by the application exceeds \fIPATH_MAX\fR requirements).
.sp
.LP
\fBfind\fR detects infinite loops; that is, entering a previously visited
directory that is an ancestor of the last file encountered.
.SH OPTIONS
.sp
.LP
The following options are supported:
.sp
.ne 2
.na
\fB\fB-E\fR\fR
.ad
.RS 6n
Interpret regular expressions followed by \fB-regex\fR and \fB-iregex\fR
primaries as extended regular expressions.
.RE

.sp
.ne 2
.na
\fB\fB-H\fR\fR
.ad
.RS 6n
Causes the file information and file type evaluated for each symbolic link
encountered on the command line to be those of the file referenced by the link,
and not the link itself. If the referenced file does not exist, the file
information and type is for the link itself. File information for all symbolic
links not on the command line is that of the link itself.
.RE

.sp
.ne 2
.na
\fB\fB-L\fR\fR
.ad
.RS 6n
Causes the file information and file type evaluated for each symbolic link to
be those of the file referenced by the link, and not the link itself. See
\fBNOTES\fR.
.RE

.sp
.LP
Specifying more than one of the mutually-exclusive options \fB-H\fR and
\fB-L\fR is not considered an error. The last option specified determines the
behavior of the utility.
.SH OPERANDS
.sp
.LP
The following operands are supported:
.sp
.ne 2
.na
```

```
\fB\fIpath\fR\fR
.ad
.RS 14n
A pathname of a starting point in the directory hierarchy.
.RE

.sp
.ne 2
.na
\fB\fIexpression\fR\fR
.ad
.RS 14n
The first argument that starts with a \fB\(mi\fR, or is a \fB!\fR or a \fB(\fR,
and all subsequent arguments are interpreted as an \fIexpression\fR made up of
the following primaries and operators. In the descriptions, wherever \fIn\fR is
used as a primary argument, it is interpreted as a decimal integer optionally
preceded by a plus (\fB+\fR) or minus (\fB\(mi\fR) sign, as follows:
.sp
.ne 2
.na
\fB+\fIn\fR\fR
.ad
.RS 6n
more than \fIn\fR
.RE

.sp
.ne 2
.na
\fB\fIn\fR\fR
.ad
.RS 6n
exactly \fIn\fR
.RE

.sp
.ne 2
.na
\fB-\fIn\fR\fR
.ad
.RS 6n
less than \fIn\fR
.RE

.RE

.SS "Expressions"
.sp
.LP
Valid expressions are:
.sp
.ne 2
.na
\fB\fB-acl\fR\fR
.ad
.RS 17n
True if the file have additional ACLs defined.
.RE

.sp
.ne 2
.na
\fB\fB-amin\fR \fIn\fR\fR
.ad
.RS 17n
File was last accessed \fIn\fR minutes ago.
```



```
.RE

.sp
.ne 2
.na
\fB\fB-atime\fR \fIn\fR\fR
.ad
.RS 17n
True if the file was accessed \fIn\fR days ago. The access time of directories
in \fIpath\fR is changed by \fBfind\fR itself.
.RE

.sp
.ne 2
.na
\fB\fB-cmin\fR \fIn\fR\fR
.ad
.RS 17n
File's status was last changed \fIn\fR minutes ago.
.RE

.sp
.ne 2
.na
\fB\fB-cpio\fR \fIdevice\fR\fR
.ad
.RS 17n
Always true. Writes the current file on \fIdevice\fR in \fBcpio\fR format
(5120-byte records).
.RE

.sp
.ne 2
.na
\fB\fB-ctime\fR \fIn\fR\fR
.ad
.RS 17n
True if the file's status was changed \fIn\fR days ago.
.RE

.sp
.ne 2
.na
\fB\fB-depth\fR\fR
.ad
.RS 17n
Always true. Causes descent of the directory hierarchy to be done so that all
entries in a directory are acted on before the directory itself. This can be
useful when \fBfind\fR is used with \fBcpio\fR(1) to transfer files that are
contained in directories without write permission.
.RE

.sp
.ne 2
.na
\fB\fB-exec\fR \fIcommand\fR\fR
.ad
.RS 17n
True if the executed command returns a zero value as exit status. The end of
command must be punctuated by an escaped semicolon (\fB;\fR). A command
argument \fB{}\fR is replaced by the current pathname. If the last argument to
\fB-exec\fR is \fB{}\fR and you specify \fB+\fR rather than the semicolon
(\fB;\fR), the command is invoked fewer times, with \fB{}\fR replaced by groups
of pathnames. If any invocation of the command returns a non-zero value as exit
status, find returns a non-zero exit status.
.RE
```

```
257 .sp
258 .ne 2
259 .na
260 \fB\fB-follow\fR\fR
261 .ad
262 .RS 17n
263 Always true and always evaluated no matter where it appears in
264 \fIexpression\fR. The behavior is unspecified if \fB-follow\fR is used when the
265 \fBfind\fR command is invoked with either the \fB-H\fR or the \fB-L\fR option.
266 Causes symbolic links to be followed. When following symbolic links, \fBfind\fR
267 keeps track of the directories visited so that it can detect infinite loops.
268 For example, such a loop would occur if a symbolic link pointed to an ancestor.
269 This expression should not be used with the find-type \fBl\fR expression. See
270 \fBNOTES\fR.
271 .RE

273 .sp
274 .ne 2
275 .na
276 \fB\fB-fstype\fR \fItype\fR\fR
277 .ad
278 .RS 17n
279 True if the filesystem to which the file belongs is of type \fItype\fR.
280 .RE

282 .sp
283 .ne 2
284 .na
285 \fB\fB-group\fR \fIgname\fR\fR
286 .ad
287 .RS 17n
288 True if the file belongs to the group \fIgname\fR. If \fIgname\fR is numeric
289 and there's no such group name, it is taken as a group \fBID\fR.
290 .RE

292 .sp
293 .ne 2
294 .na
295 \fB\fB-groupacl\fR \fIgname\fR\fR
296 .ad
297 .RS 17n
298 True if the file's ACL contains an entry for the group \fIgname\fR.
299 If \fIgname\fR is numeric and there's no such group name, it is taken
300 as a group \fBID\fR.
301 .RE

303 .sp
304 .ne 2
305 .na
306 \fB\fB-iname\fR \fIpattern\fR\fR
307 .ad
308 .RS 17n
309 Like \fB-name\fR, but the match is case insensitive.
310 .RE

312 .sp
313 .ne 2
314 .na
315 \fB\fB-inum\fR \fIn\fR\fR
316 .ad
317 .RS 17n
318 True if the file has inode number \fIn\fR.
319 .RE

321 .sp
```



```
322 .ne 2
323 .na
324 \fB\fB-ipath\fR \fIpattern\fR\fR
325 .ad
326 .RS 17n
327 Like \fB-path\fR, but the match is case insensitive.
328 .RE

330 .sp
331 .ne 2
332 .na
333 \fB\fB-iregex\fR \fIpattern\fR\fR
334 .ad
335 .RS 17n
336 Like \fB-regex\fR, but the match is case insensitive.
337 .RE

339 .sp
340 .ne 2
341 .na
342 \fB\fB-links\fR \fIn\fR\fR
343 .ad
344 .RS 17n
345 True if the file has \fIn\fR links.
346 .RE

348 .sp
349 .ne 2
350 .na
351 \fB\fB-local\fR\fR
352 .ad
353 .RS 17n
354 True if the file system type is not a remote file system type as defined in the
355 \fB/etc/dfs/fstypes\fR file. \fBnfs\fR is used as the default remote filesystem
356 type if the \fB/etc/dfs/fstypes\fR file is not present. The \fB-local\fR option
357 descends the hierarchy of non-local directories. See \fBEXAMPLES\fR for an
358 example of how to search for local files without descending.
359 .RE

361 .sp
362 .ne 2
363 .na
364 \fB\fB-ls\fR\fR
365 .ad
366 .RS 17n
367 Always true. Prints current pathname together with its associated statistics.
368 These include (respectively):
369 .RS +4
370 .TP
371 .ie t \(bu
372 .el o
373 inode number
374 .RE
375 .RS +4
376 .TP
377 .ie t \(bu
378 .el o
379 size in kilobytes (1024 bytes)
380 .RE
381 .RS +4
382 .TP
383 .ie t \(bu
384 .el o
385 protection mode
386 .RE
387 .RS +4
```

```
.TP
.ie t \(bu
.el o
number of hard links
.RE
.RS +4
.TP
.ie t \(bu
.el o
user
.RE
.RS +4
.TP
.ie t \(bu
.el o
group
.RE
.RS +4
.TP
.ie t \(bu
.el o
size in bytes
.RE
.RS +4
.TP
.ie t \(bu
.el o
modification time.
.RE
If the file is a special file, the size field instead contains the major and
minor device numbers.
.sp
If the file is a symbolic link, the pathname of the linked-to file is printed
preceded by ‘\fB\(->\fR\&’. The format is identical to that of \fBls\fR
\fB-gilds\fR (see \fBls\fR(1B)).
.sp
Formatting is done internally, without executing the \fBls\fR program.
.RE

.sp
.ne 2
.na
\fB\fB-maxdepth\fR \fIn\fR\fR
.ad
.RS 17n
Always true; descend at most \fIn\fR directory levels below the command
line arguments. If any \fB-maxdepth\fR primary is specified, it
applies to the entire expression even if it would not normally be
evaluated. \fB-maxdepth 0\fR limits the whole search to
the command line arguments.
.RE

.sp
.ne 2
.na
\fB\fB-mindepth\fR \fIn\fR\fR
.ad
.RS 17n
Always true; do not apply any tests or actions at levels less
than \fIn\fR. If any \fB-mindepth\fR primary is specified, it applies to the
entire expression even if it would not normally be evaluated.
\fB-mindepth 1\fR processes all but the command line arguments.
.RE

.sp
.ne 2
```



```
.na
\fB\fB-mmin\fR \fIn\fR\fR
.ad
.RS 17n
File’s data was last modified \fIn\fR minutes ago.
.RE

.sp
.ne 2
.na
\fB\fB-mount\fR\fR
.ad
.RS 17n
Always true. Restricts the search to the file system containing the directory
specified. Does not list mount points to other file systems.
.RE

.sp
.ne 2
.na
\fB\fB-mtime\fR \fIn\fR\fR
.ad
.RS 17n
True if the file’s data was modified \fIn\fR days ago.
.RE

.sp
.ne 2
.na
\fB\fB-name\fR \fIpattern\fR\fR
.ad
.RS 17n
True if \fIpattern\fR matches the basename of the current file name. Normal
shell file name generation characters (see \fBsh\fR(1)) can be used. A
backslash (\fB\|\e\|\fR) is used as an escape character within the pattern. The
pattern should be escaped or quoted when \fBfind\fR is invoked from the shell.
.sp
Unless the character ’\fB\&.\fR’ is explicitly specified in the beginning of
\fIpattern\fR, a current file name beginning with ’\fB\&.\fR’ does not match
\fIpattern\fR when using \fB/usr/bin/find\fR. \fB/usr/xpg4/bin/find\fR does not
make this distinction; wildcard file name generation characters can match file
names beginning with ’\fB\&.\fR’.
.RE

.sp
.ne 2
.na
\fB\fB-ncpio\fR \fIdevice\fR\fR
.ad
.RS 17n
Always true. Writes the current file on \fIdevice\fR in \fBcpio\fR \fB-c\fR
format (5120 byte records).
.RE

.sp
.ne 2
.na
\fB\fB-newer\fR \fIfile\fR\fR
.ad
.RS 17n
True if the current file has been modified more recently than the argument
\fIfile\fR.
.RE

.sp
.ne 2
```

```
.na
\fB\fB-nogroup\fR\fR
.ad
.RS 17n
True if the file belongs to non-existing group.
.RE

.sp
.ne 2
.na
\fB\fB-nouser\fR\fR
.ad
.RS 17n
True if the file belongs to non-existing user.
.RE

.sp
.ne 2
.na
\fB\fB-ok\fR \fIcommand\fR\fR
.ad
.RS 17n
Like \fB-exec\fR, except that the generated command line is printed with a
question mark first, and is executed only if the response is affirmative.
.RE

.sp
.ne 2
.na
\fB\fB-path\fR\fR
.ad
.RS 17n
Like \fB-name\fR, but matches the entire file path and not just basename.
.RE

.sp
.ne 2
.na
\fB\fB-perm\fR [\fB-\fR]\fImode\fR\fR
.ad
.RS 17n
The \fImode\fR argument is used to represent file mode bits. It is identical in
format to the symbolic mode operand, \fIsymbolic_mode_list\fR, described in
\fBchmod\fR(1), and is interpreted as follows. To start, a template is assumed
with all file mode bits cleared. An \fIop\fR symbol of:
.sp
.ne 2
.na
\fB\fB+\fR\fR
.ad
.RS 8n
Set the appropriate mode bits in the template
.RE

.sp
.ne 2
.na
\fB\fB\(mi\fR\fR
.ad
.RS 8n
Clear the appropriate bits
.RE

.sp
.ne 2
.na
```



```
\fB\fB=\fR\fR
.ad
.RS 8n
Set the appropriate mode bits, without regard to the contents of the file mode
creation mask of the process
.RE

The \fIop\fR symbol of \fB\(mi\fR cannot be the first character of \fImode\fR,
to avoid ambiguity with the optional leading hyphen. Since the initial mode is
all bits off, there are no symbolic modes that need to use \fB\(mi\fR as the
first character.
.sp
If the hyphen is omitted, the primary evaluates as true when the file
permission bits exactly match the value of the resulting template.
.sp
Otherwise, if \fImode\fR is prefixed by a hyphen, the primary evaluates as true
if at least all the bits in the resulting template are set in the file
permission bits.
.RE

.sp
.ne 2
.na
\fB\fB-perm\fR [\fB-\fR]\fIonum\fR\fR
.ad
.RS 17n
True if the file permission flags exactly match the octal number \fIonum\fR
(see \fBchmod\fR(1)). If \fIonum\fR is prefixed by a minus sign (\fB\(mi\fR),
only the bits that are set in \fIonum\fR are compared with the file permission
flags, and the expression evaluates true if they match.
.RE

.sp
.ne 2
.na
\fB\fB-print\fR\fR
.ad
.RS 17n
Always true. Causes the current pathname to be printed.
.RE

.sp
.ne 2
.na
\fB\fB-print0\fR\fR
.ad
.RS 17n
Always true. Causes the current pathname to be printed, terminated by an ASCII
NUL character (character code 0) instead of a newline.
.RE

.sp
.ne 2
.na
\fB\fB-prune\fR\fR
.ad
.RS 17n
Always yields true. Does not examine any directories or files in the directory
structure below the \fIpattern\fR just matched. (See EXAMPLES). If \fB-depth\fR
is specified, \fB-prune\fR has no effect.
.RE

.sp
.ne 2
.na
\fB\fB-regex\fR \fIpattern\fR\fB
```

```
652 .ad
653 .RS 17n
654 True if the full path of the file matches \fIpattern\fR using regular
655 expressions.
656 .RE

658 .sp
659 .ne 2
660 .na
661 \fB\fB-size\fR \fIn\fR[\fBc\fR]\fR
662 .ad
663 .RS 17n
664 True if the file is \fIn\fR blocks long (512 bytes per block). If \fIn\fR is
665 followed by a \fBc\fR, the size is in bytes.
666 .RE

668 .sp
669 .ne 2
670 .na
671 \fB\fB-type\fR \fIc\fR\fR
672 .ad
673 .RS 17n
674 True if the type of the file is \fIc\fR, where \fIc\fR is \fBb\fR, \fBc\fR,
675 \fBd\fR, \fBD\fR, \fBf\fR, \fBl\fR, \fBp\fR, or \fBs\fR for block special file,
676 character special file, directory, door, plain file, symbolic link, fifo (named
677 pipe), or socket, respectively.
678 .RE

680 .sp
681 .ne 2
682 .na
683 \fB\fB-user\fR \fIuname\fR\fR
684 .ad
685 .RS 17n
686 True if the file belongs to the user \fIuname\fR. If \fIuname\fR is numeric and
687 there's no such user name, it is taken as a user \fBID\fR.
688 .RE

690 .sp
691 .ne 2
692 .na
693 \fB\fB-useracl\fR \fIuname\fR\fR
694 .ad
695 .RS 17n
696 True if the file's ACL contains an entry for the user \fIuname\fR.
697 If \fIuname\fR is numeric and there's no such user name, it is
698 taken as a user \fBID\fR.
699 .RE

701 .sp
702 .ne 2
703 .na
704 \fB\fB-xdev\fR\fR
705 .ad
706 .RS 17n
707 Same as the \fB-mount\fR primary.
708 .RE

710 .sp
711 .ne 2
712 .na
713 \fB\fB-xattr\fR\fR
714 .ad
715 .RS 17n
716 True if the file has extended attributes.
717 .RE
```



```
719 .SS "Complex Expressions"
724 .sp
720 .LP
721 The primaries can be combined using the following operators (in order of
722 decreasing precedence):
723 .sp
724 .ne 2
725 .na
726 \fB1)\fB(\fR\fIexpression\fR\fB)\fR\fR
727 .ad
728 .sp .6
729 .RS 4n
730 True if the parenthesized expression is true (parentheses are special to the
731 shell and must be escaped).
732 .RE

734 .sp
735 .ne 2
736 .na
737 \fB2)\fB!\fR\fIexpression\fR\fR
738 .ad
739 .sp .6
740 .RS 4n
741 The negation of a primary (\fB!\fR is the unary \fInot\fR operator).
742 .RE

744 .sp
745 .ne 2
746 .na
747 \fB3) \fIexpression\fR\fB[\fR\fB-a\fR\fB]\fR \fIexpression\fR\fR
748 .ad
749 .sp .6
750 .RS 4n
751 Concatenation of primaries (the \fIand\fR operation is implied by the
752 juxtaposition of two primaries).
753 .RE

755 .sp
756 .ne 2
757 .na
758 \fB4) \fIexpression\fR\fB\fR\fB-o\fR\fIexpression\fR\fR
759 .ad
760 .sp .6
761 .RS 4n
762 Alternation of primaries (\fB-o\fR is the \fIor\fR operator).
763 .RE

765 .sp
766 .LP
767 When you use \fBfind\fR in conjunction with \fBcpio\fR, if you use the \fB-L\fR
768 option with \fBcpio\fR, you must use the \fB-L\fR option or the \fB-follow\fR
769 primitive with \fBfind\fR and vice versa. Otherwise the results are
770 unspecified.
771 .sp
772 .LP
773 If no \fIexpression\fR is present, \fB-print\fR is used as the expression.
774 Otherwise, if the specified expression does not contain any of the primaries
775 \fB-exec\fR, \fB-ok\fR, \fB-ls\fR, or \fB-print\fR, the specified expression is
776 effectively replaced by:
777 .sp
778 .LP
779 (\fIspecified\fR) \fB-print\fR
780 .sp
781 .LP
782 The \fB-user\fR, \fB-group\fR, and \fB-newer\fR primaries each evaluate their
```

```
783 respective arguments only once. Invocation of \fIcommand\fR specified by
784 \fB-exec\fR or \fB-ok\fR does not affect subsequent primaries on the same file.
785 .SH USAGE
791 .sp
786 .LP
787 See \fBlargefile\fR(5) for the description of the behavior of \fBfind\fR when
788 encountering files greater than or equal to 2 Gbyte (2^31 bytes).
789 .SH EXAMPLES
790 .LP
791 \fBExample 1 \fRWriting Out the Hierarchy Directory
792 .sp
793 .LP
794 The following commands are equivalent:

796 .sp
797 .in +2
798 .nf
799 example% \fBfind .\fR
800 example% \fBfind . -print\fR
801 .fi
802 .in -2
803 .sp

805 .sp
806 .LP
807 They both write out the entire directory hierarchy from the current directory.

809 .LP
810 \fBExample 2 \fRRemoving Files
811 .sp
812 .LP
813 The following command removes all files in your home directory named \fBa.out\fR
819 The following comand removes all files in your home directory named \fBa.out\fR
814 or \fB*.o\fR that have not been accessed for a week:

816 .sp
817 .in +2
818 .nf
819 example% \fBfind $HOME \e( -name a.out -o -name '*.o' \e) \e
820         -atime +7 -exec rm {} \e;\fR
821 .fi
822 .in -2
823 .sp

825 .LP
826 \fBExample 3 \fRPrinting All File Names But Skipping SCCS Directories
827 .sp
828 .LP
829 The following command recursively print all file names in the current directory
830 and below, but skipping \fBSCCS\fR directories:

832 .sp
833 .in +2
834 .nf
835 example% \fBfind . -name SCCS -prune -o -print\fR
836 .fi
837 .in -2
838 .sp

840 .LP
841 \fBExample 4 \fRPrinting all file names and the SCCS directory name
842 .sp
843 .LP
844 Recursively print all file names in the current directory and below, skipping
845 the contents of \fBSCCS\fR directories, but printing out the \fBSCCS\fR
846 directory name:
```



```
848 .sp
849 .in +2
850 .nf
851 example% \fBfind . -print -name SCCS -prune\fR
852 .fi
853 .in -2
854 .sp

856 .LP
857 \fBExample 5 \fRTesting for the Newer File
858 .sp
859 .LP
860 The following command is basically equivalent to the \fB-nt\fR extension to
861 \fBtest\fR(1):

863 .sp
864 .in +2
865 .nf
866 example$ \fBif [ -n "$(find
867 file1 -prune -newer file2)" ]; then

869 printf %s\e\en "file1 is newer than file2"\fR
870 .fi
871 .in -2
872 .sp

874 .LP
875 \fBExample 6 \fRSelecting a File Using 24-hour Mode
876 .sp
877 .LP
878 The descriptions of \fB-atime\fR, \fB-ctime\fR, and \fB-mtime\fR use the
879 terminology \fIn\fR ''24-hour periods''. For example, a file accessed at 23:59
880 is selected by:

882 .sp
883 .in +2
884 .nf
885 example% \fBfind . -atime -1 -print\fR
886 .fi
887 .in -2
888 .sp

890 .sp
891 .LP
892 at 00:01 the next day (less than 24 hours later, not more than one day ago).
893 The midnight boundary between days has no effect on the 24-hour calculation.

895 .LP
896 \fBExample 7 \fRPrinting Files Matching a User's Permission Mode
897 .sp
898 .LP
899 The following command recursively print all file names whose permission mode
900 exactly matches read, write, and execute access for user, and read and execute
901 access for group and other:

903 .sp
904 .in +2
905 .nf
906 example% \fBfind . -perm u=rwx,g=rx,o=rx\fR
907 .fi
908 .in -2
909 .sp

911 .sp
912 .LP
```

```
The above could alternatively be specified as follows:

.sp
.in +2
.nf
example% \fBfind . -perm a=rwx,g-w,o-w\fR
.fi
.in -2
.sp

.LP
\fBExample 8 \fRPrinting Files with Write Access for \fBother\fR
.sp
.LP
The following command recursively print all file names whose permission
includes, but is not limited to, write access for other:

.sp
.in +2
.nf
example% \fBfind . -perm -o+w\fR
.fi
.in -2
.sp

.LP
\fBExample 9 \fRPrinting Local Files without Descending Non-local Directories
.sp
.in +2
.nf
example% \fBfind . ! -local -prune -o -print\fR
.fi
.in -2
.sp

.LP
\fBExample 10 \fRPrinting the Files in the Name Space Possessing Extended
Attributes
.sp
.in +2
.nf
example% \fBfind . -xattr\fR
.fi
.in -2
.sp

.SH ENVIRONMENT VARIABLES
.sp
.LP
See \fBenviron\fR(5) for descriptions of the following environment variables
that affect the execution of \fBfind\fR: \fBLANG\fR, \fBLC_ALL\fR,
\fBLC_COLLATE\fR, \fBLC_CTYPE\fR, \fBLC_MESSAGES\fR, and \fBNLSPATH\fR.
.sp
.ne 2
.na
\fB\fBPATH\fR\fR
.ad
.RS 8n
Determine the location of the \fIutility_name\fR for the \fB-exec\fR and
\fB-ok\fR primaries.
.RE

.sp
.LP
Affirmative responses are processed using the extended regular expression
defined for the \fByesexpr\fR keyword in the \fBLC_MESSAGES\fR category of the
```



```
user's locale. The locale specified in the \fBLC_COLLATE\fR category defines
the behavior of ranges, equivalence classes, and multi-character collating
elements used in the expression defined for \fByesexpr\fR. The locale specified
in \fBLC_CTYPE\fR determines the locale for interpretation of sequences of
bytes of text data a characters, the behavior of character classes used in the
expression defined for the \fByesexpr\fR. See \fBlocale\fR(5).
.SH EXIT STATUS
.sp
.LP
The following exit values are returned:
.sp
.ne 2
.na
\fB\fB0\fR\fR
.ad
.RS 6n
All \fIpath\fR operands were traversed successfully.
.RE

.sp
.ne 2
.na
\fB\fB>0\fR\fR
.ad
.RS 6n
An error occurred.
.RE

.SH FILES
.sp
.ne 2
.na
\fB\fB/etc/passwd\fR\fR
.ad
.RS 20n
Password file
.RE

.sp
.ne 2
.na
\fB\fB/etc/group\fR\fR
.ad
.RS 20n
Group file
.RE

.sp
.ne 2
.na
\fB\fB/etc/dfs/fstypes\fR\fR
.ad
.RS 20n
File that registers distributed file system packages
.RE

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
```

```
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
CSI     Enabled
_
Interface Stability     Committed
_
Standard        See \fBstandards\fR(5).
.TE

.SH SEE ALSO
.sp
.LP
\fBchmod\fR(1), \fBcpio\fR(1), \fBsh\fR(1), \fBtest\fR(1), \fBls\fR(1B),
\fBacl\fR(5), \fBregex\fR(5), \fBstat\fR(2), \fBumask\fR(2),
\fBattributes\fR(5), \fBenviron\fR(5), \fBfsattr\fR(5), \fBlargefile\fR(5),
\fBlocale\fR(5), \fBstandards\fR(5)
.SH WARNINGS
.sp
.LP
The following options are obsolete and will not be supported in future
releases:
.sp
.ne 2
.na
\fB\fB-cpio\fR \fIdevice\fR\fR
.ad
.RS 17n
Always true. Writes the current file on \fIdevice\fR in \fBcpio\fR format
(5120-byte records).
.RE

.sp
.ne 2
.na
\fB\fB-ncpio\fR \fIdevice\fR\fR
.ad
.RS 17n
Always true. Writes the current file on \fIdevice\fR in \fBcpio\fR \fB-c\fR
format (5120-byte records).
.RE

.SH NOTES
.sp
.LP
When using \fBfind\fR to determine files modified within a range of time, use
the \fB-mtime\fR argument \fBbefore\fR the \fB-print\fR argument. Otherwise,
\fBfind\fR gives all files.
.sp
.LP
Some files that might be under the Solaris root file system are actually mount
points for virtual file systems, such as \fBmntfs\fR or \fBnamefs\fR. When
comparing against a \fBufs\fR file system, such files are not selected if
\fB-mount\fR or \fB-xdev\fR is specified in the \fBfind\fR expression.
.sp
.LP
Using the \fB-L\fR or \fB-follow\fR option is not recommended when descending a
file-system hierarchy that is under the control of other users. In particular,
when using \fB-exec\fR, symbolic links can lead the \fBfind\fR command out of
the hierarchy in which it started. Using \fB-type\fR is not sufficient to
restrict the type of files on which the \fB-exec\fR command operates, because
there is an inherent race condition between the type-check performed by the
\fBfind\fR command and the time the executed command operates on the file
argument.
```

```
**********************************************************
   12198 Sun Sep 16 19:22:52 2018
new/usr/src/man/man1/lgrpinfo.1
9842 man page typos and spelling
**********************************************************
'\" te
.\" Copyright (c) 2006, Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH LGRPINFO 1 "April 9, 2016"
.SH NAME
lgrpinfo \- display information about locality groups
.SH SYNOPSIS
.LP
.nf
\fBlgrpinfo\fR [\fB-aceGlLmrt\fR] [\fB-u \fR\fIunit\fR] [\fB-C\fR | \fB-P\fR] \f
.fi

.LP
.nf
\fBlgrpinfo\fR \fB-h\fR
.fi

.LP
.nf
\fBlgrpinfo\fR \fB-I\fR [\fB-c\fR] [\fB-G\fR] [\fB-C\fR | \fB-P\fR] \fIlgrp\fR .
.fi

.LP
.nf
\fBlgrpinfo\fR [\fB-T\fR] [\fB-aceGlLmr\fR] [\fB-u \fR\fIunit\fR]
.fi

.SH DESCRIPTION
.LP
\fBlgrpinfo\fR prints information about the locality group (\fBlgroup\fR)
hierarchy and its contents.
.sp
.LP
An lgroup represents the set of CPU and memory-like hardware devices that are
at most some distance (latency) apart from each other. All lgroups in the
system are identified by a unique integer called an \fBlgroup ID\fR.
.sp
.LP
lgroups are organized into a hierarchy to facilitate finding the nearest
resources. Leaf lgroups each contain a set of resources that are closest
(local) to each other. Each parent lgroup in the hierarchy contains the
resources of its child lgroups plus their next nearest resources. Finally, the
\fBroot\fR lgroup contains all the resources in the domain within the largest
latency.
.sp
.LP
A Uniform Memory Access (UMA) machine is simply represented by the root lgroup.
A Non Uniform Memory Access (NUMA) machine is represented by a hierarchy of
lgroups to show the corresponding levels of locality. For example, a NUMA
machine with two latencies (local and remote) has an \fBlgroup\fR hierarchy
consisting of two levels with its leaves and the root.
.sp
.LP
Every application thread is assigned a \fBhome\fR lgroup. When the system needs
to allocate a CPU or memory resource for a thread, it searches lgroup hierarchy
from the thread's home lgroup for the closest available resources to the
thread's home. See \fBplgrp\fR(1) for details.
.sp
.LP
```



```
Without arguments, \fBlgrpinfo\fR prints general information about all lgroups
in the system. If any lgroup IDs are specified on the command line, the command
only prints information about the specified lgroups. Various options control
which lgroups are displayed and the exact information that is printed for each
lgroup.
.sp
.LP
lgroups can be specified on the command line as lgroup IDs or by using specific
keywords. See \fBOPERANDS\fR.
.SH OPTIONS
.LP
You can combine options together and the order in which options are specified
is not important. Lowercase options select what information should be printed
about lgroups.
.sp
.LP
Invoking \fBlgrpinfo\fR without arguments is equivalent to:
.sp
.in +2
.nf
lgrpinfo -c -e -l -m -r -t all
.fi
.in -2
.sp

.sp
.LP
The following options are supported:
.sp
.ne 2
.na
\fB\fB-a\fR\fR
.ad
.RS 12n
Print topology, CPU, memory, load and latency information.
.sp
This option is a shorthand for
.sp
.in +2
.nf
lgrpinfo -t -c -e -m -r -l -L
.fi
.in -2
.sp

unless \fB-T\fR is specified as well. When \fB-T\fR is specified, the \fB-t\fR
option is not included.
.RE

.sp
.ne 2
.na
\fB\fB-c\fR\fR
.ad
.RS 12n
Print CPU information.
.sp
This is the default.
.RE

.sp
.ne 2
.na
\fB\fB-C\fR\fR
.ad
.RS 12n
```

```
128 Replace each lgroup in the list with its children.
129 .sp
130 This option cannot be used with the \fB-P\fR or the \fB-T\fR option. When no
131 arguments are specified, this option is applied to the lgroups displayed by
132 default.
133 .RE

135 .sp
136 .ne 2
137 .na
138 \fB\fB-e\fR\fR
139 .ad
140 .RS 12n
141 Print lgroup load average. The lgroup load averages are only displayed for leaf
142 lgroups.
143 .sp
144 This is the default.
145 .RE

147 .sp
148 .ne 2
149 .na
150 \fB\fB-G\fR\fR
151 .ad
152 .RS 12n
153 Print OS view of lgroup hierarchy.
154 .sp
155 By default, the caller's view of the lgroup hierarchy is displayed which only
156 includes what the caller can use, for example, only the CPUs in the caller's
157 processor set is displayed. See \fBlgrp_init\fR(3LGRP) on the operating system
158 and the caller's view.
159 .RE

161 .sp
162 .ne 2
163 .na
164 \fB\fB-h\fR\fR
165 .ad
166 .RS 12n
167 Print short help message and exit.
168 .RE

170 .sp
171 .ne 2
172 .na
173 \fB\fB-I\fR\fR
174 .ad
175 .RS 12n
176 Print matching IDs only.
177 .sp
178 This option is intended for scripts and can be used with \fB-c\fR, \fB-G\fR,
179 and \fB-C\fR or \fB-P\fR. If \fB-c\fR is specified, print list of CPUs
180 contained in all matching lgroups. Otherwise, the IDs for the matching lgroups
181 is displayed. See \fBEXAMPLES\fR.
182 .sp
183 When no arguments are specified, this option is applied to the lgroups
184 displayed, which, by default is all lgroups.
185 .RE

187 .sp
188 .ne 2
189 .na
190 \fB\fB-l\fR\fR
191 .ad
192 .RS 12n
193 Print information about lgroup latencies.
```



```
194 .sp
195 The latency value specified for each lgroup is defined by the operating system
196 and is platform-specific. It can only be used for relative comparison of
197 lgroups on the running system. It does not necessarily represent the actual
198 latency between hardware devices and might not be applicable across platforms.
199 .RE

201 .sp
202 .ne 2
203 .na
204 \fB\fB-L\fR\fR
205 .ad
206 .RS 12n
207 Print the lgroup latency table. The lgroup latency table displays the relative
208 latency from each lgroup to each of the other lgroups including itself.
209 .RE

211 .sp
212 .ne 2
213 .na
214 \fB\fB-m\fR\fR
215 .ad
216 .RS 12n
217 Print memory information.
218 .sp
219 Memory sizes are scaled to the unit of measure that yields an integer from
220 \fB0\fR to \fB1023\fR unless the \fB-u\fR option is specified as well. The
221 fractional part of the number is only displayed for values less than \fB10\fR.
222 This behavior is similar to using the \fB-h\fR option of \fBls\fR(1) or
223 \fBdf\fR(1M) to display a human readable format.
224 .sp
225 This is the default.
226 .RE

228 .sp
229 .ne 2
230 .na
231 \fB\fB-P\fR\fR
232 .ad
233 .RS 12n
234 Replace each lgroup in the list with its parents.
235 .sp
236 This option cannot be used with the \fB-C\fR or \fB-T\fR option. When no
237 arguments are specified, this option is applied to the lgroups displayed,
238 which, by default is all lgroups.
239 .RE

241 .sp
242 .ne 2
243 .na
244 \fB\fB-r\fR\fR
245 .ad
246 .RS 12n
247 Print information about lgroup resources.
248 .sp
249 The resources are represented by a set of lgroups in which each member lgroup
250 directly contains CPU and memory resources. If \fB-T\fR is specified as well,
251 only information about resources of the intermediate lgroups is displayed.
252 .RE

254 .sp
255 .ne 2
256 .na
257 \fB\fB-t\fR\fR
258 .ad
259 .RS 12n
```

```
260 Print information about lgroup topology.
261 .sp
262 This is the default.
263 .RE

265 .sp
266 .ne 2
267 .na
268 \fB\fB-T\fR\fR
269 .ad
270 .RS 12n
271 Print the lgroup topology of a system graphically as a tree. This option can
272 only be used with the \fB-a\fR, \fB-c\fR, \fB-e\fR, \fB-G\fR,
273 \fB-l\fR,\fB-L\fR, \fB-m\fR, \fB-r\fR, and \fB-u\fR options. It only prints
274 lgroup resources for intermediate lgroups when used with the \fB-r\fR. The
275 \fB-t\fR option is omitted when \fB-T\fR is used with \fB-a\fR. No information
276 is printed for the \fBroot\fR lgroup unless it is the only lgroup.
277 .RE

279 .sp
280 .ne 2
281 .na
282 \fB\fB-u\fR \fIunits\fR\fR
283 .ad
284 .RS 12n
285 Specify memory units. Units should be b, k, m, g, t, p, or e for bytes,
286 kilobytes, megabytes, gigabytes, terabytes, petabytes, or exabytes
287 respectively. The fractional part of the number is only displayed for values
288 less than 10. This behavior is similar to using the \fB-h\fR option of
289 \fBls\fR(1) or \fBdf\fR(1M) to display a human readable format.
290 .RE

292 .SH OPERANDS
293 .LP
294 The following operands are supported:
295 .sp
296 .ne 2
297 .na
298 \fB\fIlgrp\fR\fR
299 .ad
300 .RS 8n
301 lgroups can be specified on the command line as lgroup ID, by using one of the
302 following keywords:
303 .sp
304 .ne 2
305 .na
306 \fBall\fR
307 .ad
308 .RS 16n
309 All lgroups.
310 .sp
311 This is the default.
312 .RE

314 .sp
315 .ne 2
316 .na
317 \fBintermediate\fR
318 .ad
319 .RS 16n
320 All intermediate lgroups. An intermediate lgroup is an lgroup that has a parent
321 and children.
322 .RE

324 .sp
325 .ne 2
```



```
326 .na
327 \fBleaves\fR
328 .ad
329 .RS 16n
330 All leaf lgroups. A leaf lgroup is an lgroup that has no children in the lgroup
331 hierarchy.
332 .RE

334 .sp
335 .ne 2
336 .na
337 \fBroot\fR
338 .ad
339 .RS 16n
340 Root lgroup. Root lgroup contains all the resources in the domain within the
341 largest latency and has no parent lgroup.
342 .RE

344 .RE

346 .sp
347 .LP
348 If an invalid lgroup is specified, the lgrpinfo command prints a message on
349 standard error showing the invalid ID and continues processing other lgroups
350 specified on the command line. When none of the specified lgroups are valid,
351 \fBlgrpinfo\fR exits with an exit status of \fB2\fR.
352 .SH EXAMPLES
353 .LP
354 \fBExample 1 \fRPrinting Information about lgroups
355 .sp
356 .LP
357 The following example prints general information about lgroups in the system.

359 .sp
360 .LP
361 In this example, the system is a 2 CPU AMD Opteron machine with two nodes, each
362 having one CPU and 2 gigabytes of memory. Each of these nodes is represented by
363 a leaf lgroup. The root lgroup contains all the resources in the machine:

365 .sp
366 .in +2
367 .nf
368 $ lgrpinfo
369   lgroup 0 (root):
370           Children: 1 2
371           CPUs: 0 1
372           Memory: installed 4.0G, allocated 2.2G, free 1.8G
373           Lgroup resources: 1 2 (CPU); 1 2 (memory)
374           Latency: 83
375   lgroup 1 (leaf):
376           Children: none, Parent: 0
377           CPU: 0
378           Memory: installed 2.0G, allocated 1.2G, free 788M
379           Lgroup resources: 1 (CPU); 1 (memory)
380           Load: 0.793
381           Latency: 56
382   lgroup 2 (leaf):
383           Children: none, Parent: 0
384           CPU: 1
385           Memory: installed 2.0G, allocated 1017M, free 1.0G
386           Lgroup resources: 2 (CPU); 2 (memory)
387           Load: 0.817
388           Latency: 56
389 .fi
390 .in -2
391 .sp
```

```
.LP
\fBExample 2 \fRPrinting lgroup Topology
.sp
.LP
The following example prints the lgroup topology tree on a 4 CPU AMD Opteron
machine:

.sp
.in +2
.nf
$ lgrpinfo -T
  0
  |-- 5
  |   `-- 1
  |-- 6
  |   `-- 2
  |-- 7
  |   `-- 3
  `-- 8
      `-- 4
.fi
.in -2
.sp

.LP
\fBExample 3 \fRPrinting lgroup Topology
.sp
.LP
The following example prints the lgroup topology tree, resources, memory and
CPU information on a 2 CPU AMD Opteron machine:

.sp
.in +2
.nf
$ lgrpinfo -Ta
  0
  |-- 1
  |   CPU: 0
  |   Memory: installed 2.0G, allocated 1.2G, free 790M
  |   Load: 0.274
  |   Latency: 56
  `-- 2
      CPU: 1
      Memory: installed 2.0G, allocated 1019M, free 1.0G
      Load: 0.937
      Latency: 56

Lgroup latencies:

------------
   |  0  1  2
  ------------
  0 | 83 83 83
  1 | 83 56 83
  2 | 83 83 56
  ------------
.fi
.in -2
.sp

.LP
\fBExample 4 \fRPrinting lgroup IDs
.sp
.LP
The following example prints lgroup IDs for children of the root lgroup:
```



```
.sp
.in +2
.nf
$ lgrpinfo -I -C root
  1 2
.fi
.in -2
.sp

.LP
\fBExample 5 \fRPrinting CPU IDs
.sp
.LP
The following example prints CPU IDs for all CPUs in lgroup 1:

.sp
.in +2
.nf
$ lgrpinfo -c -I 1
  0
.fi
.in -2
.sp

.LP
\fBExample 6 \fRPrinting Information about lgroup Latencies
\fIExample 6 \fRPrinting Information about lgropu Latencies
.sp
.LP
The following example prints information about lgroup latencies:

.sp
.in +2
.nf
 $ lgrpinfo -l
  lgroup 0 (root):
          Latency: 83
  lgroup 1 (leaf):
          Latency: 56
  lgroup 2 (leaf):
          Latency: 5
.fi
.in -2
.sp

.SH EXIT STATUS
.LP
The following exit values are returned:
.sp
.ne 2
.na
\fB\fB0\fR\fR
.ad
.RS 5n
Successful completion.
.RE

.sp
.ne 2
.na
\fB\fB1\fR\fR
.ad
.RS 5n
Unable to get lgroup information from the system.
.RE
```

```
 524 .sp
 525 .ne 2
 526 .na
 527 \fB\fB2\fR\fR
 528 .ad
 529 .RS 5n
 530 All lgroups specified are invalid.
 531 .RE

 533 .sp
 534 .ne 2
 535 .na
 536 \fB\fB3\fR\fR
 537 .ad
 538 .RS 5n
 539 Invalid syntax.
 540 .RE

 542 .SH ATTRIBUTES
 543 .LP
 544 See \fBattributes\fR(5) for descriptions of the following attributes:
 545 .sp

 547 .sp
 548 .TS
 549 box;
 550 c | c
 551 l | l .
 552 ATTRIBUTE TYPE  ATTRIBUTE VALUE
 553 _
 554 Interface Stability     See below.
 555 .TE

 557 .sp
 558 .LP
 559 The human readable output is Unstable.
 560 .SH SEE ALSO
 561 .LP
 562 \fBls\fR(1), \fBplgrp\fR(1), \fBpmap\fR(1), \fBproc\fR(1), \fBps\fR(1),
 563 \fBdf\fR(1M), \fBprstat\fR(1M), \fBlgrp_init\fR(3LGRP), \fBliblgrp\fR(3LIB),
 564 \fBproc\fR(4), \fBattributes\fR(5)
```

```
**********************************************************
    9644 Sun Sep 16 19:22:52 2018
new/usr/src/man/man1/localedef.1
9842 man page typos and spelling
**********************************************************
```

```
.\"
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for
.\" permission to reproduce portions of its copyrighted documentation.
.\" Original documentation from The Open Group can be obtained online at
.\" http://www.opengroup.org/bookstore/.
.\"
.\" The Institute of Electrical and Electronics Engineers and The Open
.\" Group, have given us permission to reprint portions of their
.\" documentation.
.\"
.\" In the following statement, the phrase ‘‘this text’’ refers to portions
.\" of the system documentation.
.\"
.\" Portions of this text are reprinted and reproduced in electronic form
.\" in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition,
.\" Standard for Information Technology -- Portable Operating System
.\" Interface (POSIX), The Open Group Base Specifications Issue 6,
.\" Copyright (C) 2001-2004 by the Institute of Electrical and Electronics
.\" Engineers, Inc and The Open Group.  In the event of any discrepancy
.\" between these versions and the original IEEE and The Open Group
.\" Standard, the original IEEE and The Open Group Standard is the referee
.\" document.  The original Standard can be obtained online at
.\" http://www.opengroup.org/unix/online.html.
.\"
.\" This notice shall appear on any product containing this material.
.\"
.\" The contents of this file are subject to the terms of the
.\" Common Development and Distribution License (the "License").
.\" You may not use this file except in compliance with the License.
.\"
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
.\" or http://www.opensolaris.org/os/licensing.
.\" See the License for the specific language governing permissions
.\" and limitations under the License.
.\"
.\" When distributing Covered Code, include this CDDL HEADER in each
.\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
.\" If applicable, add the following below this CDDL HEADER, with the
.\" fields enclosed by brackets "[]" replaced with your own identifying
.\" information: Portions Copyright [yyyy] [name of copyright owner]
.\"
.\"
.\" Copyright (c) 1992, X/Open Company Limited  All Rights Reserved
.\" Portions Copyright (c) 2003, Sun Microsystems, Inc.  All Rights Reserved
.\" Portions Copyright 2013 DEY Storage Systems, Inc.
.\"
.TH LOCALEDEF 1 "April 9, 2016"
.SH NAME
localedef \- define locale environment
.SH SYNOPSIS
.LP
.nf
\fBlocaledef\fR [\fB-c\fR] [\fB-v\fR] [\fB-U\fR\] [\fB-f\fR \fIcharmap\fR]
     [\fB-w\fR \fIwidthfile\fR] [\fB-i\fR \fIsourcefile\fR]
     [\fB-u\fR \fIcode_set_name\fR] \fIlocalename\fR
.fi

.SH DESCRIPTION
.LP
The \fBlocaledef\fR utility converts source definitions for locale categories
into a format usable by the functions and utilities whose operational behavior
```



```
is determined by the setting of the locale environment variables; see
\fBenviron\fR(5).
.sp
.LP
The utility reads source definitions for one or more locale categories
belonging to the same locale from the file named in the \fB-i\fR option (if
specified) or from standard input.
.sp
.LP
Each category source definition is identified by the corresponding environment
variable name and terminated by an \fBEND\fR \fIcategory-name\fR statement. The
following categories are supported.
.sp
.ne 2
.na
\fB\fBLC_CTYPE\fR\fR
.ad
.RS 15n
Defines character classification and case conversion.
.RE

.sp
.ne 2
.na
\fB\fBLC_COLLATE\fR\fR
.ad
.RS 15n
Defines collation rules.
.RE

.sp
.ne 2
.na
\fB\fBLC_MONETARY\fR\fR
.ad
.RS 15n
Defines the format and symbols used in formatting of monetary information.
.RE

.sp
.ne 2
.na
\fB\fBLC_NUMERIC\fR\fR
.ad
.RS 15n
Defines the decimal delimiter, grouping and grouping symbol for non-monetary
numeric editing.
.RE

.sp
.ne 2
.na
\fB\fBLC_TIME\fR\fR
.ad
.RS 15n
Defines the format and content of date and time information.
.RE

.sp
.ne 2
.na
\fB\fBLC_MESSAGES\fR\fR
.ad
.RS 15n
Defines the format and values of affirmative and negative responses.
.RE
```

```
.SH OPTIONS
.LP
The following options are supported:
.sp
.ne 2
.na
\fB\fB-c\fR\fR
.ad
.RS 23n
Creates permanent output even if warning messages have been issued.
.RE

.sp
.ne 2
.na
\fB\fB-v\fR\fR
.ad
.RS 23n
Emit verbose debugging output on standard output.
.RE

.sp
.ne 2
.na
\fB\fB-U\fR\fR
.ad
.RS 23n
Ignore the presence of character symbols that have no matching character
definition.  This facilitates the use of a common locale definition file
to be used across multiple encodings, even when some symbols are not
present in a given encoding.
.sp
Support for this option is an illumos extension.
.RE

.sp
.ne 2
.na
\fB\fB-f\fR \fIcharmap\fR\fR
.ad
.RS 23n
Specifies the pathname of a file containing a mapping of character symbols and
collating element symbols to actual character encodings. This option must be
specified if symbolic names (other than collating symbols defined in a
\fBcollating-symbol\fR keyword) are used. If the \fB-f\fR option is not
present, the default character mapping will be used.
.RE

.sp
.ne 2
.na
\fB\fB-w\fR \fIwidthfile\fR\fR
.ad
.RS 23n
The path name of the file containing character screen width definitions.
If not supplied, then default screen widths will be assumed, which will
generally not account for East Asian encodings requiring more than a single
character cell to display, nor for combining or accent marks that occupy
no additional screen width.
.sp
The support for width files is an illumos extension.
.RE

.sp
.ne 2
```



```
.na
\fB\fB-i\fR \fIsourcefile\fR\fR
.ad
.RS 23n
The path name of a file containing the source definitions. If this option is
not present, source definitions will be read from standard input.
.RE

.sp
.ne 2
.na
\fB\fB-u\fR \fIcode_set_name\fR\fR
.ad
.RS 23n
Specifies the name of a codeset used as the target mapping of character symbols
and collating element symbols whose encoding values are defined in terms of the
ISO/IEC 10646-1: 2000 standard position constant values. See NOTES.
.RE

.SH OPERANDS
.LP
The following operand is supported:
.sp
.ne 2
.na
\fB\fIlocalename\fR\fR
.ad
.RS 14n
Identifies the locale. If the name contains one or more slash characters,
\fIlocalename\fR will be interpreted as a path name where the created locale
definitions will be stored. This capability may be restricted to users with
appropriate privileges. (As a consequence of specifying one \fIlocalename\fR,
although several categories can be processed in one execution, only categories
belonging to the same locale can be processed.)
.RE

.SH OUTPUT
.LP
\fBlocaledef\fR creates a directory of files that represents the locale's
data. The contents of this directory should generally be copied into the
data. The contants of this directory should generally be copied into the
appropriate subdirectory of /usr/lib/locale in order the definitions to
be visible to programs linked with libc.
.sp
.SH ENVIRONMENT VARIABLES
.LP
See \fBenviron\fR(5) for definitions of the following environment variables
that affect the execution of \fBlocaledef\fR: \fBLANG\fR, \fBLC_ALL\fR,
\fBLC_COLLATE\fR, \fBLC_CTYPE\fR, \fBLC_MESSAGES\fR, and \fBNLSPATH\fR.
.SH EXIT STATUS
.LP
The following exit values are returned:
.sp
.ne 2
.na
\fB\fB0\fR\fR
.ad
.RS 6n
No errors occurred and the locales were successfully created.
.RE

.sp
.ne 2
.na
\fB\fB1\fR\fR
.ad
```

```
.RS 6n
Warnings occurred and the locales were successfully created.
.RE

.sp
.ne 2
.na
\fB\fB2\fR\fR
.ad
.RS 6n
The locale specification exceeded implementation limits or the coded character
set or sets used were not supported by the implementation, and no locale was
created.
.RE

.sp
.ne 2
.na
\fB\fB>3\fR\fR
.ad
.RS 6n
Warnings or errors occurred and no output was created.
.RE

.sp
.LP
If an error is detected, no permanent output will be created.
.SH FILES
.ne 2
.na
\fB/usr/lib/locale/\fR\fIlocalename\fR\fB/\fR
.ad
.sp .6
.RS 4n
The directory containing locale data.
.RE

.SH ATTRIBUTES
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
Interface Stability     Standard
.TE

.SH SEE ALSO
.LP
\fBlocale\fR(1), \fBiconv_open\fR(3C), \fBnl_langinfo\fR(3C),
\fBstrftime\fR(3C), \fBattributes\fR(5), \fBcharmap\fR(5), \fBenviron\fR(5),
\fBextensions\fR(5), \fBlocale\fR(5), \fBstandards\fR(5)
.SH WARNINGS
.LP
If warnings occur, permanent output will be created if the \fB-c\fR option was
specified. The following conditions will cause warning messages to be issued:
.RS +4
.TP
.ie t \(bu
.el o
If a symbolic name not found in the \fIcharmap\fR file is used for the
```



```
descriptions of the \fBLC_CTYPE\fR or \fBLC_COLLATE\fR categories (for other
categories, this will be an error condition).
.RE
.RS +4
.TP
.ie t \(bu
.el o
If optional keywords not supported by the implementation are present in the
source.
.RE
.SH NOTES
.LP
When the \fB-u\fR option is used, the \fIcode_set_name\fR option-argument is
interpreted as a name of a codeset to which the ISO/IEC 10646-1: 2000 standard
position constant values are converted. Both the ISO/IEC 10646-1: 2000 standard
position constant values and other formats (decimal, hexadecimal, or octal) are
valid as encoding values within the charmap file. The codeset can be any
codeset that is supported by the \fBiconv_open\fR(3C) function on the system.
.sp
.LP
When conflicts occur between the charmap specification of \fIcode_set_name\fR,
\fImb_cur_max\fR, or \fImb_cur_min\fR and the corresponding value for the
codeset represented by the \fB-u\fR option-argument \fIcode_set_name\fR, the
\fBlocaledef\fR utility fails as an error.
.sp
.LP
When conflicts occur between the charmap encoding values specified for symbolic
names of characters of the portable character set and the character encoding
values defined by the US-ASCII, the result is unspecified.
```

```
**********************************************************
   21019 Sun Sep 16 19:22:53 2018
new/usr/src/man/man1/lp.1
9842 man page typos and spelling
**********************************************************
```

```
.\"
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for
.\" permission to reproduce portions of its copyrighted documentation.
.\" Original documentation from The Open Group can be obtained online at
.\" http://www.opengroup.org/bookstore/.
.\"
.\" The Institute of Electrical and Electronics Engineers and The Open
.\" Group, have given us permission to reprint portions of their
.\" documentation.
.\"
.\" In the following statement, the phrase ‘‘this text’’ refers to portions
.\" of the system documentation.
.\"
.\" Portions of this text are reprinted and reproduced in electronic form
.\" in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition,
.\" Standard for Information Technology -- Portable Operating System
.\" Interface (POSIX), The Open Group Base Specifications Issue 6,
.\" Copyright (C) 2001-2004 by the Institute of Electrical and Electronics
.\" Engineers, Inc and The Open Group.  In the event of any discrepancy
.\" between these versions and the original IEEE and The Open Group
.\" Standard, the original IEEE and The Open Group Standard is the referee
.\" document.  The original Standard can be obtained online at
.\" http://www.opengroup.org/unix/online.html.
.\"
.\" This notice shall appear on any product containing this material.
.\"
.\" The contents of this file are subject to the terms of the
.\" Common Development and Distribution License (the "License").
.\" You may not use this file except in compliance with the License.
.\"
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
.\" or http://www.opensolaris.org/os/licensing.
.\" See the License for the specific language governing permissions
.\" and limitations under the License.
.\"
.\" When distributing Covered Code, include this CDDL HEADER in each
.\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
.\" If applicable, add the following below this CDDL HEADER, with the
.\" fields enclosed by brackets "[]" replaced with your own identifying
.\" information: Portions Copyright [yyyy] [name of copyright owner]
.\"
.\"
.\" Copyright 1989 AT&T
.\" Portions Copyright (c) 1992, X/Open Company Limited All Rights Reserved
.\" Copyright (c) 2006, Sun Microsystems, Inc. All Rights Reserved
.\"
.TH LP 1 "Feb 25, 2017"
.SH NAME
lp \- submit print request
.SH SYNOPSIS
.LP
.nf
\fBlp\fR [\fB-c\fR] [\fB-m\fR] [\fB-p\fR] [\fB-s\fR] [\fB-w\fR] [\fB-d\fR \fIdes
     [\fB-H\fR \fIspecial-handling\fR] [\fB-n\fR \fInumber\fR] [\fB-o\fR \fIopti
     [\fB-P\fR \fIpage-list\fR] [\fB-q\fR \fIpriority-level\fR]
     [\fB-S\fR \fIcharacter-set\fR | \fIprint-wheel\fR] [\fB-t\fR \fItitle\fR]
     [\fB-T\fR \fIcontent-type\fR [\fB-r\fR]] [\fB-y\fR \fImode-list\fR] [\fIfil
.fi

.LP
.nf
```



```
\fBlp\fR \fB-i\fR \fIrequest-ID\fR... [\fB-c\fR] [\fB-m\fR] [\fB-p\fR] [\fB-s\fR
     [\fB-d\fR \fIdestination\fR] [\fB-f\fR \fIform-name\fR] [\fB-H\fR \fIspecia
     [\fB-n\fR \fInumber\fR] [\fB-o\fR \fIoption\fR] [\fB-P\fR \fIpage-list\fR]
     [\fB-q\fR \fIpriority-level\fR] [\fB-S\fR \fIcharacter-set\fR | \fIprint-wh
     [\fB-t\fR \fItitle\fR] [\fB-T\fR \fIcontent-type\fR [\fB-r\fR]] [\fB-y\fR \
.fi

.SH DESCRIPTION
.LP
The \fBlp\fR utility submits print requests to a destination. There are two
formats of the \fBlp\fR command.
.sp
.LP
The first form of \fBlp\fR prints files (\fIfile\fR) and associated information
(collectively called a \fIprint request\fR). If \fIfile\fR is not specified,
\fBlp\fR assumes the standard input. Use a hyphen (\fB\(mi\fR) with \fIfile\fR
to specify the standard input. Files are printed in the order in which they
appear on the command line.
.sp
.LP
The second form of \fBlp\fR changes print request options. This form of
\fBlp\fR can only be used with print services and protocols that support job
modification. The LP print service allows print requests to be modified when
they are in a queue local to the system that the lp commands was executed on.
The Internet Print Protocol (IPP) allows job modification on remote ipp print
services.
.sp
.LP
The print request identified by \fIrequest-ID\fR is changed according to the
printing options specified. The printing options available are the same as
those with the first form of the \fBlp\fR. If the request has finished printing
when the \fBlp\fR command is executed, the change is rejected. If the request
is in the process of printing, it is stopped and restarted from the beginning
(unless the \fB-P\fR option has been given).
.sp
.LP
The print client commands locate destination information using the "printers"
database in the name service switch. See \fBnsswitch.conf\fR(4),
\fBprinters\fR(4), and \fBprinters.conf\fR(4) for details.
.SH OPTIONS
.LP
Printers that have a 4.\fIx\fR or \fBBSD-based print server\fR are not
configured to handle \fBBSD\fR protocol extensions. \fBlp\fR handles print
requests sent to such destinations differently (see NOTES).
.sp
.LP
The following options are supported:
.sp
.ne 2
.na
\fB\fB-c\fR\fR
.ad
.RS 23n
Copies \fIfile\fR before printing.
.sp
Unless \fB-c\fR is specified, users should not remove any \fIfile\fR before the
print request has completely printed. Changes made to \fIfile\fR after the
print request is made but before it is printed might be reflected in the
printed output. \fIfile\fR is linked (as opposed to copied).
.RE

.sp
.ne 2
.na
\fB\fB-d\fR \fIdestination\fR\fR
.ad
```

```
.RS 23n
Prints \fIfile\fR on a specific destination. The \fB-d\fR option is used to set
the destination only when the job is first created. (\fBNote:\fR To move
existing jobs to a different destination, see \fBlpmove\fR(1M).)
\fIdestination\fR can be either a printer or a class of printers (see
\fBlpadmin\fR(1M)). Specify \fIdestination\fR using atomic, URI-style
(\fIscheme\fR://\fIendpoint\fR), or POSIX-style
(\fIserver\fR\fB:\fR\fIdestination\fR) names. See \fBprinters.conf\fR(4) for
more information.
.RE

.sp
.ne 2
.na
\fB\fB-f\fR \fIform-name\fR\fR
.ad
.RS 23n
Prints \fIfile\fR on \fIform-name\fR. The \fBLP\fR print service ensures that
the form is mounted on the printer. The print request is rejected if the
printer does not support \fIform-name\fR, if \fIform-name\fR is not defined for
the system, or if the user is not allowed to use \fIform-name\fR (see
\fBlpforms\fR(1M)).
.RE

.sp
.ne 2
.na
\fB\fB-H\fR \fIspecial-handling\fR\fR
.ad
.RS 23n
Prints the print request according to the value of \fIspecial-handling\fR. The
following \fIspecial-handling\fR values are acceptable:
.sp
.ne 2
.na
\fB\fBhold\fR\fR
.ad
.RS 13n
Do not print the print request until notified. If printing has already begun,
stop it. Other print requests are placed ahead of a request that has been put
on hold (\fIheld print request\fR) until the print request is resumed.
.RE

.sp
.ne 2
.na
\fB\fBresume\fR\fR
.ad
.RS 13n
Resume a held print request. If the print request had begun to print when held,
it is the next print request printed, unless it is superseded by an
\fIimmediate\fR print request.
.RE

.sp
.ne 2
.na
\fB\fBimmediate\fR\fR
.ad
.RS 13n
Print the print request next. If more than one print request is assigned, the
most recent print request is printed next. If a print request is currently
printing on the desired printer, a \fBhold\fR request must be issued to allow
the immediate request to print. The \fBimmediate\fR request is only available
to \fBLP\fR administrators.
.RE
```



```
.RE

.sp
.ne 2
.na
\fB\fB-i\fR \fIrequest-ID\fR\fR
.ad
.RS 23n
Changes options for the print request identified by \fIrequest-ID\fR. There
must be a space between \fB-i\fR and \fIrequest-ID\fR.
.sp
This option applies to jobs that are in a local queue on a print server. This
also applies to remote queues on when the remote print server supports IPP with
job modification.
.RE

.sp
.ne 2
.na
\fB\fB-m\fR\fR
.ad
.RS 23n
Sends mail after \fIfile\fR has printed (see \fBmail\fR(1)). By default, no
mail is sent upon normal completion of a print request.
.RE

.sp
.ne 2
.na
\fB\fB-n\fR \fInumber\fR\fR
.ad
.RS 23n
Prints a specific number of copies of \fIfile\fR. Specify \fInumber\fR as a
digit. The default for \fInumber\fR is \fB1\fR.
.RE

.sp
.ne 2
.na
\fB\fB-o\fR \fIoption\fR\fR
.ad
.RS 23n
Specifies printer-dependent \fIoptions\fR. Specify several options by
specifying \fB-o\fR \fIoption\fR multiple times (\fB-o\fR \fIoption\fR \fB-o\fR
\fIoption\fR \fB-o\fR \fIoption\fR ). Printer-dependent options can also be
specified using the \fB-o\fR keyletter once, followed by a list of options
enclosed in double quotes (\fB-o\fR"\fIoption\fR \fIoption option\fR").
.sp
\fIoption\fRs take the following forms:
.sp
.ne 2
.na
\fB\fIkey\fR\fB=\fR\fIvalue\fR\fR
.ad
.RS 13n
Associates information with the request for use by the backend print service.
The keys and values that can be used are specific to the backend print service
and queue configuration.
.RE

.sp
.ne 2
.na
\fB\fB[no]key\fR\fR
.ad
```

```
.RS 13n
 Associates boolean information with the request for use by the backend print
service. The keys that can be used are specific to the backend print service
and queue configuration.
.RE

The following options are commonly used with the LP print service:
.sp
.ne 2
.na
\fB\fBnobanner\fR\fR
.ad
.sp .6
.RS 4n
Does not print a banner page with the request. This option can be disallowed by
the \fBLP\fR administrator.
.sp
On a system that is configured with Trusted Extensions, use of this option
requires the \fBsolaris.print.nobanner\fR authorization.
.RE

.sp
.ne 2
.na
\fB\fBnofilebreak\fR\fR
.ad
.sp .6
.RS 4n
Prints multiple files without inserting a form feed between them.
.RE

.sp
.ne 2
.na
\fB\fBnolabels\fR\fR
.ad
.sp .6
.RS 4n
On a system that is configured with Trusted Extensions, specifies suppression
of page header and footer labels. Use of this option requires the
\fBsolaris.print.unlabeled\fR authorization.
.RE

.sp
.ne 2
.na
\fB\fBlength=\fR\fInumber\fR\fBi\fR | \fInumber\fR\fBc\fR | \fInumber\fR\fR
.ad
.sp .6
.RS 4n
Prints the print request with pages of a specific length in inches,
centimeters, or number of lines. Append the letter \fBi\fR for inches or
\fBc\fR for centimeters to \fInumber\fR. Indicate the number of lines by
\fBc\fR for centimenters to \fInumber\fR. Indicate the number of lines by
specifying \fInumber\fR alone. \fBlength=66\fR indicates a page length of
\fB66\fR lines. \fBlength=11i\fR indicates a page length of \fB11\fR inches.
\fBlength=27.94c\fR indicates a page length of \fB27.94\fR centimeters.
.sp
This option can not be used with the \fB-f\fR option.
.RE

.sp
.ne 2
.na
\fB\fBwidth=\fR\fInumber\fR\fBi\fR | \fInumber\fR\fBc\fR | \fInumber\fR\fR
.ad
```



```
.sp .6
.RS 4n
Prints the print request with pages of a specific width in inches, centimeters,
or number of columns. Append the letter \fBi\fR for inches or \fBc\fR for
centimeters to \fInumber\fR. Indicate the number of columns by specifying
\fInumber\fR alone. \fBwidth=65\fR indicates a page width of \fB65\fR columns.
\fBwidth=6.5i\fR indicates a page width of \fB6.5\fR inches. \fBwidth=10c\fR
indicates a page width of \fB10\fR centimeters.
.sp
This option can not be used with the \fB-f\fR option.
.RE

.sp
.ne 2
.na
\fB\fBlpi=\fR\fInumber\fR\fR
.ad
.sp .6
.RS 4n
Prints the print request with the line pitch set to \fInumber\fR lines in an
inch. Use \fInumber\fR to specify the number of lines in an inch.
.sp
This option can not be used with the \fB-f\fR option.
.RE

.sp
.ne 2
.na
\fB\fBcpi=\fR\fIn\fR|\fBpica\fR|\fBelite\fR|\fBcompressed\fR\fR
.ad
.sp .6
.RS 4n
Prints the print request with the character pitch set to \fInumber\fR
characters in an inch. Use \fInumber\fR to specify the number of characters in
an inch. Use \fBpica\fR to set character pitch to pica (\fB10\fR characters per
inch), or \fBelite\fR to set character pitch to elite (\fB12\fR characters per
inch) Use \fBcompressed\fR to set character pitch to as many characters as the
printer can handle. There is no standard number of characters per inch for all
printers; see the \fBterminfo\fR database (see \fBterminfo\fR(4)) for the
default character pitch for your printer. This option can not be used with the
\fB-f\fR option.
.RE

.sp
.ne 2
.na
\fB\fBstty=\fR\fIstty-option-list\fR\fR
.ad
.sp .6
.RS 4n
Prints the request using a list of options valid for the \fBstty\fR command
(see \fBstty\fR(1). Enclose the list in single quotes (\fB''\fR) if it contains
blanks.
.RE

.RE

.sp
.ne 2
.na
\fB\fB-P\fR \fIpage-list\fR\fR
.ad
.RS 23n
Prints the pages specified in \fIpage-list\fR in ascending order. Specify
\fIpage-list\fR as a of range of numbers, single page number, or a combination
of both.
```

```
391 .sp
392 The \fB-P\fR option can only be used if there is a filter available to handle
393 it; otherwise, the print request is rejected.
394 .RE

396 .sp
397 .ne 2
398 .na
399 \fB\fB-p\fR\fR
400 .ad
401 .RS 23n
402 Enables notification on completion of the print request. Delivery of the
403 notification is dependent on additional software.
404 .RE

406 .sp
407 .ne 2
408 .na
409 \fB\fB-q\fR \fIpriority-level\fR\fR
410 .ad
411 .RS 23n
412 Assigns the print request a priority in the print queue. Specify
413 \fIpriority-level\fR as an integer between from \fB0\fR and \fB39\fR. Use
414 \fB0\fR to indicate the highest priority; \fB39\fR to indicate the lowest
415 priority. If no priority is specified, the default priority for a print service
416 is assigned by the \fBLP\fR administrator. The \fBLP\fR administrator can also
417 assign a default priority to individual users.
418 .RE

420 .sp
421 .ne 2
422 .na
423 \fB\fB-s\fR\fR
424 .ad
425 .RS 23n
426 Suppresses the display of messages sent from \fBlp\fR.
427 .RE

429 .sp
430 .ne 2
431 .na
432 \fB\fB-S\fR \fIcharacter-set\fR \fB|\fR\fR
433 .ad
434 .br
435 .na
436 \fB\fB-S\fR \fIprint-wheel\fR\fR
437 .ad
438 .RS 23n
439 Prints the request using the \fIcharacter-set\fR or \fIprint-wheel\fR. If a
440 form was requested and requires a character set or print wheel other than the
441 one specified with the \fB-S\fR option, the request is rejected. Printers using
442 mountable print wheels or font cartridges use the print wheel or font cartridge
443 mounted at the time of the print request, unless the \fB-S\fR option is
444 specified.
445 .sp
446 Printers Using Print Wheels: If \fBprint\fR \fIwheel\fR is not one listed by
447 the \fBLP\fR administrator as acceptable for the printer the request is
448 rejected unless the print wheel is already mounted on the printer.
449 .sp
450 Printers Using Selectable or Programmable Character Sets: If the \fB-S\fR
451 option is not specified, \fBlp\fR uses the standard character set. If
452 \fIcharacter-set\fR is not defined in the \fBterminfo\fR database for the
453 printer (see \fBterminfo\fR(4)), or is not an alias defined by the \fBLP\fR
454 administrator, the request is rejected.
455 .RE
```



```
457 .sp
458 .ne 2
459 .na
460 \fB\fB-t\fR \fItitle\fR\fR
461 .ad
462 .RS 23n
463 Prints a title on the banner page of the output. Enclose \fItitle\fR in quotes
464 if it contains blanks. If \fItitle\fR is not not specified, the name of the
465 file is printed on the banner page.
466 .RE

468 .sp
469 .ne 2
470 .na
471 \fB\fB\fR\fB-T\fR\fIcontent-type\fR [\fB-r\fR]\fR
472 .ad
473 .RS 23n
474 Prints the request on a printer that can support the specified
475 \fIcontent-type\fR. If no printer accepts this type directly, a filter is used
476 to convert the content into an acceptable type. If the \fB-r\fR option is
477 specified, a filter is not used. If \fB-r\fR is specified, and no printer
478 accepts the \fIcontent-type\fR directly, the request is rejected. If the
479 \fIcontent-type\fR is not acceptable to any printer, either directly or with a
480 filter, the request is rejected.
481 .RE

483 .sp
484 .ne 2
485 .na
486 \fB\fB-w\fR\fR
487 .ad
488 .RS 23n
489 Writes a message on the user’s terminal after the \fIfile\fRs have been
490 printed. If the user is not logged in, then mail is sent instead.
491 .RE

493 .sp
494 .ne 2
495 .na
496 \fB\fB-y\fR \fImode-list\fR\fR
497 .ad
498 .RS 23n
499 Prints the request according to the printing modes listed in \fImode-list\fR.
500 The allowed values for \fImode-list\fR are locally defined.
501 .sp
502 This option can be used only if there is a filter available to handle it;
503 otherwise, the print request is rejected.
504 .RE

506 .SH OPERANDS
507 .LP
508 The following operand is supported:
509 .sp
510 .ne 2
511 .na
512 \fB\fIfile\fR\fR
513 .ad
514 .RS 8n
515 The name of the file to be printed. Specify \fIfile\fR as a pathname or as a
516 hyphen (\fB\(mi\fR) to indicate the standard input. If \fIfile\fR is not
517 specified, \fBlp\fR uses the standard input.
518 .RE

520 .SH USAGE
521 .LP
522 See \fBlargefile\fR(5) for the description of the behavior of \fBlp\fR when
```

```
encountering files greater than or equal to 2 Gbyte ( 2^31 bytes).
.SH ENVIRONMENT VARIABLES
.LP
See \fBenviron\fR(5) for descriptions of the following environment variables
that affect the execution of \fBlp\fR: \fBLANG\fR, \fBLC_ALL\fR,
\fBLC_CTYPE\fR, \fBLC_MESSAGES\fR, \fBNLSPATH\fR, and \fBPATH\fR.
.sp
.ne 2
.na
\fB\fBLC_TIME\fR\fR
.ad
.RS 11n
Determine the format and contents of date and time strings displayed in the
\fBlp\fR banner page, if any.
.RE

.sp
.ne 2
.na
\fB\fBLPDEST\fR\fR
.ad
.RS 11n
Determine the destination. If the \fBLPDEST\fR environment variable is not set,
the \fBPRINTER\fR environment variable shall be used. The \fB-d\fR \fIdest\fR
option takes precedence over \fBLPDEST\fR. Results are undefined when \fB-d\fR
is not specified and \fBLPDEST\fR contains a value that is not a valid
destination name.
.RE

.sp
.ne 2
.na
\fB\fBPRINTER\fR\fR
.ad
.RS 11n
Determine the output device or destination. If the \fBLPDEST\fR and
\fBPRINTER\fR environment variables are not set, an unspecified output device
is used. The \fB-d\fR \fIdest\fR option and the \fBLPDEST\fR environment
variable shall take precedence over \fBPRINTER\fR. Results are undefined when
\fB-d\fR is not specified, \fBLPDEST\fR is unset, and \fBPRINTER\fR contains a
value that is not a valid device or destination name.
.RE

.sp
.ne 2
.na
\fB\fBTZ\fR\fR
.ad
.RS 11n
Determine the timezone used to calculate date and time strings displayed in the
\fBlp\fR banner page, if any. If \fBTZ\fR is unset or null, an unspecified
default timezone shall be used.
.RE

.SH EXIT STATUS
.LP
The following exit values are returned:
.sp
.ne 2
.na
\fB\fB0\fR\fR
.ad
.RS 12n
Successful completion.
.RE
```



```
.sp
.ne 2
.na
\fBnon-zero\fR
.ad
.RS 12n
An error occurred.
.RE

.SH FILES
.ne 2
.na
\fB\fB/etc/printers.conf\fR\fR
.ad
.RS 24n
System printer configuration database
.RE

.sp
.ne 2
.na
\fB\fB$HOME/.printers\fR\fR
.ad
.RS 24n
User-configurable printer database
.RE

.sp
.ne 2
.na
\fB\fBou=printers\fR\fR
.ad
.RS 24n
LDAP version of \fB/etc/printers.conf\fR
.RE

.sp
.ne 2
.na
\fB\fBprinters.conf.byname\fR\fR
.ad
.RS 24n
\fBNIS\fR version of \fB/etc/printers.conf\fR
.RE

.SH ATTRIBUTES
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
CSI     Enabled. See \fBNOTES\fR.
_
Interface Stability     Standard
.TE

.SH SEE ALSO
.LP
\fBcancel\fR(1), \fBenable\fR(1), \fBlpq\fR(1B), \fBlpr\fR(1B), \fBlprm\fR(1B),
\fBlpstat\fR(1), \fBmail\fR(1), \fBpostprint\fR(1), \fBpr\fR(1), \fBstty\fR(1),
```

```
655 \fBaccept\fR(1M), \fBlpadmin\fR(1M), \fBlpfilter\fR(1M), \fBlpforms\fR(1M),
656 \fBlpmove\fR(1M), \fBlpsched\fR(1M), \fBlpshut\fR(1M), \fBlpsystem\fR(1M),
657 \fBlpusers\fR(1M), \fBnsswitch.conf\fR(4), \fBprinters\fR(4),
658 \fBprinters.conf\fR(4), \fBterminfo\fR(4), \fBattributes\fR(5),
659 \fBenviron\fR(5), \fBlargefile\fR(5), \fBstandards\fR(5)
660 .SH NOTES
661 .LP
662 \fBCSI\fR-capability assumes that printer names are composed of \fBASCII\fR
663 characters.
664 .sp
665 .LP
666 Print jobs are assumed to contain one type of data. That type of data is either
667 specified on the command line or autodetected (simple, PostScript) based on the
668 contents of the first file in the job.
669 .sp
670 .LP
671 When using the BSD printing protocol to send print requests to a remote print
672 service, functionality is limited.
673 .sp
674 .LP
675 Printers that have a 4.\fIx\fR or BSD-based print server are not configured to
676 handle BSD protocol extensions. \fBlp\fR handles print requests sent to such
677 printers in the following ways:
678 .RS +4
679 .TP
680 1.
681 Print requests with more than 52 filenames are truncated to 52 files.
682 .RE
683 .RS +4
684 .TP
685 2.
686 The \fB-f\fR, \fB-H\fR, \fB-o\fR, \fB-P\fR, \fB-p\fR, \fB-q\fR, \fB-S\fR,
687 \fB-T\fR, and \fB-y\fR options might require a protocol extension to pass to a
688 print server. If \fBlp\fR cannot handle the print request, it displays a
689 warning message.
690 .sp
691 \fBLP\fR administrators enable protocol extensions by setting a printer's
692 \fBprinter-uri-supported\fR (or \fBbsdaddr\fR) entry in
693 \fB/etc/printers.conf\fR. Changing the \fBprinter-uri-supported\fR entry in
694 \fB/etc/printers.conf\fR to:
695 .sp
696 .in +2
697 .nf
698 \fBprinter-uri-supported=lpd\e://\fR\fIserver\fR\fB/\fR\fIprinters\fR\fB/\fR\fId
699 .fi
700 .in -2
701 .sp

703 .sp
704 .in +2
705 .nf
706 \fBbsdaddr=\fR\fIserver\fR\fB,\fR\fIdestination\fR\fB,Solaris\fR
707 .fi
708 .in -2
709 .sp

711 Adding \fBSolaris\fR to either of these values causes the \fBlp\fR command to
712 generate a set of BSD print protocol extensions that can be processed by a
713 Solaris print server.
714 .RE
715 .sp
716 .LP
717 As a result of several limitations in the BSD print protocol, it is recommended
718 that the IPP protocol be used for communication with print servers.
719 .sp
720 .LP
```



```
721 When IPP is in use, the user is prompted for a passphrase if the remote print
722 service is configured to require authentication.
```

```
**********************************************************
   33606 Sun Sep 16 19:22:53 2018
new/usr/src/man/man1/priocntl.1
9842 man page typos and spelling
**********************************************************
'\" te
.\" Copyright (c) 2008, Sun Microsystems, Inc.  All Rights Reserved.
.\" Copyright 1989 AT&T
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH PRIOCNTL 1 "Apr 1, 2008"
.SH NAME
priocntl \- display or set scheduling parameters of specified process(es)
.SH SYNOPSIS
.LP
.nf
\fBpriocntl\fR \fB-l\fR
.fi

.LP
.nf
\fBpriocntl\fR \fB-d\fR [\fB-i\fR \fIidtype\fR] [\fIidlist\fR]
.fi

.LP
.nf
\fBpriocntl\fR \fB-s\fR [\fB-c\fR \fIclass\fR] [\fIclass-specific\fR \fIoptions\
     [\fB-i\fR \fIidtype\fR] [\fIidlist\fR]
.fi

.LP
.nf
\fBpriocntl\fR \fB-e\fR [\fB-c\fR \fIclass\fR] [\fIclass-specific\fR \fIoptions\
     [\fIargument(s)\fR]
.fi

.SH DESCRIPTION
.sp
.LP
The \fBpriocntl\fR command displays or sets scheduling parameters of the
specified process(es). It can also be used to display the current configuration
information for the system's process scheduler or execute a command with
specified scheduling parameters.
.sp
.LP
Processes fall into distinct classes with a separate scheduling policy applied
to each class. The process classes currently supported are the real-time class,
time-sharing class, interactive class, fair-share class, and the fixed priority
class. The characteristics of these classes and the class-specific options they
accept are described below in the USAGE section under the headings \fBReal-Time
Class\fR, \fBTime-Sharing Class\fR, \fBInter-Active Class\fR, \fBFair-Share
Class\fR, and \fBFixed-Priority Class\fR. With appropriate permissions, the
\fBpriocntl\fR command can change the class and other scheduling parameters
associated with a running process.
.sp
.LP
In the default configuration, a runnable real-time process runs before any
other process. Therefore, inappropriate use of real-time processes can have a
dramatic negative impact on system performance.
.sp
.LP
If an \fIidlist\fR is present, it must appear last on the command line and the
elements of the list must be separated by white space. If no \fIidlist\fR is
present, an \fIidtype\fR argument of \fBpid\fR, \fBppid\fR, \fBpgid\fR,
\fBsid\fR, \fBtaskid\fR, \fBclass\fR, \fBuid\fR, \fBgid\fR, \fBprojid\fR, or
```



```
\fBzoneid\fR specifies the process \fBID\fR, parent process \fBID\fR, process
group \fBID\fR, session \fBID\fR, task \fBID\fR, class, user \fBID\fR, group
\fBID\fR, project \fBID\fR, or zone \fBID\fR, respectively, of the
\fBpriocntl\fR command itself.
.sp
.LP
The command
.sp
.in +2
.nf
\fBpriocntl -d [-i \fIidtype\fR] [\fIidlist\fR]\fR
.fi
.in -2
.sp

.sp
.LP
displays the class and class-specific scheduling parameters of the process(es)
specified by \fIidtype\fR and \fIidlist\fR.
.sp
.LP
The command
.sp
.in +2
.nf
\fBpriocntl -s [-c \fIclass\fR] [\fIclass-specific options\fR] \e
     [-i \fIidtype\fR] [\fIidlist\fR]\fR
.fi
.in -2
.sp

.sp
.LP
sets the class and class-specific parameters of the specified processes to the
values given on the command line. The \fB-c\fR \fIclass\fR option specifies the
class to be set. (The valid \fIclass\fR arguments are \fBRT\fR for real-time,
\fBTS\fR for time-sharing, \fBIA\fR for inter-active, \fBFSS\fR for fair-share,
or \fBFX\fR for fixed-priority.)
.sp
.LP
The class-specific parameters to be set are specified by the class-specific
options as explained under the appropriate heading below. If the \fB-c\fR
\fIclass\fR option is omitted, \fIidtype\fR and \fIidlist\fR must specify a set
of processes which are all in the same class, otherwise an error results. If no
class-specific options are specified, the process's class-specific parameters
are set to the default values for the class specified by \fB-c\fR \fIclass\fR
(or to the default parameter values for the process's current class if the
\fB-c\fR \fIclass\fR option is also omitted).
.sp
.LP
In order to change the scheduling parameters of a process using \fBpriocntl\fR
the real or effective user \fBID\fR (respectively, groupID) of the user
invoking \fBpriocntl\fR must match the real or effective user \fBID\fR
(respectively, groupID) of the receiving process or the effective user \fBID\fR
of the user must be super-user. These are the minimum permission requirements
enforced for all classes. An individual class can impose additional permissions
requirements when setting processes to that class or when setting
class-specific scheduling parameters.
.sp
.LP
When \fIidtype\fR and \fIidlist\fR specify a set of processes, \fBpriocntl\fR
acts on the processes in the set in an implementation-specific order. If
\fBpriocntl\fR encounters an error for one or more of the target processes, it
can or cannot continue through the set of processes, depending on the nature of
the error.
.sp
```

```
127 .LP
128 If the error is related to permissions, \fBpriocntl\fR prints an error message
129 and then continues through the process set, resetting the parameters for all
130 target processes for which the user has appropriate permissions. If
131 \fBpriocntl\fR encounters an error other than permissions, it does not continue
132 through the process set but prints an error message and exits immediately.
133 .sp
134 .LP
135 A special \fBsys\fR scheduling class exists for the purpose of scheduling the
136 execution of certain special system processes (such as the swapper process). It
137 is not possible to change the class of any process to \fBsys\fR. In addition,
138 any processes in the \fBsys\fR class that are included in the set of processes
139 specified by \fIidtype\fR and \fIidlist\fR are disregarded by \fBpriocntl\fR.
140 For example, if \fIidtype\fR were \fBuid\fR, an \fIidlist\fR consisting of a
141 zero would specify all processes with a \fBUID\fR of \fB0\fR, except processes
142 in the \fBsys\fR class and (if changing the parameters using the \fB-s\fR
143 option) the \fBinit\fR process.
144 .sp
145 .LP
146 The \fBinit\fR process (process \fBID\fR \fB1\fR) is a special case. In order
147 for the \fBpriocntl\fR command to change the class or other scheduling
148 parameters of the \fBinit\fR process, \fIidtype\fR must be \fBpid\fR and
149 \fIidlist\fR must be consist of only a \fB1\fR. The \fBinit\fR process can be
150 assigned to any class configured on the system, but the time-sharing class is
151 almost always the appropriate choice. Other choices can be highly undesirable;
152 see the \fISystem Administration Guide: Basic Administration\fR for more
153 information.
154 .sp
155 .LP
156 The command
157 .sp
158 .in +2
159 .nf
160 \fBpriocntl -e [-c \fIclass\fR\fR\fB]\fR\fB [\fIclass-specific options\fR] \fIco
161      [\fIargument...\fR]\fR
162 .fi
163 .in -2
164 .sp

166 .sp
167 .LP
168 executes the specified command with the class and scheduling parameters
169 specified on the command line (\fIarguments\fR are the arguments to the
170 command). If the \fB-c\fR \fIclass\fR option is omitted the command is run in
171 the user's current class.
172 .SH OPTIONS
174 .sp
173 .LP
174 The following options are supported:
175 .sp
176 .ne 2
177 .na
178 \fB\fB-c\fR \fIclass\fR\fR
179 .ad
180 .RS 13n
181 Specifies the \fIclass\fR to be set. (The valid \fIclass\fR arguments are
182 \fBRT\fR for real-time, \fBTS\fR for time-sharing, \fBIA\fR for inter-active,
183 \fBFSS\fR for fair-share, or \fBFX\fR for fixed-priority.) If the specified
184 class is not already configured, it is automatically configured.
185 .RE

187 .sp
188 .ne 2
189 .na
190 \fB\fB-d\fR\fR
191 .ad
```



```
192 .RS 13n
193 Displays the scheduling parameters associated with a set of processes.
194 .RE

196 .sp
197 .ne 2
198 .na
199 \fB\fB-e\fR\fR
200 .ad
201 .RS 13n
202 Executes a specified command with the class and scheduling parameters
203 associated with a set of processes.
204 .RE

206 .sp
207 .ne 2
208 .na
209 \fB\fB-i\fR \fIidtype\fR\fR
210 .ad
211 .RS 13n
212 This option, together with the \fIidlist\fR arguments (if any), specifies one
213 or more processes to which the \fBpriocntl\fR command is to apply. The
214 interpretation of \fIidlist\fR depends on the value of \fIidtype\fR. If the
215 \fB-i\fR \fIidtype\fR option is omitted when using the \fB-d\fR or \fB-s\fR
216 options the default \fIidtype\fR of \fBpid\fR is assumed.
217 .sp
218 The valid \fIidtype\fR arguments and corresponding interpretations of
219 \fIidlist\fR are as follows:
220 .sp
221 .ne 2
222 .na
223 \fB\fB-i\fR \fBall\fR\fR
224 .ad
225 .RS 13n
226 The \fBpriocntl\fR command applies to all existing processes. No \fIidlist\fR
227 should be specified (if one is specified, it is ignored). The permission
228 restrictions described below still apply.
229 .RE

231 .sp
232 .ne 2
233 .na
234 \fB\fB-i\fR \fBctid\fR\fR
235 .ad
236 .RS 13n
237 idlist is a list of process contract IDs. The \fBpriocntl\fR command applies to
238 all processes with a process contract ID equal to an ID from the list.
239 .RE

241 .sp
242 .ne 2
243 .na
244 \fB\fB-i\fR \fBclass\fR\fR
245 .ad
246 .RS 13n
247 \fIidlist\fR consists of a single class name (\fBRT\fR for real-time, \fBTS\fR
248 for time-sharing, \fBIA\fR for inter-active, \fBFSS\fR for fair-share, or
249 \fBFX\fR for fixed-priority). The \fBpriocntl\fR command applies to all
250 processes in the specified class.
251 .RE

253 .sp
254 .ne 2
255 .na
256 \fB\fB-i\fR \fBgid\fR\fR
257 .ad
```

```
.RS 13n
\fIidlist\fR is a list of group \fBID\fRs. The \fBpriocntl\fR command applies
to all processes with an effective group \fBID\fR equal to an \fBID\fR from the
list.
.RE

.sp
.ne 2
.na
\fB\fB-i\fR \fBpgid\fR\fR
.ad
.RS 13n
\fIidlist\fR is a list of process group \fBID\fRs. The \fBpriocntl\fR command
applies to all processes in the specified process groups.
.RE

.sp
.ne 2
.na
\fB\fB-i\fR \fBpid\fR\fR
.ad
.RS 13n
\fIidlist\fR is a list of process \fBID\fRs. The \fBpriocntl\fR command applies
to the specified processes.
.RE

.sp
.ne 2
.na
\fB\fB-i\fR \fBppid\fR\fR
.ad
.RS 13n
\fIidlist\fR is a list of parent process \fBID\fRs. The \fBpriocntl\fR command
applies to all processes whose parent process \fBID\fR is in the list.
.RE

.sp
.ne 2
.na
\fB\fB-i\fR \fBprojid\fR\fR
.ad
.RS 13n
\fIidlist\fR is a list of project \fBID\fRs. The \fBpriocntl\fR command applies
to all processes with an effective project \fBID\fR equal to an \fBID\fR from
the list.
.RE

.sp
.ne 2
.na
\fB\fB-i\fR \fBsid\fR\fR
.ad
.RS 13n
\fIidlist\fR is a list of session \fBID\fRs. The \fBpriocntl\fR command applies
to all processes in the specified sessions.
.RE

.sp
.ne 2
.na
\fB\fB-i\fR \fBtaskid\fR\fR
.ad
.RS 13n
\fIidlist\fR is a list of task \fBID\fRs. The \fBpriocntl\fR command applies to
all processes in the specified tasks.
.RE
```



```
.sp
.ne 2
.na
\fB\fB-i\fR \fBuid\fR\fR
.ad
.RS 13n
\fIidlist\fR is a list of user \fBID\fRs. The \fBpriocntl\fR command applies to
all processes with an effective user \fBID\fR equal to an \fBID\fR from the
list.
.RE

.sp
.ne 2
.na
\fB\fB-i\fR \fBzoneid\fR\fR
.ad
.RS 13n
\fIidlist\fR is a list of zone \fBID\fRs. The \fBpriocntl\fR command applies to
all processes with an effective zone \fBID\fR equal to an \fBID\fR from the
list.
.RE

.RE

.sp
.ne 2
.na
\fB\fB-l\fR\fR
.ad
.RS 13n
Displays a list of the classes currently configured in the system along with
class-specific information about each class. The format of the class-specific
information displayed is described under USAGE.
.RE

.sp
.ne 2
.na
\fB\fB-s\fR\fR
.ad
.RS 13n
Sets the scheduling parameters associated with a set of processes.
.RE

.sp
.LP
The valid class-specific options for setting real-time parameters are:
.sp
.ne 2
.na
\fB\fB-p\fR \fIrtpri\fR\fR
.ad
.RS 21n
Sets the real-time priority of the specified process(es) to \fIrtpri\fR.
.RE

.sp
.ne 2
.na
\fB\fB-t\fR \fItqntm\fR [\fB-r\fR \fIres\fR]\fR
.ad
.RS 21n
Sets the time quantum of the specified process(es) to \fItqntm\fR. You can
optionally specify a resolution as explained below.
.RE
```

```
391 .sp
392 .ne 2
393 .na
394 \fB\fB-q\fR \fItqsig\fR\fR
395 .ad
396 .RS 21n
397 Sets the real-time time quantum signal of the specified process(es) to
398 \fItqsig\fR.
399 .RE

401 .sp
402 .LP
403 The valid class-specific options for setting time-sharing parameters are:
404 .sp
405 .ne 2
406 .na
407 \fB\fB-m\fR \fItsuprilim\fR\fR
408 .ad
409 .RS 16n
410 Sets the user priority limit of the specified process(es) to \fItsuprilim\fR.
411 .RE

413 .sp
414 .ne 2
415 .na
416 \fB\fB-p\fR \fItsupri\fR\fR
417 .ad
418 .RS 16n
419 Sets the user priority of the specified process(es) to \fItsupri\fR.
420 .RE

422 .sp
423 .LP
424 The valid class-specific options for setting inter-active parameters are:
425 .sp
426 .ne 2
427 .na
428 \fB\fB-m\fR \fIiauprilim\fR\fR
429 .ad
430 .RS 16n
431 Sets the user priority limit of the specified process(es) to \fIiauprilim\fR.
432 .RE

434 .sp
435 .ne 2
436 .na
437 \fB\fB-p\fR \fIiaupri\fR\fR
438 .ad
439 .RS 16n
440 Sets the user priority of the specified process(es) to \fIiaupri\fR.
441 .RE

443 .sp
444 .LP
445 The valid class-specific options for setting fair-share parameters are:
446 .sp
447 .ne 2
448 .na
449 \fB\fB-m\fR \fIfssuprilim\fR\fR
450 .ad
451 .RS 17n
452 Sets the user priority limit of the specified process(es) to \fIfssuprilim\fR.
453 .RE

455 .sp
```



```
456 .ne 2
457 .na
458 \fB\fB-p\fR \fIfssupri\fR\fR
459 .ad
460 .RS 17n
461 Sets the user priority of the specified process(es) to \fIfssupri\fR.
462 .RE

464 .sp
465 .LP
466 The valid class-specific options for setting fixed-priority parameters are:
467 .sp
468 .ne 2
469 .na
470 \fB\fB-m\fR \fIfxuprilim\fR\fR
471 .ad
472 .RS 16n
473 Sets the user priority limit of the specified process(es) to \fIfxuprilim\fR.
474 .RE

476 .sp
477 .ne 2
478 .na
479 \fB\fB-p\fR \fIfxupri\fR\fR
480 .ad
481 .RS 16n
482 Sets the user priority of the specified process(es) to \fIfxupri\fR.
483 .RE

485 .sp
486 .ne 2
487 .na
488 \fB\fB-t\fR \fItqntm\fR\fR
489 .ad
490 .RS 16n
491 [\fB-r\fR \fIres\fR] Sets the time quantum of the specified process(es) to
492 \fItqntm\fR. You can optionally specify a resolution as explained below.
493 .RE

495 .SH USAGE
496 .SS "Real-Time Class"
499 .sp
497 .LP
498 The real-time class provides a fixed priority preemptive scheduling policy for
499 those processes requiring fast and deterministic response and absolute
500 user/application control of scheduling priorities. If the real-time class is
501 configured in the system, it should have exclusive control of the highest range
502 of scheduling priorities on the system. This ensures that a runnable real-time
503 process is given \fBCPU\fR service before any process belonging to any other
504 class.
505 .sp
506 .LP
507 The real-time class has a range of real-time priority (\fIrtpri\fR) values that
508 can be assigned to processes within the class. Real-time priorities range from
509 0 to \fIx\fR, where the value of \fIx\fR is configurable and can be displayed
510 for a specific installation that has already configured a real-time scheduler,
511 by using the command
512 .sp
513 .in +2
514 .nf
515 \fBpriocntl -l\fR
516 .fi
517 .in -2
518 .sp

520 .sp
```

```
.LP
The real-time scheduling policy is a fixed priority policy. The scheduling
priority of a real-time process never changes except as the result of an
explicit request by the user/application to change the \fIrtpri\fR value of the
process.
.sp
.LP
For processes in the real-time class, the \fIrtpri\fR value is, for all
practical purposes, equivalent to the scheduling priority of the process. The
\fIrtpri\fR value completely determines the scheduling priority of a real-time
process relative to other processes within its class. Numerically higher
\fIrtpri\fR values represent higher priorities. Since the real-time class
controls the highest range of scheduling priorities in the system, it is
guaranteed that the runnable real-time process with the highest \fIrtpri\fR
value is always selected to run before any other process in the system.
.sp
.LP
In addition to providing control over priority, \fBpriocntl\fR provides for
control over the length of the time quantum allotted to processes in the
real-time class. The time quantum value specifies the maximum amount of time a
process can run, assuming that it does not complete or enter a resource or
event wait state (\fBsleep\fR). Notice that if another process becomes runnable
at a higher priority, the currently running process can be preempted before
receiving its full time quantum.
.sp
.LP
The command
.sp
.in +2
.nf
\fBpriocntl -d [-i \fIidtype\fR] [\fIidlist\fR]\fR
.fi
.in -2
.sp

.sp
.LP
displays the real-time priority, time quantum (in millisecond resolution), and
time quantum signal value for each real-time process in the set specified by
\fIidtype\fR and \fIidlist\fR.
.sp
.LP
Any combination of the \fB-p\fR, \fB-t\fR [\fB-r\fR], and \fB-q\fR options can
be used with \fBpriocntl\fR \fB-s\fR or \fBpriocntl\fR \fB-e\fR for the
real-time class. If an option is omitted and the process is currently
real-time, the associated parameter is unaffected. If an option is omitted when
changing the class of a process to real-time from some other class, the
associated parameter is set to a default value. The default value for
\fIrtpri\fR is \fB0\fR and the default for time quantum is dependent on the
value of \fIrtpri\fR and on the system configuration; see \fBrt_dptbl\fR(4).
.sp
.LP
When using the \fB-t\fR \fItqntm\fR option, you can optionally specify a
resolution using the \fB-r\fR \fIres\fR option. (If no resolution is specified,
millisecond resolution is assumed.) If \fIres\fR is specified, it must be a
positive integer between \fB1\fR and \fB1,000,000,000\fR inclusively and the
resolution used is the reciprocal of \fIres\fR in seconds. For example,
specifying \fB-t\fR \fB10\fR \fB-r\fR \fB100\fR would set the resolution to
hundredths of a second and the resulting time quantum length would be 10/100
seconds (one tenth of a second). Although very fine (nanosecond) resolution can
be specified, the time quantum length is rounded up by the system to the next
integral multiple  of the system clock’s resolution. Requests for time quantums
of zero or quantums greater than the (typically very large)
implementation-specific maximum quantum result in an error.
.sp
.LP
```



```
The real-time time quantum signal can be used to notify runaway real-time
processes about the consumption of their time quantum. Those processes, which
are monitored by the real-time time quantum signal, receive the configured
signal in the event of time quantum expiration. The default value (\fB0\fR) of
the time quantum signal \fItqsig\fR denotes no signal delivery. A positive
value denotes the delivery of the signal specified by the value. Like
\fBkill\fR(1) and other commands operating on signals, the \fB-q\fR \fItqsig\fR
option is also able to handle symbolically named signals, like \fBXCPU\fR or
\fBKILL\fR.
.sp
.LP
In order to change the class of a process to real-time (from any other class),
the user invoking \fBpriocntl\fR must have super-user privilege. In order to
change the \fIrtpri\fR value or time quantum of a real-time process, the user
invoking \fBpriocntl\fR must either be super-user, or must currently be in the
real-time class (shell running as a real-time process) with a real or effective
user \fBID\fR matching the real or effective user \fBID\fR of the target
process.
.sp
.LP
The real-time priority, time quantum, and time quantum signal are inherited
across the \fBfork\fR(2) and \fBexec\fR(2) system calls. When using the time
quantum signal with a user defined signal handler across the \fBexec\fR(2)
system call, the new image must install an appropriate user defined signal
handler before the time quantum expires. Otherwise, unpredictable behavior would
handler before the time quantum expires. Otherwise, unpredicable behavior would
result.
.SS "Time-Sharing Class"
.sp
.LP
The time-sharing scheduling policy provides for a fair and effective allocation
of the \fBCPU\fR resource among processes with varying \fBCPU\fR consumption
characteristics. The objectives of the time-sharing policy are to provide good
response time to interactive processes and good throughput to \fBCPU\fR-bound
jobs, while providing a degree of user/application control over scheduling.
.sp
.LP
The time-sharing class has a range of time-sharing user priority (\fItsupri\fR)
values that can be assigned to processes within the class. User priorities
range from \(mi\fIx\fR to +\fIx\fR, where the value of \fIx\fR is configurable.
The range for a specific installation can be displayed by using the command
.sp
.in +2
.nf
\fBpriocntl -l\fR
.fi
.in -2
.sp

.sp
.LP
The purpose of the user priority is to provide some degree of user/application
control over the scheduling of processes in the time-sharing class. Raising or
lowering the \fItsupri\fR value of a process in the time-sharing class raises
or lowers the scheduling priority of the process. It is not guaranteed,
however, that a time-sharing process with a higher \fItsupri\fR value runs
before one with a lower \fItsupri\fR value. This is because the \fItsupri\fR
value is just one factor used to determine the scheduling priority of a
time-sharing process. The system can dynamically adjust the internal scheduling
priority of a time-sharing process based on other factors such as recent
\fBCPU\fR usage.
.sp
.LP
In addition to the system-wide limits on user priority (displayed with
\fBpriocntl\fR \fB-l\fR), there is a per process user priority limit
(\fItsuprilim\fR), which specifies the maximum \fItsupri\fR value that can be
```

set for a given process.
.sp
.LP
The command
.sp
.in +2
.nf
\fBpriocntl -d [-i \fIidtype\fR] [\fIidlist\fR]\fR
.fi
.in -2
.sp

.sp
.LP
displays the user priority and user priority limit for each time-sharing process in the set specified by \fIidtype\fR and \fIidlist\fR.
.sp
.LP
Any time-sharing process can lower its own \fItsuprilim\fR (or that of another process with the same user \fBID\fR). Only a time-sharing process with super-user privilege can raise a \fItsuprilim\fR. When changing the class of a process to time-sharing from some other class, super-user privilege is required in order to set the initial \fItsuprilim\fR to a value greater than zero.
.sp
.LP
Any time-sharing process can set its own \fItsupri\fR (or that of another process with the same user \fBID\fR) to any value less than or equal to the process's \fItsuprilim\fR. Attempts to set the \fItsupri\fR above the \fItsuprilim\fR (and/or set the \fItsuprilim\fR below the \fItsupri\fR) result in the \fItsupri\fR being set equal to the \fItsuprilim\fR.
.sp
.LP
Any combination of the \fB-m\fR and \fB-p\fR options can be used with \fBpriocntl\fR \fB-s\fR or \fBpriocntl\fR \fB-e\fR for the time-sharing class. If an option is omitted and the process is currently time-sharing, the associated parameter is normally unaffected. The exception is when the \fB-p\fR option is omitted and \fB-m\fR is used to set a \fItsuprilim\fR below the current \fItsupri\fR. In this case, the \fItsupri\fR is set equal to the \fItsuprilim\fR which is being set. If an option is omitted when changing the class of a process to time-sharing from some other class, the associated parameter is set to a default value. The default value for \fItsuprilim\fR is \fB0\fR and the default for \fItsupri\fR is to set it equal to the \fItsuprilim\fR value which is being set.
.sp
.LP
The time-sharing user priority and user priority limit are inherited across the \fBfork\fR(2) and \fBexec\fR(2) system calls.
.SS "Inter-Active Class"
.sp
.LP
The inter-active scheduling policy provides for a fair and effective allocation of the \fBCPU\fR resource among processes with varying \fBCPU\fR consumption characteristics while providing good responsiveness for user interaction. The objectives of the inter-active policy are to provide good response time to interactive processes and good throughput to \fBCPU\fR-bound jobs. The priorities of processes in the inter-active class can be changed in the same manner as those in the time-sharing class, though the modified priorities continue to be adjusted to provide good responsiveness for user interaction.
.sp
.LP
The inter-active user priority limit, \fIiaupri\fR, is equivalent to \fItsupri\fR. The inter-active per process user priority, \fIiauprilim\fR, is equivalent to \fItsuprilim\fR.
.sp
.LP
Inter-active class processes that have the \fIiamode\fR ("interactive mode")

bit set are given a priority boost value of \fB10\fR, which is factored into the user mode priority of the process when that calculation is made, that is, every time a process's priority is adjusted. This feature is used by the X windowing system, which sets this bit for those processes that run inside of the current active window to give them a higher priority.
.SS "Fair-Share Class"
.sp
.LP
The fair-share scheduling policy provides a fair allocation of system \fBCPU\fR resources among projects, independent of the number of processes they own. Projects are given "shares" to control their entitlement to \fBCPU\fR resources. Resource usage is remembered over time, so that entitlement is reduced for heavy usage, and increased for light usage, with respect to other projects. \fBCPU\fR time is scheduled among processes according to their owner's entitlements, independent of the number of processes each project owns.
.sp
.LP
The \fBFSS\fR scheduling class supports the notion of per-process user priority and user priority limit for compatibility with the time-share scheduler. The fair share scheduler attempts to provide an evenly graded effect across the whole range of user priorities. Processes with negative \fIfssupri\fR values receive time slices less frequently than normal, while processes with positive \fIfssupri\fR values receive time slices more frequently than normal. Notice that user priorities do not interfere with shares. That is, changing a \fIfssupri\fR value of a process is not going to affect its project's overall \fBCPU\fR usage which only relates to the amount of shares it is allocated compared to other projects.
.sp
.LP
The priorities of processes in the fair-share class can be changed in the same manner as those in the time-share class.
.SS "Fixed-Priority Class"
.sp
.LP
The fixed-priority class provides a fixed priority preemptive scheduling policy for those processes requiring that the scheduling priorities do not get dynamically adjusted by the system and that the user/application have control of the scheduling priorities.
.sp
.LP
The fixed-priority class shares the same range of scheduling priorities with the time-sharing class, by default. The fixed-priority class has a range of fixed-priority user priority (\fIfxupri\fR) values that can be assigned to processes within the class. User priorities range from 0 to \fIx\fR, where the value of \fIx\fR is configurable. The range for a specific installation can be displayed by using the command
.sp
.in +2
.nf
\fBpriocntl -l\fR
.fi
.in -2
.sp

.sp
.LP
The purpose of the user priority is to provide user/application control over the scheduling of processes in the fixed-priority class. For processes in the fixed-priority class, the \fIfxupri\fR value is, for all practical purposes, equivalent to the scheduling priority of the process. The \fIfxupri\fR value completely determines the scheduling priority of a fixed-priority process relative to other processes within its class. Numerically higher \fIfxupri\fR values represent higher priorities.
.sp
.LP
In addition to the system-wide limits on user priority (displayed with

```
780 \fBpriocntl\fR  \fB-l\fR), there is a per process user priority  limit
781 (\fIfxuprilim\fR), which specifies the maximum \fIfxupri\fR value that can be
782 set for a given process.
783 .sp
784 .LP
785 Any fixed-priority process can lower its own \fIfxuprilim\fR (or that of
786 another process with the same user \fBID\fR). Only a process with super-user
787 privilege can raise a \fIfxuprilim\fR. When changing the class of a process to
788 fixed-priority from some other class, super-user privilege is required in order
789 to set the initial \fIfxuprilim\fR to a value greater than zero.
790 .sp
791 .LP
792 Any fixed-priority process can set its own \fIfxupri\fR (or that of another
793 process with the same user \fBID\fR) to any value less than or equal to the
794 process's \fIfxuprilim\fR. Attempts to set the \fIfxupri\fR above the
795 \fIfxuprilim\fR (or set the \fIfxuprilim\fR below the \fIfxupri\fR) result in
796 the \fIfxupri\fR being set equal to the \fIfxuprilim\fR.
797 .sp
798 .LP
799 In addition to providing control over priority, \fBpriocntl\fR provides for
800 control over the length of the time quantum allotted to processes in the
801 fixed-priority class. The time quantum value specifies the maximum amount of
802 time a process can run, before surrendering the \fBCPU\fR, assuming that it
803 does not complete or enter a resource or event wait state (sleep). Notice that
804 if another process becomes runnable at a higher priority, the currently running
805 process can be preempted before receiving its full time quantum.
806 .sp
807 .LP
808 Any combination of the \fB-m\fR, \fB-p\fR, and \fB-t\fR options can be used
809 with \fBpriocntl\fR \fB-s\fR or \fBpriocntl\fR \fB-e\fR for the fixed-priority
810 class. If an option is omitted and the process is currently fixed-priority, the
811 associated parameter is normally unaffected. The exception is when the \fB-p\fR
812 option is omitted and the \fB-m\fR option is used to set a \fIfxuprilim\fR
813 below the current \fIfxupri\fR. In this case, the \fIfxupri\fR is set equal to
814 the \fIfxuprilim\fR which is being set. If an option is omitted when changing
815 the class of a process to fixed-priority from some other class, the associated
816 parameter is set to a default value. The default value for \fIfxuprilim\fR is
817 \fB0\fR. The default for \fIfxupri\fR is to set it equal to the \fIfxuprilim\fR
818 value which is being set. The default for time quantum is dependent on the
819 \fIfxupri\fR and on the system configuration. See \fBfx_dptbl\fR(4).
820 .sp
821 .LP
822 The time quantum of processes in the fixed-priority class      can be changed
823 in the same manner as those in the real-time class.
824 .sp
825 .LP
826 The fixed-priority user priority, user priority limit, and time quantum are
827 inherited across the \fBfork\fR(2) and \fBexec\fR(2) system calls.
828 .SH EXAMPLES
836 .sp
829 .LP
830 The following are real-time class examples:
831 .LP
832 \fBExample 1 \fRSetting the Class
833 .sp
834 .LP
835 The following example sets the class of any non-real-time processes selected by
836 \fIidtype\fR and \fIidlist\fR to real-time and sets their real-time priority to
837 the default value of \fB0\fR. The real-time priorities of any processes
838 currently in the real-time class are unaffected. The time quantums of all of
839 the specified processes are set to \fB1/10\fR seconds.

841 .sp
842 .in +2
843 .nf
844 example% \fBpriocntl -s -c RT -t 1 -r 10 -i \fIidtype idlist\fR\fR
```



```
845 .fi
846 .in -2
847 .sp

849 .LP
850 \fBExample 2 \fRExecuting a Command in Real-time
851 .sp
852 .LP
853 The following example executes \fIcommand\fR in the real-time class with a
854 real-time priority of \fB15\fR and a time quantum of \fB20\fR milliseconds:

856 .sp
857 .in +2
858 .nf
859 example% \fBpriocntl -e -c RT -p 15 -t 20 \fIcommand\fR\fR
860 .fi
861 .in -2
862 .sp

864 .LP
865 \fBExample 3 \fRExecuting a Command in Real-time with a Specified Quantum
866 Signal
867 .sp
868 .LP
869 The following example executes \fIcommand\fR in the real-time class with a
870 real-time priority of \fB11\fR, a time quantum of \fB250\fR milliseconds, and
871 where the specified real-time quantum signal is \fBSIGXCPU\fR:

873 .sp
874 .in +2
875 .nf
876 example% \fBpriocntl -e -c RT -p 11 -t 250 -q XCPU \fIcommand\fR\fR
877 .fi
878 .in -2
879 .sp

881 .sp
882 .LP
883 The following are time-sharing class examples:
884 .LP
885 \fBExample 4 \fRSetting the Class of non-time-sharing Processes
886 .sp
887 .LP
888 The following example sets the class of any non-time-sharing processes selected
889 by \fIidtype\fR and \fIidlist\fR to time-sharing and sets both their user
890 priority limit and user priority to \fB0\fR. Processes already in the
891 time-sharing class are unaffected.

893 .sp
894 .in +2
895 .nf
896 example% \fBpriocntl -s -c TS -i \fIidtype idlist\fR\fR
897 .fi
898 .in -2
899 .sp

901 .LP
902 \fBExample 5 \fRExecuting a Command in the Time-sharing Class
903 .sp
904 .LP
905 The following example executes \fIcommand\fR with the arguments \fIarguments\fR
906 in the time-sharing class with a user priority limit of \fB0\fR and a user
907 priority of \fB\(mi15\fR:

909 .sp
910 .in +2
```

```
.nf
example% \fBpriocntl -e -c TS -m 0 -p \fR\fB-15\fR \fB\fIcommand\fR [\fIargument
.fi
.in -2
.sp

.LP
\fBExample 6 \fRExecuting a Command in Fixed-Priority Class
.sp
.LP
The following example executes a command in the fixed-priority  class with a
user priority limit of \fB20\fR and user priority of \fB10\fR and time quantum
of \fB250\fR milliseconds:

.sp
.in +2
.nf
example% \fBpriocntl -e -c FX -m 20 -p 10 -t 250 command\fR
.fi
.in -2
.sp

.SH EXIT STATUS
.sp
.LP
The following exit values are returned:
.sp
.LP
For options \fB-d\fR, \fB-l\fR, and \fB-s\fR:
.sp
.ne 2
.na
\fB\fB0\fR\fR
.ad
.RS 5n
Successful operation.
.RE

.sp
.ne 2
.na
\fB\fB1\fR\fR
.ad
.RS 5n
Error condition.
.RE

.sp
.LP
For option \fB-e\fR:
.sp
.LP
Return of the Exit Status of the executed command denotes successful operation.
Otherwise,
.sp
.ne 2
.na
\fB\fB1\fR\fR
.ad
.RS 5n
Command could not be executed at the specified priority.
.RE

.SH ATTRIBUTES
.sp
.LP
```



```
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
CSI     Enabled
.TE

.SH SEE ALSO
.sp
.LP
\fBkill\fR(1), \fBnice\fR(1), \fBps\fR(1), \fBdispadmin\fR(1M), \fBexec\fR(2),
\fBfork\fR(2), \fBpriocntl\fR(2), \fBfx_dptbl\fR(4), \fBprocess\fR(4),
\fBrt_dptbl\fR(4), \fBattributes\fR(5), \fBzones\fR(5), \fBFSS\fR(7)
.sp
.LP
\fISystem Administration Guide: Basic Administration\fR
.SH DIAGNOSTICS
.sp
.LP
\fBpriocntl\fR prints the following error messages:
.sp
.ne 2
.na
\fB\fBProcess(es) not found\fR\fR
.ad
.sp .6
.RS 4n
None of the specified processes exists.
.RE

.sp
.ne 2
.na
\fB\fBSpecified processes from different classes\fR\fR
.ad
.sp .6
.RS 4n
The \fB-s\fR option is being used to set parameters, the \fB-c\fR \fIclass\fR
option is not present, and processes from more than one class are specified.
.RE

.sp
.ne 2
.na
\fB\fBInvalid option or argument\fR\fR
.ad
.sp .6
.RS 4n
An unrecognized or invalid option or option argument is used.
.RE
```

```
**********************************************************
   20461 Sun Sep 16 19:22:53 2018
new/usr/src/man/man1/truss.1
9842 man page typos and spelling
**********************************************************
'\" te
.\" Copyright (c) 2007, Sun Microsystems, Inc. All Rights Reserved.
.\" Copyright 1989 AT&T
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH TRUSS 1 "Jul 31, 2004"
.SH NAME
truss \- trace system calls and signals
.SH SYNOPSIS
.LP
.nf
\fBtruss\fR [\fB-fcaeildDE\fR] [\fB-\fR [tTvx] [!] \fIsyscall\fR ,...]
     [\fB-\fR [sS] [!] \fIsignal\fR ,...] [\fB-\fR [mM] [!] \fIfault\fR ,...]
     [\fB-\fR [rw] [!] \fIfd\fR ,...]
     [\fB-\fR [uU] [!] \fIlib\fR ,... : [:] [!] \fIfunc\fR ,...]
     [\fB-o\fR \fIoutfile\fR] \fIcommand\fR | \fB-p\fR \fIpid\fR[\fI/lwps\fR]...
.fi

.SH DESCRIPTION
.sp
.LP
The \fBtruss\fR utility executes the specified command and produces a trace of
the system calls it performs, the signals it receives, and the machine faults
it incurs. Each line of the trace output reports either the fault or signal
name or the system call name with its arguments and return value(s). System
call arguments are displayed symbolically when possible using defines from
relevant system headers. For any path name pointer argument, the pointed-to
string is displayed. Error returns are reported using the error code names
described in \fBIntro\fR(3). If, in the case of an error, the kernel reports a
missing privilege, a privilege name as described in \fBprivileges\fR(5) is
reported in square brackets (\fB[ ]\fR) after the error code name.
.sp
.LP
Optionally (see the \fB-u\fR option), \fBtruss\fR also produce an entry/exit
trace of user-level function calls executed by the traced process, indented to
indicate nesting.
.SH OPTIONS
.sp
.LP
For those options that take a list argument, the name \fBall\fR can be used as
a shorthand to specify all possible members of the list. If the list begins
with a \fB!\fR, the meaning of the option is negated (for example, exclude
rather than trace). Multiple occurrences of the same option can be specified.
For the same name in a list, subsequent options (those to the right) override
previous ones (those to the left).
.sp
.LP
The following options are supported:
.sp
.ne 2
.na
\fB\fB-a\fR\fR
.ad
.sp .6
.RS 4n
Shows the argument strings that are passed in each \fBexec()\fR system call.
.RE

.sp
.ne 2
```



```
.na
\fB\fB-c\fR\fR
.ad
.sp .6
.RS 4n
Counts traced system calls, faults, and signals rather than displaying the
trace line-by-line. A summary report is produced after the traced command
terminates or when \fBtruss\fR is interrupted. If \fB-f\fR is also specified,
the counts include all traced system calls, faults, and signals for child
processes.
.RE

.sp
.ne 2
.na
\fB\fB-d\fR\fR
.ad
.sp .6
.RS 4n
Includes a time stamp on each line of trace output. The time stamp appears as a
field containing \fIseconds\fR\|.\|\fIfraction\fR at the start of the line.
This represents a time in seconds relative to the beginning of the trace. The
first line of the trace output shows the base time from which the individual
time stamps are measured, both as seconds since the epoch (see \fBtime\fR(2))
and as a date string (see \fBctime\fR(3C) and \fBdate\fR(1)). The times that
are reported are the times that the event in question occurred. For all system
calls, the event is the completion of the system call, not the start of the
system call.
.RE

.sp
.ne 2
.na
\fB\fB-D\fR\fR
.ad
.sp .6
.RS 4n
Includes a time delta on each line of trace output. The value appears as a
field containing \fIseconds\fR\|.\|\fIfraction\fR and represents the elapsed
time for the \fBLWP\fR that incurred the event since the last reported event
incurred by that \fBLWP.\fR Specifically, for system calls, this is not the
time spent within the system call.
.RE

.sp
.ne 2
.na
\fB\fB-e\fR\fR
.ad
.sp .6
.RS 4n
Shows the environment strings that are passed in each \fBexec()\fR system call.
.RE

.sp
.ne 2
.na
\fB\fB-E\fR\fR
.ad
.sp .6
.RS 4n
Includes a time delta on each line of trace output. The value appears as a
field containing \fIseconds\fR\fB\&.\fR\fIfraction\fR and represents the
difference in time elapsed between the beginning and end of a system call.
.sp
In contrast to  the \fB-D\fR option, this is the amount of time spent within
```

```
126 the system call.
127 .RE

129 .sp
130 .ne 2
131 .na
132 \fB\fB-f\fR\fR
133 .ad
134 .sp .6
135 .RS 4n
136 Follows all children created by \fBfork()\fR or \fBvfork()\fR and includes
137 their signals, faults, and system calls in the trace output. Normally, only the
138 first-level command or process is traced. When \fB-f\fR is specified, the
139 process-id is included with each line of trace output to indicate which process
140 executed the system call or received the signal.
141 .RE

143 .sp
144 .ne 2
145 .na
146 \fB\fB-i\fR\fR
147 .ad
148 .sp .6
149 .RS 4n
150 Does not display interruptible sleeping system calls. Certain system calls,
151 such as \fBopen()\fR and \fBread()\fR on terminal devices or pipes, can sleep
152 for indefinite periods and are interruptible. Normally, \fBtruss\fR reports
153 such sleeping system calls if they remain asleep for more than one second. The
154 system call is reported again a second time when it completes. The \fB-i\fR
155 option causes such system calls to be reported only once, when they complete.
156 .RE

158 .sp
159 .ne 2
160 .na
161 \fB\fB-l\fR\fR
162 .ad
163 .sp .6
164 .RS 4n
165 Includes the id of the responsible lightweight process (\fILWP\fR) with each
166 line of trace output. If \fB-f\fR is also specified, both the process-id and
167 the LWP-id are included.
168 .RE

170 .sp
171 .ne 2
172 .na
173 \fB\fB-m\fR [\fB!\fR]\fIfault\fR,...\fR
174 .ad
175 .sp .6
176 .RS 4n
177 Machine faults to trace or exclude. Those faults specified in the
178 comma-separated list are traced. Faults can be specified by name or number (see
179 \fB<sys/fault.h>\fR). If the list begins with a \fB!\fR, the specified faults
180 are excluded from the trace output. Default is \fB-mall\fR \fB-m\fR
181 \fB!fltpage\fR.
182 .RE

184 .sp
185 .ne 2
186 .na
187 \fB\fB-M\fR [\fB!\fR]\fIfault\fR,...\fR
188 .ad
189 .sp .6
190 .RS 4n
191 Machine faults that stop the process. The specified faults are added to the set
```



```
192 specified by \fB-m\fR. If one of the specified faults is incurred, \fBtruss\fR
193 leaves the process stopped and abandoned (see the \fB-T\fR option). Default is
194 \fB\fR\fB-M\fR\fB!all\fR.
195 .RE

197 .sp
198 .ne 2
199 .na
200 \fB\fB-o\fR \fIoutfile\fR\fR
201 .ad
202 .sp .6
203 .RS 4n
204 File to be used for the trace output. By default, the output goes to standard
205 error.
206 .RE

208 .sp
209 .ne 2
210 .na
211 \fB\fB-p\fR\fR
212 .ad
213 .sp .6
214 .RS 4n
215 Interprets the \fIcommand\fR arguments to \fBtruss\fR as a list of process-ids
216 for existing processes (see \fBps\fR(1)) rather than as a command to be
217 executed. \fBtruss\fR takes control of each process and begins tracing it
218 provided that the userid and groupid of the process match those of the user or
219 that the user is a privileged user. Users can trace only selected threads by
220 appending \fB/\fR\fIthread-id\fR to the process-id. Mutiple threads can be
221 selected using the \fB-\fR and \fB,\fR delimiters. For example \fB/1,2,7-9\fR
222 traces threads \fB1\fR, \fB2\fR, \fB7\fR, \fB8\fR, and \fB9\fR. Processes can
223 also be specified by their names in the \fB/proc\fR directory, for example,
224 \fB/proc/12345\fR.
225 .RE

227 .sp
228 .ne 2
229 .na
230 \fB\fB-r\fR [\fB!\fR]\fIfd\fR,...\fR
231 .ad
232 .sp .6
233 .RS 4n
234 Shows the full contents of the \fBI/O\fR buffer for each \fBread()\fR on any of
235 the specified file descriptors. The output is formatted 32 bytes per line and
236 shows each byte as an \fBASCII\fR character (preceded by one blank) or as a
237 2-character C language escape sequence for control characters such as
238 horizontal tab (\|\e\|t) and newline (\|\e\|n). If \fBASCII\fR interpretation
239 is not possible, the byte is shown in 2-character hexadecimal representation.
240 (The first 12 bytes of the \fBI/O\fR buffer for each traced \fBprint >read()\fR
241 are shown even in the absence of \fB-r\fR.) Default is
242 \fB\fR\fB-r\fR\fB!all\fR.
243 .RE

245 .sp
246 .ne 2
247 .na
248 \fB\fB-s\fR [\fB!\fR]\fIsignal\fR,...\fR
249 .ad
250 .sp .6
251 .RS 4n
252 Signals to trace or exclude. Those signals specified in the comma-separated
253 list are traced. The trace output reports the receipt of each specified signal,
254 even if the signal is being ignored (not blocked). (Blocked signals are not
255 received until they are unblocked.) Signals can be specified by name or number
256 (see \fB<sys/signal.h>\fR). If the list begins with a \fB!\fR, the specified
257 signals are excluded from the trace output. Default is \fB-sall\fR.
```

```
.RE

.sp
.ne 2
.na
\fB\fB-S\fR [\fB!\fR]\fIsignal\fR,...\fR
.ad
.sp .6
.RS 4n
Signals that stop the process. The specified signals are added to the set
specified by \fB-s\fR. If one of the specified signals is received, \fBtruss\fR
leaves the process stopped and abandoned (see the \fB-T\fR option). Default is
\fB\fR\fB-S\fR\fB!all\fR.
.RE

.sp
.ne 2
.na
\fB\fB-t\fR [\fB!\fR]\fIsyscall\fR,...\fR
.ad
.sp .6
.RS 4n
System calls to trace or exclude. Those system calls specified in the
comma-separated list are traced. If the list begins with a \fB!\fR, the
specified system calls are excluded from the trace output. Default is
\fB-tall\fR.
.RE

.sp
.ne 2
.na
\fB\fB-T\fR [\fB!\fR]\fIsyscall\fR,...\fR
.ad
.sp .6
.RS 4n
Specifies system calls that stop the process. The specified system calls are
added to the set specified by \fB-t\fR. If one of the specified system calls is
encountered, \fBtruss\fR leaves the process stopped and abandoned. That is,
\fBtruss\fR releases the process and exits but leaves the process in the
stopped state at completion of the system call in question. A debugger or other
process inspection tool (see \fBproc\fR(1)) can then be applied to the stopped
process. \fBtruss\fR can be reapplied to the stopped process with the same or
different options to continue tracing. Default is \fB\fR\fB-T\fR\fB!all\fR.
.sp
A process left stopped in this manner cannot be restarted by the application of
\fBkill\fR \fB-CONT\fR because it is stopped on an event of interest via
\fB/proc\fR, not by the default action of a stopping signal (see
\fBsignal.h\fR(3HEAD)). The \fBprun\fR(1) command described in \fBproc\fR(1)
can be used to set the stopped process running again.
.RE

.sp
.ne 2
.na
\fB\fB-u\fR
[\fB!\fR]\fIlib\fR,...\fB:\fR[\fB:\fR][\fB!\fR]\fIfunc\fR,\|.\|.\|.\fR
.ad
.sp .6
.RS 4n
User-level function call tracing. \fIlib\fR,\|.\|.\|. is a comma-separated list
of dynamic library names, excluding the ''\fB\&.so.\fR\fIn\fR'' suffix.
\fIfunc\fR,\|.\|.\|. is a comma-separated list of function names. In both cases
the names can include name-matching metacharacters \fB*\fR,\fB?\fR,\fB[]\fR
with the same meanings as those of \fBsh\fR(1) but as applied to the
library/function name spaces, not to files. An empty library or function list
defaults to \fB*\fR, trace all libraries or functions in a library. A leading
```



```
\fB!\fR on either list specifies an exclusion list, names of libraries or
functions not to be traced. Excluding a library excludes all functions in that
library; any function list following a library exclusion list is ignored.
.sp
A single \fB:\fR separating the library list from the function list means to
trace calls into the libraries from outside the libraries, but omit calls made
to functions in a library from other functions in the same library. A double
\fB:\|:\fR means to trace all calls, regardless of origin.
.sp
Library patterns do not match either the executable file or the dynamic linker
unless there is an exact match (\fBl*\fR does not match \fBld.so.1\fR). To
trace functions in either of these objects, the names must be specified
exactly, as in:
.sp
.in +2
.nf
\fBtruss -u a.out -u ld ...\fR
.fi
.in -2
.sp

\fBa.out\fR is the literal name to be used for this purpose; it does not stand
for the name of the executable file. Tracing \fBa.out\fR function calls implies
all calls (default is \fB::\fR).
.sp
Multiple \fB-u\fR options can be specified and they are honored left-to-right.
The id of the thread that performed the function call is included in the trace
output for the call. \fBtruss\fR searches the dynamic symbol table in each
library to find function names and also searches the standard symbol table if
it has not been stripped.
.RE

.sp
.ne 2
.na
\fB\fB-U\fR
[\fB!\fR]\fIlib\fR,\|.\|.\|.\|\fB:\fR[\fB:\fR][\fB!\fR]\fIfunc\fR,\|.\|.\|.\fR
.ad
.sp .6
.RS 4n
User-level function calls that stop the process. The specified functions are
added to the set specified by \fB-u\fR. If one of the specified functions is
called, \fBtruss\fR leaves the process stopped and abandoned (see the \fB-T\fR
option).
.RE

.sp
.ne 2
.na
\fB\fB-v\fR [\fB!\fR]\fIsyscall\fR,...\fR
.ad
.sp .6
.RS 4n
Verbose. Displays the contents of any structures passed by address to the
specified system calls (if traced by \fB-t\fR). Input values as well as values
returned by the operating system are shown. For any field used as both input
and output, only the output value is shown. Default is
\fB\fR\fB-v\fR\fB!all\fR.
.RE

.sp
.ne 2
.na
\fB\fB-w\fR [\fB!\fR]\fIfd\fR,...\fR
.ad
.sp .6
```

```
.RS 4n
Shows the contents of the I/O buffer for each \fBwrite()\fR on any of the
specified file descriptors (see the \fB-r\fR option). Default is
\fB\fR\fB-w\fR\fB!all\fR.
.RE

.sp
.ne 2
.na
\fB\fB-x\fR [\fB!\fR]\fIsyscall\fR,...\fR
.ad
.sp .6
.RS 4n
Displays the arguments to the specified system calls (if traced by \fB-t\fR) in
raw form, usually hexadecimal, rather than symbolically. This is for unredeemed
hackers who must see the raw bits to be happy. Default is
\fB\fR\fB-x\fR\fB!all\fR.
.RE

.sp
.LP
See \fIman pages section 2: System Calls\fR for system call names accepted by
the \fB-t\fR, \fB-T\fR, \fB-v\fR, and \fB-x\fR options. System call numbers are
also accepted.
.sp
.LP
If \fBtruss\fR is used to initiate and trace a specified command and if the
\fB-o\fR option is used or if standard error is redirected to a non-terminal
file, then \fBtruss\fR runs with hangup, interrupt, and quit signals ignored.
This facilitates tracing of interactive programs that catch interrupt and quit
signals from the terminal.
.sp
.LP
If the trace output remains directed to the terminal, or if existing processes
are traced (the \fB-p\fR option), then \fBtruss\fR responds to hangup,
interrupt, and quit signals by releasing all traced processes and exiting. This
enables the user to terminate excessive trace output and to release
previously-existing processes. Released processes continue normally, as though
they had never been touched.
.SH EXAMPLES
.LP
\fBExample 1 \fRTracing a Command
.sp
.LP
The following example produces a trace of the \fBfind\fR(1) command on the
terminal:

.sp
.in +2
.nf
example$ \fBtruss find . -print >find.out\fR
.fi
.in -2
.sp

.LP
\fBExample 2 \fRTracing Common System Calls
.sp
.LP
The following example shows only a trace of the open, close, read, and write
system calls:

.sp
.in +2
.nf
example$ \fBtruss -t open,close,read,write find . -print >find.out\fR
```



```
.fi
.in -2
.sp

.LP
\fBExample 3 \fRTracing a Shell Script
.sp
.LP
The following example produces a trace of the \fBspell\fR(1) command on the
file \fBtruss.out\fR:

.sp
.in +2
.nf
example$ \fBtruss -f -o truss.out spell \fIdocument\fR\fR
.fi
.in -2
.sp

.sp
.LP
\fBspell\fR is a shell script, so the \fB-f\fR flag is needed to trace not only
the shell but also the processes created by the shell. (The spell script runs a
pipeline of eight processes.)

.LP
\fBExample 4 \fRAbbreviating Output
.sp
.LP
The following example abbreviates output:
The following example abreviates output:

.sp
.in +2
.nf
example$ \fBtruss nroff -mm \fIdocument\fR >nroff.out\fR
.fi
.in -2
.sp

.sp
.LP
because 97% of the output reports \fBlseek()\fR, \fBread()\fR, and
\fBwrite()\fR system calls. To abbreviate it:

.sp
.in +2
.nf
example$ \fBtruss -t !lseek,read,write nroff -mm \fIdocument\fR >nroff.out\fR
.fi
.in -2
.sp

.LP
\fBExample 5 \fRTracing Library Calls From Outside the C Library
.sp
.LP
The following example traces all user-level calls made to any function in the C
library from outside the C library:

.sp
.in +2
.nf
example$ \fBtruss -u libc ...\fR
.fi
.in -2
```

```
.sp

.LP
\fBExample 6 \fRTracing library calls from within the C library
.sp
.LP
The following example includes calls made to functions in the C library from
within the C library itself:

.sp
.in +2
.nf
example$ \fBtruss -u libc:: ...\fR
.fi
.in -2
.sp

.LP
\fBExample 7 \fRTracing Library Calls Other Than the C Library
.sp
.LP
The following example traces all user-level calls made to any library other
than the C library:

.sp
.in +2
.nf
example$ \fBtruss -u '*' -u !libc ...\fR
.fi
.in -2
.sp

.LP
\fBExample 8 \fRTracing \fBprintf\fR and \fBscanf\fR Function Calls
.sp
.LP
The following example traces all user-level calls to functions in the printf
and scanf family contained in the C library:

.sp
.in +2
.nf
example$ \fBtruss -u 'libc:*printf,*scanf' ...\fR
.fi
.in -2
.sp

.LP
\fBExample 9 \fRTracing Every User-level Function Call
.sp
.LP
The following example traces every user-level function call from anywhere to
anywhere:

.sp
.in +2
.nf
example$ \fBtruss -u a.out -u ld:: -u :: ...\fR
.fi
.in -2
.sp

.LP
\fBExample 10 \fRTracing a System Call Verbosely
.sp
.LP
```



```
The following example verbosely traces the system call activity of process #1,
\fBinit\fR(1M) (if you are a privileged user):

.sp
.in +2
.nf
example# \fBtruss -p -v all 1\fR
.fi
.in -2
.sp

.sp
.LP
Interrupting \fBtruss\fR returns \fBinit\fR to normal operation.

.SH FILES
.sp
.ne 2
.na
\fB\fB/proc/*\fR\fR
.ad
.RS 11n
Process files
.RE

.SH SEE ALSO
.sp
.LP
\fBdate\fR(1), \fBfind\fR(1), \fBproc\fR(1), \fBps\fR(1), \fBsh\fR(1),
\fBspell\fR(1), \fBinit\fR(1M), \fBIntro\fR(3), \fBexec\fR(2), \fBfork\fR(2),
\fBlseek\fR(2), \fBopen\fR(2), \fBread\fR(2), \fBtime\fR(2), \fBvfork\fR(2),
\fBwrite\fR(2), \fBctime\fR(3C), \fBsignal.h\fR(3HEAD), \fBproc\fR(4),
\fBattributes\fR(5), \fBprivileges\fR(5), \fBthreads\fR(5)
.sp
.LP
\fIman pages section 2: System Calls\fR
.SH NOTES
.sp
.LP
Some of the system calls described in \fIman pages section 2: System Calls\fR
differ from the actual operating system interfaces. Do not be surprised by
minor deviations of the trace output from the descriptions in that document.
.sp
.LP
Every machine fault (except a page fault) results in the posting of a signal to
the \fBLWP\fR that incurred the fault. A report of a received signal
immediately follows each report of a machine fault (except a page fault) unless
that signal is being blocked.
.sp
.LP
The operating system enforces certain security restrictions on the tracing of
processes. In particular, any command whose object file (\fBa.out\fR) cannot be
read by a user cannot be traced by that user; set-uid and set-gid commands can
be traced only by a privileged user. Unless it is run by a privileged user,
\fBtruss\fR loses control of any process that performs an \fBexec()\fR of a
set-id or unreadable object file; such processes continue normally, though
independently of \fBtruss\fR, from the point of the \fBexec()\fR.
.sp
.LP
To avoid collisions with other controlling processes, \fBtruss\fR does not
trace a process that it detects is being controlled by another process via the
\fB/proc\fR interface. This allows \fBtruss\fR to be applied to
\fBproc\fR(4)-based debuggers as well as to another instance of itself.
.sp
.LP
The trace output contains tab characters under the assumption that standard tab
```

```
650 stops are set (every eight positions).
651 .sp
652 .LP
653 The trace output for multiple processes or for a multithreaded process (one
654 that contains more than one \fBLWP)\fR is not produced in strict time order.
655 For example, a \fBread()\fR on a pipe can be reported before the corresponding
656 \fBwrite()\fR. For any one \fBLWP\fR (a traditional process contains only one),
657 the output is strictly time-ordered.
658 .sp
659 .LP
660 When tracing more than one process, \fBtruss\fR runs as one controlling process
661 for each process being traced. For the example of the \fBspell\fR command shown
662 above, \fBspell\fR itself uses 9 process slots, one for the shell and 8 for the
663 8-member pipeline, while \fBtruss\fR adds another 9 processes, for a total of
664 18.
665 .sp
666 .LP
667 Not all possible structures passed in all possible system calls are displayed
668 under the \fB-v\fR option.
```

```
**********************************************************
   26978 Sun Sep 16 19:22:53 2018
new/usr/src/man/man1m/th_define.1m
9842 man page typos and spelling
**********************************************************
   1 '\" te
   2 .\" Copyright (c) 2001 Sun Microsystems, Inc. All Rights Reserved
   3 .\" The contents of this file are subject to the terms of the Common Development
   4 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
   5 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
   6 .TH TH_DEFINE 1M "April 9, 2016"
   7 .SH NAME
   8 th_define \- create fault injection test harness error specifications
   9 .SH SYNOPSIS
  10 .LP
  11 .nf
  12 \fBth_define\fR [\fB-n\fR \fIname\fR \fB-i\fR \fIinstance\fR| \fB-P\fR \fIpath\f
  13      [\fB-r\fR \fIreg_number\fR] [\fB-l\fR \fIoffset\fR [\fIlength\fR]]
  14      [\fB-c\fR \fIcount\fR [\fIfailcount\fR]] [\fB-o\fR \fIoperator\fR [\fIopera
  15      [\fB-f\fR \fIacc_chk\fR] [\fB-w\fR \fImax_wait_period\fR [\fIreport_interva
  16 .fi

  18 .LP
  19 .nf
  20 \fBor\fR
  21 .fi

  23 .LP
  24 .nf
  25 \fBth_define\fR [\fB-n\fR \fIname\fR \fB-i\fR \fIinstance\fR| \fB-P\fR \fIpath\f
  26      [\fB-a\fR log [\fIacc_types\fR] [\fB-r\fR \fIreg_number\fR] [\fB-l\fR \fIof
  27      [\fB-c\fR \fIcount\fR [\fIfailcount\fR]] [\fB-s\fR \fIcollect_time\fR] [\fB
  28      [\fB-x\fR \fIflags\fR] [\fB-C\fR \fIcomment_string\fR]
  29      [\fB-e\fR \fIfixup_script\fR [\fIargs\fR]]
  30 .fi

  32 .LP
  33 .nf
  34 \fBor\fR
  35 .fi

  37 .LP
  38 .nf
  39 \fBth_define\fR [\fB-h\fR]
  40 .fi

  42 .SH DESCRIPTION
  43 .LP
  44 The \fBth_define\fR utility provides an interface to the \fBbus_ops\fR fault
  45 injection \fBbofi\fR device driver for defining error injection specifications
  46 (referred to as errdefs). An errdef corresponds to a specification of how to
  47 corrupt a device driver's accesses to its hardware. The command line arguments
  48 determine the precise nature of the fault to be injected. If the supplied
  49 arguments define a consistent errdef, the \fBth_define\fR process will store
  50 the errdef with the \fBbofi\fR driver and suspend itself until the criteria
  51 given by the errdef become satisfied (in practice, this will occur when the
  52 access counts go to zero).
  53 .sp
  54 .LP
  55 You use the \fBth_manage\fR(1M) command with the \fBstart\fR option to activate
  56 the resulting errdef. The effect of \fBth_manage\fR with the \fBstart\fR option
  57 is that the \fBbofi\fR driver acts upon the errdef by matching the number of
  58 hardware accesses\(emspecified in \fIcount\fR, that are of the type specified
  59 in \fIacc_types\fR, made by instance number \fIinstance\fR\(emof the driver
  60 whose name is \fIname\fR, (or by the driver instance specified by \fIpath\fR)
  61 to the register set (or DMA handle) specified by \fIreg_number\fR, that lie
```



```
  62 within the range \fIoffset\fR to \fIoffset\fR +\fI length\fR from the beginning
  63 of the register set or DMA handle. It then applies \fIoperator\fR and
  64 \fIoperand\fR to the next \fIfailcount\fR matching accesses.
  65 .sp
  66 .LP
  67 If \fIacc_types\fR includes \fBlog\fR, \fBth_define\fR runs in automatic test
  68 script generation mode, and a set of test scripts (written in the Korn shell)
  69 is created and placed in a sub-directory of the current directory with the name
  70 \fB\fI<driver>\fR\&.test.\fI<id>\fR\fR (for example, \fBglm.test.978177106\fR).
  71 A separate, executable script is generated for each access handle that matches
  72 the logging criteria. The log of accesses is placed at the top of each script
  73 as a record of the session. If the current directory is not writable, file
  74 output is written to standard output. The base name of each test file is the
  75 driver name, and the extension is a number that discriminates between different
  76 access handles. A control script (with the same name as the created test
  77 directory) is generated that will run all the test scripts sequentially.
  78 .sp
  79 .LP
  80 Executing the scripts will install, and then activate, the resulting error
  81 definitions. Error definitions are activated sequentially and the driver
  82 instance under test is taken offline and brought back online before each test
  83 (refer to the \fB-e\fR option for more information). By default, logging
  84 applies to all \fBPIO\fR accesses, all interrupts, and all DMA accesses to and
  85 from areas mapped for both reading and writing. You can constrain logging by
  86 specifying additional \fIacc_types\fR, \fIreg_number\fR, \fIoffset\fR and
  87 \fIlength\fR. Logging will continue for \fIcount\fR matching accesses, with an
  88 optional time limit of \fIcollect_time\fR seconds.
  89 .sp
  90 .LP
  91 Either the \fB-n\fR or \fB-P\fR option must be provided. The other options are
  92 optional. If an option (other than \fB-a\fR) is specified multiple times, only
  93 the final value for the option is used. If an option is not specified, its
  94 associated value is set to an appropriate default, which will provide maximal
  95 error coverage as described below.
  96 .SH OPTIONS
  97 .LP
  98 The following options are available:
  99 .sp
 100 .ne 2
 101 .na
 102 \fB\fB-n\fR \fIname\fR \fR
 103 .ad
 104 .sp .6
 105 .RS 4n
 106 Specify the name of the driver to test. (String)
 107 .RE

 109 .sp
 110 .ne 2
 111 .na
 112 \fB\fB-i\fR\fI instance\fR \fR
 113 .ad
 114 .sp .6
 115 .RS 4n
 116 Test only the specified driver instance (-1 matches all instances of driver).
 117 (Numeric)
 118 .RE

 120 .sp
 121 .ne 2
 122 .na
 123 \fB\fB-P\fR\fI path\fR \fR
 124 .ad
 125 .sp .6
 126 .RS 4n
 127 Specify the full device path of the driver to test. (String)
```

```
.RE

.sp
.ne 2
.na
\fB\fB-r\fR \fIreg_number\fR \fR
.ad
.sp .6
.RS 4n
Test only the given register set or DMA handle (-1 matches all register sets
and DMA handles). (Numeric)
.RE

.sp
.ne 2
.na
\fB\fB-a\fR\fI acc_types\fR \fR
.ad
.sp .6
.RS 4n
Only the specified access types will be matched. Valid values for the
\fIacc_types\fR argument are \fBlog\fR, \fBpio\fR, \fBpio_r\fR, \fBpio_w\fR,
\fBdma\fR, \fBdma_r\fR, \fBdma_w\fR and \fBintr\fR. Multiple access types,
separated by spaces, can be specified. The default is to match all hardware
accesses.
.sp
If \fIacc_types\fR is set to \fBlog\fR, logging will match all \fBPIO\fR
accesses, interrupts and DMA accesses to and from areas mapped for both reading
and writing. \fBlog\fR can be combined with other \fIacc_types\fR, in which
case the matching condition for logging will be restricted to the specified
additional \fIacc_types\fR. Note that \fBdma_r\fR will match only DMA handles
mapped for reading only; \fBdma_w\fR will match only DMA handles mapped for
writing only; \fBdma\fR will match only DMA handles mapped for both reading and
writing.
.RE

.sp
.ne 2
.na
\fB\fB-l\fR \fIoffset \fR\fB[\fR\fIlength\fR\fB]\fR\fR
.ad
.sp .6
.RS 4n
Constrain the range of qualifying accesses. The \fIoffset\fR and \fIlength\fR
arguments indicate that any access of the type specified with the \fB-a\fR
option, to the register set or DMA handle specified with the \fB-r\fR option,
lie at least \fIoffset\fR bytes into the register set or DMA handle and at most
\fIoffset\fR + \fIlength\fR bytes into it. The default for \fIoffset\fR is 0.
The default for \fIlength\fR is the maximum value that can be placed in an
\fBoffset_t\fR C data type (see \fBtypes.h\fR). Negative values are converted
into unsigned quantities. Thus, \fB\fR\fBth_define\fR\fB \fR\fB-l\fR 0 \fB-1\fR
is maximal.
.RE

.sp
.ne 2
.na
\fB\fB-c\fR \fIcount\fR\fB[\fR\fIfailcount\fR\fB]\fR \fR
.ad
.sp .6
.RS 4n
Wait for \fIcount\fR number of matching accesses, then apply an operator and
operand (see the \fB-o\fR option) to the next \fIfailcount\fR number of
matching accesses. If the access type (see the \fB-a\fR option) includes
logging, the number of logged accesses is given by \fIcount\fR +
\fIfailcount\fR - 1. The -1 is required because the last access coincides with
```



```
the first faulting access.
.sp
Note that access logging may be combined with error injection if
\fIfailcount\fR and \fIoperator\fR are nonzero and if the access type includes
logging and any of the other access types (\fBpio\fR, \fBdma\fR and \fBintr\fR)
See the description of access types in the definition of the \fB-a\fR option,
above.
.sp
When the \fIcount\fR and \fIfailcount\fR fields reach zero, the status of the
errdef is reported to standard output. When all active errdefs created by the
\fBth_define\fR process complete, the process exits. If \fIacc_types\fR
includes \fBlog\fR, \fIcount\fR determines how many accesses to log. If
\fIcount\fR is not specified, a default value is used. If \fIfailcount\fR is
set in this mode, it will simply increase the number of accesses logged by a
further \fIfailcount\fR - 1.
.RE

.sp
.ne 2
.na
\fB\fB-o\fR\fI operator \fR\fB[\fR\fIoperand\fR\fB]\fR \fR
.ad
.sp .6
.RS 4n
For qualifying PIO read and write accesses, the value read from or written to
the hardware is corrupted according to the value of \fIoperator\fR:
.sp
.ne 2
.na
\fB\fBEQ\fR\fR
.ad
.RS 7n
\fIoperand\fR is returned to the driver.
.RE

.sp
.ne 2
.na
\fB\fBOR\fR\fR
.ad
.RS 7n
\fIoperand\fR is bitwise ORed with the real value.
.RE

.sp
.ne 2
.na
\fB\fBAND\fR\fR
.ad
.RS 7n
\fIoperand\fR is bitwise ANDed with the real value.
.RE

.sp
.ne 2
.na
\fB\fBXOR\fR\fR
.ad
.RS 7n
\fIoperand\fR is bitwise XORed with the real value.
.RE

For PIO write accesses, the following operator is allowed:
.sp
.ne 2
.na
```

```
\fB\fBNO\fR\fR
.ad
.RS 6n
Simply ignore the driver’s attempt to write to the hardware.
.RE

Note that a driver performs PIO via the \fBddi_get\fIX\fR()\fR,
\fBddi_put\fIX\fR()\fR, \fBddi_rep_get\fIX\fR()\fR and
\fBddi_rep_put\fIX\fR()\fR routines (where \fIX\fR is 8, 16, 32 or 64).
Accesses made using \fBddi_get\fIX\fR()\fR and \fBddi_put\fIX\fR()\fR are
treated as a single access, whereas an access made using the
\fBddi_rep_*\fR(9F) routines are broken down into their respective number of
accesses, as given by the \fIrepcount\fR parameter to these DDI calls. If the
access is performed via a DMA handle, \fIoperator\fR and \fIvalue\fR are
applied to every access that comprises the DMA request. If interference with
interrupts has been requested then the operator may take any of the following
values:
.sp
.ne 2
.na
\fB\fBDELAY\fR\fR
.ad
.RS 9n
After \fIcount\fR accesses (see the \fB-c\fR option), delay delivery of the
next \fIfailcount\fR number of interrupts for \fIoperand\fR number of
microseconds.
.RE

.sp
.ne 2
.na
\fB\fBLOSE\fR\fR
.ad
.RS 9n
After \fIcount\fR number of interrupts, fail to deliver the next
\fIfailcount\fR number of real interrupts to the driver.
.RE

.sp
.ne 2
.na
\fB\fBEXTRA\fR\fR
.ad
.RS 9n
After \fIcount\fR number of interrupts, start delivering \fIoperand\fR number
of extra interrupts for the next \fIfailcount\fR number of real interrupts.
.RE

The default value for \fIoperand\fR and \fIoperator\fR is to corrupt the data
access by flipping each bit (XOR with -1).
.RE

.sp
.ne 2
.na
\fB\fB-f\fR \fIacc_chk\fR\fR
.ad
.sp .6
.RS 4n
If the \fIacc_chk\fR parameter is set to 1 or \fBpio\fR, then the driver’s
calls to \fBddi_check_acc_handle\fR(9F) return \fBDDI_FAILURE\fR when the
access count goes to 1. If the \fIacc_chk\fR parameter is set to 2 or
\fBdma\fR, then the driver’s calls to \fBddi_check_dma_handle\fR(9F) return
\fBDDI_FAILURE\fR when the access count goes to 1.
.RE
```



```
.sp
.ne 2
.na
\fB\fB-w\fR \fImax_wait_period\fR\fB [\fR\fIreport_interval\fR\fB]\fR \fR
.ad
.sp .6
.RS 4n
Constrain the period for which an error definition will remain active. The
option applies only to non-logging errdefs. If an error definition remains
active for \fImax_wait_period\fR seconds, the test will be aborted. If
\fIreport_interval\fR is set to a nonzero value, the current status of the
error definition is reported to standard output every \fIreport_interval\fR
seconds. The default value is zero. The status of the errdef is reported in
parsable format (eight fields, each separated by a colon (\fB:\fR) character,
the last of which is a string enclosed by double quotes and the remaining seven
fields are integers):
.sp
\fIft\fR:\fImt\fR:\fIac\fR:\fIfc\fR:\fIchk\fR:\fIec\fR:\fIs\fR:\fI"message"\fR
which are defined as follows:
.sp
.ne 2
.na
\fB\fIft\fR\fR
.ad
.RS 13n
The UTC time when the fault was injected.
.RE

.sp
.ne 2
.na
\fB\fImt\fR\fR
.ad
.RS 13n
The UTC time when the driver reported the fault.
.RE

.sp
.ne 2
.na
\fB\fIac\fR\fR
.ad
.RS 13n
The number of remaining non-faulting accesses.
.RE

.sp
.ne 2
.na
\fB\fIfc\fR\fR
.ad
.RS 13n
The number of remaining faulting accesses.
.RE

.sp
.ne 2
.na
\fB\fIchk\fR\fR
.ad
.RS 13n
The value of the \fIacc_chk\fR field of the errdef.
.RE

.sp
.ne 2
```

```
.na
\fB\fIec\fR\fR
.ad
.RS 13n
The number of fault reports issued by the driver against this errdef (\fImt\fR
holds the time of the initial report).
.RE

.sp
.ne 2
.na
\fB\fIs\fR\fR
.ad
.RS 13n
The severity level reported by the driver.
.RE

.sp
.ne 2
.na
\fB\fI"message"\fR\fR
.ad
.RS 13n
Textual reason why the driver has reported a fault.
.RE

.RE

.sp
.ne 2
.na
\fB\fB-h\fR\fR
.ad
.sp .6
.RS 4n
Display the command usage string.
.RE

.sp
.ne 2
.na
\fB\fB-s\fR \fIcollect_time\fR \fR
.ad
.sp .6
.RS 4n
If \fIacc_types\fR is given with the \fB-a\fR option and includes \fBlog\fR,
the errdef will log accesses for \fIcollect_time\fR seconds (the default is to
log until the log becomes full). Note that, if the errdef specification matches
multiple driver handles, multiple logging errdefs are registered with the
\fBbofi\fR driver and logging terminates when all logs become full or when
\fIcollect_time\fR expires or when the associated errdefs are cleared. The
current state of the log can be checked with the \fBth_manage\fR(1M) command,
using the \fBbroadcast\fR parameter. A log can be terminated by running
\fBth_manage\fR(1M) with the \fBclear_errdefs\fR option or by sending a
\fBSIGALRM\fR signal to the \fBth_define\fR process. See \fBalarm\fR(2) for the
semantics of \fBSIGALRM\fR.
.RE

.sp
.ne 2
.na
\fB\fB-p\fR \fIpolicy\fR\fR
.ad
.sp .6
.RS 4n
Applicable when the \fIacc_types\fR option includes \fBlog\fR. The parameter
```



```
modifies the policy used for converting from logged accesses to errdefs. All
policies are inclusive:
.RS +4
.TP
.ie t \(bu
.el o
Use \fBrare\fR to bias error definitions toward rare accesses (default).
.RE
.RS +4
.TP
.ie t \(bu
.el o
Use \fBoperator\fR to produce a separate error definition for each operator
type (default).
.RE
.RS +4
.TP
.ie t \(bu
.el o
Use \fBcommon\fR to bias error definitions toward common accesses.
.RE
.RS +4
.TP
.ie t \(bu
.el o
Use \fBmedian\fR to bias error definitions toward median accesses.
.RE
.RS +4
.TP
.ie t \(bu
.el o
Use \fBmaximal\fR to produce multiple error definitions for duplicate accesses.
.RE
.RS +4
.TP
.ie t \(bu
.el o
Use \fBunbiased\fR to create unbiased error definitions.
.RE
.RS +4
.TP
.ie t \(bu
.el o
Use \fBonebyte\fR, \fBtwobyte\fR, \fBfourbyte\fR, or \fBeightbyte\fR to select
errdefs corresponding to 1, 2, 4 or 8 byte accesses (if chosen, the
\fB-x\fR\fBr\fR option is enforced in order to ensure that \fBddi_rep_*()\fR
calls are decomposed into \fBmultiple single accesses\fR).
.RE
.RS +4
.TP
.ie t \(bu
.el o
Use \fBmultibyte\fR to create error definitions for multibyte accesses
performed using \fBddi_rep_get*()\fR and \fBddi_rep_put*()\fR.
.RE
Policies can be combined by adding together these options. See the NOTES
section for further information.
.RE

.sp
.ne 2
.na
\fB\fB-x\fR \fIflags\fR\fR
.ad
.sp .6
.RS 4n
```

```
524 Applicable when the \fIacc_types\fR option includes \fBlog\fR. The \fIflags\fR
525 parameter modifies the way in which the \fBbofi\fR driver logs accesses. It is
526 specified as a string containing any combination of the following letters:
527 .sp
528 .ne 2
529 .na
530 \fB\fBw\fR\fR
531 .ad
532 .RS 5n
533 Continuous logging (that is, the log will wrap when full).
534 .RE

536 .sp
537 .ne 2
538 .na
539 \fB\fBt\fR\fR
540 .ad
541 .RS 5n
542 Timestamp each log entry (access times are in seconds).
543 .RE

545 .sp
546 .ne 2
547 .na
548 \fB\fBr\fR\fR
549 .ad
550 .RS 5n
551 Log repeated I/O as individual accesses (for example, a \fBddi_rep_get16\fR(9F)
552 call which has a repcount of \fIN\fR is logged \fIN\fR times with each
553 transaction logged as size 2 bytes. Without this option, the default logging
554 behavior is to log this access once only, with a transaction size of twice the
555 \fIrepcount\fR).
556 .RE

558 .RE

560 .sp
561 .ne 2
562 .na
563 \fB\fB-C\fR \fIcomment_string\fR\fR
564 .ad
565 .sp .6
566 .RS 4n
567 Applicable when the \fIacc_types\fR option includes \fBlog\fR. It provides a
568 comment string to be placed in any generated test scripts. The string must be
569 enclosed in double quotes.
570 .RE

572 .sp
573 .ne 2
574 .na
575 \fB\fB-e\fR \fIfixup_script\fR \fB[\fR\fIargs\fR\fB]\fR \fR
576 .ad
577 .sp .6
578 .RS 4n
579 Applicable when the \fIacc_types\fR option includes \fBlog\fR. The output of a
580 logging errdefs is to generate a test script for each driver access handle. Use
581 this option to embed a command in the resulting script before the errors are
582 injected. The generated test scripts will take an instance offline and bring it
583 back online before injecting errors in order to bring the instance into a known
584 fault-free state. The executable \fIfixup_script\fR will be called twice with
585 the set of optional \fIargs\fR\(em once just before the instance is taken
586 offline and again after the instance has been brought online. The following
587 variables are passed into the environment of the called executable:
588 .sp
589 .ne 2
```



```
590 .na
591 \fB\fBDRIVER_PATH\fR\fR
592 .ad
593 .RS 22n
594 Identifies the device path of the instance.
595 .RE

597 .sp
598 .ne 2
599 .na
600 \fB\fBDRIVER_INSTANCE\fR\fR
601 .ad
602 .RS 22n
603 Identifies the instance number of the device.
604 .RE

606 .sp
607 .ne 2
608 .na
609 \fB\fBDRIVER_UNCONFIGURE\fR\fR
610 .ad
611 .RS 22n
612 Has the value 1 when the instance is about to be taken offline.
613 .RE

615 .sp
616 .ne 2
617 .na
618 \fB\fBDRIVER_CONFIGURE\fR\fR
619 .ad
620 .RS 22n
621 Has the value 1 when the instance has just been brought online.
622 .RE

624 Typically, the executable ensures that the device under test is in a suitable
625 state to be taken offline (unconfigured) or in a suitable state for error
626 injection (for example configured, error free and servicing a workload). A
627 minimal script for a network driver could be:
628 .sp
629 .in +2
630 .nf
631 #!/bin/ksh

633 driver=xyznetdriver
634 ifnum=$driver$DRIVER_INSTANCE

636 if [[ $DRIVER_CONFIGURE = 1 ]]; then
637         ifconfig $ifnum plumb
638         ifconfig $ifnum ...
639         ifworkload start $ifnum
640 elif [[ $DRIVER_UNCONFIGURE = 1 ]]; then
641         ifworkload stop $ifnum
642         ifconfig $ifnum down
643         ifconfig $ifnum unplumb
644 fi
645 exit $?
646 .fi
647 .in -2
648 .sp

650 The \fB-e\fR option must be the last option on the command line.
651 .RE

653 .sp
654 .LP
655 If the \fB-a\fR \fBlog\fR option is selected but the \fB-e\fR option is not
```

```
656 given, a default script is used. This script repeatedly attempts to detach and
657 then re-attach the device instance under test.
658 .SH EXAMPLES
659 .SS "Examples of Error Definitions"
660 .LP
661 \fBth_define -n foo -i 1 -a log\fR
662 .sp
663 .LP
664 Logs all accesses to all handles used by instance 1 of the \fBfoo\fR driver
665 while running the default workload (attaching and detaching the instance). Then
666 generates a set of test scripts to inject appropriate errdefs while running
667 that default workload.
668 .sp
669 .LP
670 \fBth_define -n foo -i 1 -a log pio\fR
671 .sp
672 .LP
673 Logs PIO accesses to each PIO handle used by instance 1 of the \fBfoo\fR driver
674 while running the default workload (attaching and detaching the instance). Then
675 generates a set of test scripts to inject appropriate errdefs while running
676 that default workload.
677 .sp
678 .LP
679 \fBth_define -n foo -i 1 -p onebyte median -e fixup arg -now\fR
680 .sp
681 .LP
682 Logs all accesses to all handles used by instance 1 of the \fBfoo\fR driver
683 while running the workload defined in the fixup script \fBfixup\fR with
684 arguments \fBarg\fR and \fB-now\fR. Then generates a set of test scripts to
685 inject appropriate errdefs while running that workload. The resulting error
686 definitions are requested to focus upon single byte accesses to locations that
687 are accessed a \fBmedian\fR number of times with respect to frequency of access
688 to I/O addresses.
689 .sp
690 .LP
691 \fBth_define -n se -l 0x20 1 -a pio_r -o OR 0x4 -c 10 1000\fR
692 .sp
693 .LP
694 Simulates a stuck serial chip command by forcing 1000 consecutive read accesses
695 made by any instance of the \fBse\fR driver to its command status register,
696 thereby returning status busy.
697 .sp
698 .LP
699 \fBth_define -n foo -i 3 -r 1 -a pio_r -c 0 1 -f 1 -o OR 0x100\fR
700 .sp
701 .LP
702 Causes 0x100 to be ORed into the next physical I/O read access from any
703 register in register set 1 of instance 3 of the \fBfoo\fR driver. Subsequent
704 calls in the driver to \fBddi_check_acc_handle()\fR return \fBDDI_FAILURE\fR.
705 .sp
706 .LP
707 \fBth_define -n foo -i 3 -r 1 -a pio_r -c 0 1 -o OR 0x0\fR
708 .sp
709 .LP
710 Causes 0x0 to be ORed into the next physical I/O read access from any register
711 in register set 1 of instance 3 of the \fBfoo\fR driver. This is of course a
712 no-op.
713 .sp
714 .LP
715 \fBth_define -n foo -i 3 -r 1 -l 0x8100 1 -a pio_r -c 0 10 -o EQ 0x70003\fR
716 .sp
717 .LP
718 Causes the next ten next physical I/O reads from the register at offset 0x8100
719 in register set 1 of instance 3 of the \fBfoo\fR driver to return 0x70003.
720 .sp
721 .LP
```



```
722 \fBth_define -n foo -i 3 -r 1 -l 0x8100 1 -a pio_w -c 100 3 -o AND
723 0xfffffffffffffefff\fR
724 .sp
725 .LP
726 The next 100 physical I/O writes to the register at offset 0x8100 in register
727 set 1 of instance 3 of the \fBfoo\fR driver take place as normal. However, on
728 each of the three subsequent accesses, the 0x1000 bit will be cleared.
729 .sp
730 .LP
731 \fBth_define -n foo -i 3 -r 1 -l 0x8100 0x10 -a pio_r -c 0 1 -f 1 -o XOR 7\fR
732 .sp
733 .LP
734 Causes the bottom three bits to have their values toggled for the next physical
735 I/O read access to registers with offsets in the range 0x8100 to 0x8110 in
736 register set 1 of instance 3 of the \fBfoo\fR driver. Subsequent calls in the
737 driver to \fBddi_check_acc_handle()\fR return \fBDDI_FAILURE\fR.
738 .sp
739 .LP
740 \fBth_define -n foo -i 3 -a pio_w -c 0 1 -o NO 0\fR
741 .sp
742 .LP
743 Prevents the next physical I/O write access to any register in any register set
744 of instance 3 of the \fBfoo\fR driver from going out on the bus.
745 .sp
746 .LP
747 \fBth_define -n foo -i 3 -l 0 8192 -a dma_r -c 0 1 -o OR 7\fR
748 .sp
749 .LP
750 Causes 0x7 to be ORed into each \fBlong long\fR in the first 8192 bytes of the
751 next DMA read, using any DMA handle for instance 3 of the \fBfoo\fR driver.
752 .sp
753 .LP
754 \fBth_define -n foo -i 3 -r 2 -l 0 8 -a dma_r -c 0 1 -o OR
755 0x7070707070707070\fR
756 .sp
757 .LP
758 Causes 0x70 to be ORed into each byte of the first \fBlong long\fR of the next
759 DMA read, using the DMA handle with sequential allocation number 2 for instance
760 3 of the \fBfoo\fR driver.
761 .sp
762 .LP
763 \fBth_define -n foo -i 3 -l 256 256 -a dma_w -c 0 1 -f 2 -o OR 7\fR
764 .sp
765 .LP
766 Causes 0x7 to be ORed into each \fBlong long\fR in the range from offset 256 to
767 offset 512 of the next DMA write, using any DMA handle for instance 3 of the
768 \fBfoo\fR driver. Subsequent calls in the driver to
769 \fBddi_check_dma_handle()\fR return \fBDDI_FAILURE\fR.
770 .sp
771 .LP
772 \fBth_define -n foo -i 3 -r 0 -l 0 8 -a dma_w -c 100 3 -o AND
773 0xfffffffffffffefff\fR
774 .sp
775 .LP
776 The next 100 DMA writes using the DMA handle with sequential allocation number
777 0 for instance 3 of the \fBfoo\fR driver take place as normal. However, on each
778 of the three subsequent accesses, the 0x1000 bit will be cleared in the first
779 \fBlong long\fR of the transfer.
780 .sp
781 .LP
782 \fBth_define -n foo -i 3 -a intr -c 0 6 -o LOSE 0\fR
783 .sp
784 .LP
785 Causes the next six interrupts for instance 3 of the \fBfoo\fR driver to be
786 lost.
787 .sp
```

```
.LP
\fBth_define -n foo -i 3 -a intr -c 30 1 -o EXTRA 10\fR
.sp
.LP
When the thirty-first subsequent interrupt for instance 3 of the \fBfoo\fR
driver occurs, a further ten interrupts are also generated.
.sp
.LP
\fBth_define -n foo -i 3 -a intr -c 0 1 -o DELAY 1024\fR
.sp
.LP
Causes the next interrupt for instance 3 of the \fBfoo\fR driver to be delayed
by 1024 microseconds.
.SH NOTES
.LP
The policy option in the \fBth_define\fR \fB-p\fR syntax determines how a set
of logged accesses will be converted into the set of error definitions. Each
logged access will be matched against the chosen policies to determine whether
an error definition should be created based on the access.
.sp
.LP
Any number of policy options can be combined to modify the generated error
definitions.
.SS "Bytewise Policies"
.LP
These select particular I/O transfer sizes. Specifying a byte policy will
These select particular I/O transfer sizes. Specifying a byte policy will
exclude other byte policies that have not been chosen. If none of the byte type
policies is selected, all transfer sizes are treated equally. Otherwise, only
those specified transfer sizes will be selected.
.sp
.ne 2
.na
\fB\fBonebyte\fR\fR
.ad
.RS 13n
Create errdefs for one byte accesses (\fBddi_get8()\fR)
.RE

.sp
.ne 2
.na
\fB\fBtwobyte\fR\fR
.ad
.RS 13n
Create errdefs for two byte accesses (\fBddi_get16()\fR)
.RE

.sp
.ne 2
.na
\fB\fBfourbyte\fR\fR
.ad
.RS 13n
Create errdefs for four byte accesses (\fBddi_get32()\fR)
.RE

.sp
.ne 2
.na
\fB\fBeightbyte\fR\fR
.ad
.RS 13n
Create errdefs for eight byte accesses (\fBddi_get64()\fR)
.RE
```



```
.sp
.ne 2
.na
\fB\fBmultibyte\fR\fR
.ad
.RS 13n
Create errdefs for repeated byte accesses (\fBddi_rep_get*()\fR)
.RE

.SS "Frequency of Access Policies"
.LP
The frequency of access to a location is determined according to the access
type, location and transfer size (for example, a two-byte read access to
address A is considered distinct from a four-byte read access to address A).
The algorithm is to count the number of accesses (of a given type and size) to
a given location, and find the locations that were most and least accessed (let
\fImaxa\fR and \fImina\fR be the number of times these locations were accessed,
and \fImean\fR the total number of accesses divided by total number of
locations that were accessed). Then a rare access is a location that was
accessed less than
.sp
.LP
\fI(mean - mina) / 3 + mina\fR
.sp
.LP
times. Similarly for the definition of common accesses:
.sp
.LP
\fImaxa - (maxa - mean) / 3\fR
.sp
.LP
A location whose access patterns lies within these cutoffs is regarded as a
location that is accessed with median frequency.
.sp
.ne 2
.na
\fB\fBrare\fR\fR
.ad
.RS 10n
Create errdefs for locations that are rarely accessed.
.RE

.sp
.ne 2
.na
\fB\fBcommon\fR\fR
.ad
.RS 10n
Create errdefs for locations that are commonly accessed.
.RE

.sp
.ne 2
.na
\fB\fBmedian\fR\fR
.ad
.RS 10n
Create errdefs for locations that are accessed a median frequency.
.RE

.SS "Policies for Minimizing errdefs"
.LP
If a transaction is duplicated, either a single or multiple errdefs will be
written to the test scripts, depending upon the following two policies:
.sp
.ne 2
```

```
 919 .na
 920 \fB\fBmaximal\fR\fR
 921 .ad
 922 .RS 13n
 923 Create multiple errdefs for locations that are repeatedly accessed.
 924 .RE

 926 .sp
 927 .ne 2
 928 .na
 929 \fB\fBunbiased\fR\fR
 930 .ad
 931 .RS 13n
 932 Create a single errdef for locations that are repeatedly accessed.
 933 .RE

 935 .sp
 936 .ne 2
 937 .na
 938 \fB\fBoperators\fR\fR
 939 .ad
 940 .RS 13n
 941 For each location, a default operator and operand is typically applied. For
 942 maximal test coverage, this default may be modified using the \fBoperators\fR
 943 policy so that a separate errdef is created for each of the possible corruption
 944 operators.
 945 .RE

 947 .SH SEE ALSO
 948 .LP
 949 \fBkill\fR(1), \fBth_manage\fR(1M), \fBalarm\fR(2),
 950 \fBddi_check_acc_handle\fR(9F), \fBddi_check_dma_handle\fR(9F)
```

```
**********************************************************
   16624 Sun Sep 16 19:22:53 2018
new/usr/src/man/man1m/zoneadm.1m
9842 man page typos and spelling
**********************************************************
'\" te
.\" Copyright 2015 Nexenta Systems, Inc. All rights reserved.
.\" Copyright (c) 2009 Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH ZONEADM 1M "May 13, 2017"
.SH NAME
zoneadm \- administer zones
.SH SYNOPSIS
.LP
.nf
\fBzoneadm\fR \fB-z\fR \fIzonename\fR [\fB-u\fR \fIuuid-match\fR] \fIsubcommand\
     [\fIsubcommand_options\fR]
.fi

.LP
.nf
\fBzoneadm\fR [\fB-R\fR \fIroot\fR] [\fB-z\fR \fIzonename\fR] [\fB-u\fR \fIuuid-
     [\fIlist_options\fR]
.fi

.LP
.nf
\fBzoneadm\fR [\fB-R\fR \fIroot\fR] \fB-z\fR \fIzonename\fR [\fB-u\fR \fIuuid-ma
.fi

.SH DESCRIPTION
.LP
The \fBzoneadm\fR utility is used to administer system zones. A zone is an
application container that is maintained by the operating system runtime.
.SH SECURITY
.LP
Once a process has been placed in a zone other than zone \fB0\fR, the process
or any of its children cannot change zones.
.SH OPTIONS
.LP
The following options are supported:
.sp
.ne 2
.na
\fB\fB-R\fR \fIroot\fR\fR
.ad
.sp .6
.RS 4n
Specify an alternate root (boot environment). This option can only be used in
conjunction with the "\fBlist\fR" and "\fBmark\fR" subcommands.
.RE

.sp
.ne 2
.na
\fB\fB-u\fR \fIuuid-match\fR\fR
.ad
.sp .6
.RS 4n
Unique identifier for a zone, as assigned by \fBlibuuid\fR(3LIB). If this
option is present and the argument is a non-empty string, then the zone
matching the \fBUUID\fR is selected instead of the one named by the \fB-z\fR
option, if such a zone is present.
.RE
```



```
.sp
.ne 2
.na
\fB\fB-z\fR \fIzonename\fR\fR
.ad
.sp .6
.RS 4n
String identifier for a zone.
.RE

.SH SUBCOMMANDS
.LP
Subcommands which can result in destructive actions or loss of work have a
\fB-F\fR flag to force the action. If input is from a terminal device, the user
is prompted if such a command is given without the \fB-F\fR flag; otherwise, if
such a command is given without the \fB-F\fR flag, the action is disallowed,
with a diagnostic message written to standard error. If a zone installation or
uninstallation is interrupted, the zone is left in the incomplete state. Use
uninstall to reset such a zone back to the configured state.
.sp
.LP
The following subcommands are supported:
.sp
.ne 2
.na
\fB\fBattach\fR [\fB-F\fR] [\fB-n\fR \fIpath\fR] [\fIbrand-specific
options\fR]\fR
.ad
.sp .6
.RS 4n
The \fBattach\fR subcommand takes a zone that has been detached from one system
and attaches the zone onto a new system. Therefore, it is advised (though not
required) that the \fBdetach\fR subcommand should be run before the "attach"
takes place. Once you have the new zone in the configured state, use the
\fBattach\fR subcommand to set up the zone root instead of installing the zone
as a new zone.
.sp
The \fB-F\fR option can be used to force the zone into the "installed" state
with no validation. This option should be used with care since it can leave the
zone in an unsupportable state if it was moved from a source system to a target
system that is unable to properly host the zone. The \fB-n\fR option can be
used to run the \fBattach\fR subcommand, without executing the command. It uses
the output of the "\fBdetach\fR \fB-n\fR" subcommand as input and is useful to
identify any conflicting issues, such as the network device being incompatible,
and can also determine whether the host is capable of supporting the zone. The
path can be "\fB-\fR", to read the input from standard input.
.sp
The zone’s brand may include additional options that govern how the zone will
be attached. See \fBbrands\fR(5) for specific brand information.
.sp
The zone being attached must first be configured using the \fBzonecfg\fR (see
\fBzonecfg\fR(1M)) command. This does not apply when running "\fBattach\fR
\fB-n\fR".
.sp
Use the following command to attach a zone:
.sp
.in +2
.nf
# \fBzoneadm -z my-zone attach\fR
.fi
.in -2
.sp

.RE
```

```
.sp
.ne 2
.na
\fB\fBboot\fR [\fB--\fR \fIboot_options\fR]\fR
.ad
.sp .6
.RS 4n
Boot (or activate) the specified zones.
.sp
The following \fIboot_options\fR are supported:
.sp
.ne 2
.na
\fB\fB-i\fR \fIaltinit\fR\fR
.ad
.sp .6
.RS 4n
Select an alternative executable to be the primordial Process. \fIaltinit\fR is
a valid path to an executable. The default primordial process is
\fBinit\fR(1M).
.RE

.sp
.ne 2
.na
\fB\fB-m\fR \fIsmf_options\fR\fR
.ad
.sp .6
.RS 4n
The \fIsmf_options\fR include two categories of options to control booting
behavior of the service management facility: recovery options and messages
options.
.sp
Message options determine the type and amount of messages that \fBsmf\fR(5)
displays during boot. Service options determine the services which are used to
boot the system. See \fBkernel\fR(1M) for a listing of the \fB-m\fR suboptions.
.RE

.sp
.ne 2
.na
\fB\fB-s\fR\fR
.ad
.sp .6
.RS 4n
Boots only to milestone \fBsvc:/milestone/single-user:default\fR. This
milestone is equivalent to init level \fBs\fR. See \fBsvc.startd\fR(1M) and
\fBinit\fR(1M).
.RE

.RE

.sp
.ne 2
.na
\fB\fBclone\fR [\fB-m\fR \fIcopy\fR] [\fB-s\fR \fIzfs_snapshot\fR]
\fIsource_zone\fR\fR
.ad
.sp .6
.RS 4n
Install a zone by copying an existing installed zone. This subcommand is an
alternative way to install the zone.
.sp
.ne 2
.na
\fB\fB-m\fR \fIcopy\fR\fR
```



```
.ad
.sp .6
.RS 4n
Force the clone to be a copy, even if a "\fBZFS\fR clone" is possible.
.RE

.sp
.ne 2
.na
\fB\fB-s\fR \fIzfs_snapshot\fR\fR
.ad
.sp .6
.RS 4n
Specify the name of a \fBZFS\fR snapshot to use as the source of the clone. The
\fIsnapshot\fR must be a \fIsnapshot\fR of the source zone taken from a
previous "\fBzoneadm\fR clone" installation.
.RE

The source zone must be halted before this subcommand can be used.
.RE

.sp
.ne 2
.na
\fB\fBdetach\fR [\fB-n\fR]\fR
.ad
.sp .6
.RS 4n
Detach the specified zone. Detaching a zone is the first step in moving a zone
from one system to another. The full procedure to migrate a zone is that the
zone is detached, the \fIzonepath\fR directory is moved to the new host, and
then the zone is attached on the new host. Once the zone is detached, it is
left in the configured state. If you try to install or clone to a configured
zone that has been detached, you will receive an error message and the
\fBinstall\fR or \fBclone\fR subcommand will not be allowed to proceed. The
\fB-n\fR option can be used to run the \fBdetach\fR subcommand, without
executing the command. This generates the information needed for running the
"\fBattach\fR \fB-n\fR" subcommand, which is useful to identify any conflicting
issues, such as the network device being incompatible or if the host is capable
of supporting the zone. The information is sent to standard output and can be
saved to a file or piped to the "\fBattach\fR \fB-n\fR" subcommand.
.sp
Use the following command to detach a zone:
.sp
.in +2
.nf
# zoneadm -z my-zone detach
.fi
.in -2
.sp

The source zone must be halted before this subcommand can be used.
.RE

.sp
.ne 2
.na
\fB\fBhalt\fR\fR
.ad
.sp .6
.RS 4n
Halt the specified zones. \fBhalt\fR bypasses running the shutdown scripts
inside the zone. It also removes run time resources of the zone.
.RE

.sp
```

```
.ne 2
.na
\fB\fBhelp\fR [\fIsubcommand\fR]\fR
.ad
.sp .6
.RS 4n
Display general help. If you specify \fIsubcommand\fR, displays help on
\fIsubcommand\fR.
.RE

.sp
.ne 2
.na
\fB\fBinstall\fR [\fB-x\fR \fInodataset\fR] [\fIbrand-specific options\fR]\fR
.ad
.sp .6
.RS 4n
Install the specified zone on the system. This subcommand automatically
attempts to verify first, most verification errors are fatal. See the
\fBverify\fR subcommand.
.sp
.ne 2
.na
\fB\fB-x\fR \fInodataset\fR\fR
.ad
.sp .6
.RS 4n
Do not create a \fBZFS\fR file system.
.RE

The zone’s brand may include additional options that govern how the software
will be installed in the zone. See \fBbrands\fR(5) for specific brand
information.
.RE

.sp
.ne 2
.na
\fB\fBlist\fR [\fIlist_options\fR]\fR
.ad
.sp .6
.RS 4n
Display the name of the current zones, or the specified zone if indicated.
.sp
By default, all running zones are listed. If you use this subcommand with the
\fBzoneadm\fR \fB-z\fR \fIzonename\fR option, it lists only the specified zone,
regardless of its state. In this case, the \fB-i\fR and \fB-c\fR options are
disallowed.
.sp
If neither the \fB-i\fR or \fB-c\fR options are given, all running zones are
listed.
.sp
The following \fIlist_options\fR are supported:
.sp
.ne 2
.na
\fB\fB-c\fR\fR
.ad
.sp .6
.RS 4n
Display all configured zones. This option overrides the \fB-i\fR option.
Display all configured zones. This option overides the \fB-i\fR option.
.RE

.sp
.ne 2
```



```
.na
\fB\fB-i\fR\fR
.ad
.sp .6
.RS 4n
Expand the display to all installed zones.
.RE

.sp
.ne 2
.na
\fB\fB-p\fR\fR
.ad
.sp .6
.RS 4n
Request machine parsable output. The output format is a list of lines, one per
zone, with colon- delimited fields. These fields are:
.sp
.in +2
.nf
zoneid:zonename:state:zonepath:uuid:brand:ip-type
.fi
.in -2
.sp

If the \fBzonepath\fR contains embedded colons, they can be escaped by a
backslash ("\:"), which is parsable by using the shell \fBread\fR(1) function
with the environmental variable \fBIFS\fR. The \fIuuid\fR value is assigned by
\fBlibuuid\fR(3LIB) when the zone is installed, and is useful for identifying
the same zone when present (or renamed) on alternate boot environments. Any
software that parses the output of the "\fBzoneadm list -p\fR" command must be
able to handle any fields that may be added in the future.
.sp
The \fB-v\fR and \fB-p\fR options are mutually exclusive. If neither \fB-v\fR
nor \fB-p\fR is used, just the zone name is listed.
.RE

.sp
.ne 2
.na
\fB\fB-v\fR\fR
.ad
.sp .6
.RS 4n
Display verbose information, including zone name, id, current state, root
directory, brand type, ip-type, and options.
.sp
The \fB-v\fR and \fB-p\fR options are mutually exclusive. If neither \fB-v\fR
nor \fB-p\fR is used, just the zone name is listed.
.RE

.RE

.sp
.ne 2
.na
\fB\fBmark incomplete\fR\fR
.ad
.sp .6
.RS 4n
Change the state of an installed zone to "incomplete." This command may be
useful in cases where administrative changes on the system have rendered a zone
unusable or inconsistent. This change cannot be undone (except by uninstalling
the zone).
.RE
```

```
391 .sp
392 .ne 2
393 .na
394 \fB\fBmove\fR \fInew_zonepath\fR\fR
395 .ad
396 .sp .6
397 .RS 4n
398 Move the \fIzonepath\fR to \fInew_zonepath\fR. The zone must be halted before
399 this subcommand can be used. The \fInew_zonepath\fR must be a local file system
400 and normal restrictions for \fIzonepath\fR apply.
401 .RE

403 .sp
404 .ne 2
405 .na
406 \fB\fBready\fR\fR
407 .ad
408 .sp .6
409 .RS 4n
410 Prepares a zone for running applications but does not start any user processes
411 in the zone.
412 .RE

414 .sp
415 .ne 2
416 .na
417 \fB\fBreboot\fR\ [\fB--\fR \fIboot_options\fR]]\fR
418 .ad
419 .sp .6
420 .RS 4n
421 Restart the zones. This is equivalent to a \fBhalt\fR \fBboot\fR sequence. This
422 subcommand fails if the specified zones are not active. See \fIboot\fR subcomman
423 for the boot options.
424 .RE

426 .sp
427 .ne 2
428 .na
429 \fB\fBshutdown\fR [\fB-r\fR [\fB--\fR \fIboot_options\fR]]\fR
430 .ad
431 .sp .6
432 .RS 4n
433 Gracefully shutdown the specified zone. This subcommand waits for all zone
434 processes to finish; the default timeout is SCF_PROPERTY_TIMEOUT value from
435 the SMF service system/zones. If the \fB-r\fR option is specified, reboot the
436 zone. See \fIboot\fR subcommand for the boot options.
437 .RE

439 .sp
440 .ne 2
441 .na
442 \fB\fBuninstall [\fR\fB-F\fR\fB]\fR\fR
443 .ad
444 .sp .6
445 .RS 4n
446 Uninstall the specified zone from the system. Use this subcommand with caution.
447 It removes all of the files under the \fIzonepath\fR of the zone in question.
448 You can use the \fB-F\fR flag to force the action.
449 .RE

451 .sp
452 .ne 2
453 .na
454 \fB\fBverify\fR\fR
455 .ad
456 .sp .6
```



```
457 .RS 4n
458 Check to make sure the configuration of the specified zone can safely be
459 installed on the machine. Following is a break-down of the checks by
460 \fBresource/property\fR type:
461 .sp
462 .ne 2
463 .na
464 \fB\fBzonepath\fR\fR
465 .ad
466 .sp .6
467 .RS 4n
468 \fBzonepath\fR and its parent directory exist and are owned by root with
469 appropriate modes . The appropriate modes are that \fBzonepath\fR is \fB700\fR,
470 its parent is not \fBgroup\fR or \fBworld-writable\fR and so forth.
471 \fBzonepath\fR is not over an NFS mount. A sub-directory of the \fBzonepath\fR
472 named "root" does not exist.
473 .sp
474 If \fBzonepath\fR does not exist, the \fBverify\fR does not fail, but merely
475 warns that a subsequent install will attempt to create it with proper
476 permissions. A \fBverify\fR subsequent to that might fail should anything go
477 wrong.
478 .sp
479 \fBzonepath\fR cannot be a symbolic link.
480 .RE

482 .sp
483 .ne 2
484 .na
485 \fB\fBfs\fR\fR
486 .ad
487 .sp .6
488 .RS 4n
489 Any \fBfs\fR resources have their \fItype\fR value checked. An error is
490 reported if the value is one of \fBproc\fR, \fBmntfs\fR, \fBautofs\fR,
491 or \fBnfs\fR or the filesystem does not have an associated mount
492 binary at \fB/usr/lib/fs/\fI<fstype>\fR/mount\fR.
493 .sp
494 It is an error for the \fIdirectory\fR to be a relative path.
495 .sp
496 It is an error for the path specified by \fBraw\fR to be a relative path or if
497 there is no \fBfsck\fR binary for a given filesystem type at
498 \fB/usr/lib/fs/\fI<fstype>\fR/fsck\fR. It is also an error if a corresponding
499 \fBfsck\fR binary exists but a \fBraw\fR path is not specified.
500 .RE

502 .sp
503 .ne 2
504 .na
505 \fB\fBnet\fR\fR
506 .ad
507 .sp .6
508 .RS 4n
509 All physical network interfaces exist. All network address resources are one
510 of:
511 .RS +4
512 .TP
513 .ie t \(bu
514 .el o
515 a valid IPv4 address, optionally followed by "\fB/\fR" and a prefix length;
516 .RE
517 .RS +4
518 .TP
519 .ie t \(bu
520 .el o
521 a valid IPv6 address, which must be followed by "\fB/\fR" and a prefix length;
522 .RE
```

```
.RS +4
.TP
.ie t \(bu
.el o
a host name which resolves to an IPv4 address.
.RE
Note that hostnames that resolve to IPv6 addresses are not supported.
.sp
The physical interface name is the network interface name.
.sp
A zone can be configured to be either exclusive-IP or shared-IP. For a
shared-IP zone, both the physical and address properties must be set. For an
exclusive-IP zone, the physical property must be set and the address property
cannot be set.
.RE

.sp
.ne 2
.na
\fB\fBrctl\fR\fR
.ad
.sp .6
.RS 4n
It also verifies that any defined resource control values are valid on the
current machine. This means that the privilege level is \fBprivileged\fR, the
limit is lower than the currently defined system value, and that the defined
action agrees with the actions that are valid for the given resource control.
.RE

.RE

.SH EXAMPLES
.LP
\fBExample 1 \fRUsing the \fB-m\fR Option
.sp
.LP
The following command illustrates the use of the \fB-m\fR option.

.sp
.in +2
.nf
# \fBzoneadm boot -- -m verbose\fR
.fi
.in -2
.sp

.LP
\fBExample 2 \fRUsing the \fB-i\fR Option
.sp
.LP
The following command illustrates the use of the \fB-i\fR option.

.sp
.in +2
.nf
# \fBzoneadm boot -- -i /sbin/init\fR
.fi
.in -2
.sp

.LP
\fBExample 3 \fRUsing the \fB-s\fR Option
.sp
.LP
The following command illustrates the use of the \fB-s\fR option.
```



```
.sp
.in +2
.nf
# \fBzoneadm boot -- -s\fR
.fi
.in -2
.sp

.SH EXIT STATUS
.LP
The following exit values are returned:
.sp
.ne 2
.na
\fB\fB0\fR\fR
.ad
.sp .6
.RS 4n
Successful completion.
.RE

.sp
.ne 2
.na
\fB\fB1\fR\fR
.ad
.sp .6
.RS 4n
An error occurred.
.RE

.sp
.ne 2
.na
\fB\fB2\fR\fR
.ad
.sp .6
.RS 4n
Invalid usage.
.RE

.SH ATTRIBUTES
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
Interface Stability     Committed
.TE

.SH SEE ALSO
.LP
\fBread\fR(1), \fBsvcs\fR(1), \fBzlogin\fR(1), \fBzonename\fR(1),
\fBinit\fR(1M), \fBkernel\fR(1M), \fBsvcadm\fR(1M), \fBsvc.startd\fR(1M),
\fBsvc.startd\fR(1M), \fBzonecfg\fR(1M), \fBlibuuid\fR(3LIB),
\fBattributes\fR(5), \fBbrands\fR(5), \fBnative\fR(5), \fBsmf\fR(5),
\fBzones\fR(5)
.SH NOTES
.LP
The \fBzones\fR(5) service is managed by the service management facility,
```

```
655 \fBsmf\fR(5), under the service identifier:
656 .sp
657 .in +2
658 .nf
659 svc:/system/zones:default
660 .fi
661 .in -2
662 .sp

664 .sp
665 .LP
666 Administrative actions on this service, such as enabling, disabling, or
667 requesting restart, can be performed using \fBsvcadm\fR(1M). The service's
668 status can be queried using the \fBsvcs\fR(1) command.
```

```
**********************************************************
   13931 Sun Sep 16 19:22:53 2018
new/usr/src/man/man2/chmod.2
9842 man page typos and spelling
**********************************************************
   1 .\"
   2 .\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for
   3 .\" permission to reproduce portions of its copyrighted documentation.
   4 .\" Original documentation from The Open Group can be obtained online at
   5 .\" http://www.opengroup.org/bookstore/.
   6 .\"
   7 .\" The Institute of Electrical and Electronics Engineers and The Open
   8 .\" Group, have given us permission to reprint portions of their
   9 .\" documentation.
  10 .\"
  11 .\" In the following statement, the phrase ‘‘this text’’ refers to portions
  12 .\" of the system documentation.
  13 .\"
  14 .\" Portions of this text are reprinted and reproduced in electronic form
  15 .\" in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition,
  16 .\" Standard for Information Technology -- Portable Operating System
  17 .\" Interface (POSIX), The Open Group Base Specifications Issue 6,
  18 .\" Copyright (C) 2001-2004 by the Institute of Electrical and Electronics
  19 .\" Engineers, Inc and The Open Group.  In the event of any discrepancy
  20 .\" between these versions and the original IEEE and The Open Group
  21 .\" Standard, the original IEEE and The Open Group Standard is the referee
  22 .\" document.  The original Standard can be obtained online at
  23 .\" http://www.opengroup.org/unix/online.html.
  24 .\"
  25 .\" This notice shall appear on any product containing this material.
  26 .\"
  27 .\" The contents of this file are subject to the terms of the
  28 .\" Common Development and Distribution License (the "License").
  29 .\" You may not use this file except in compliance with the License.
  30 .\"
  31 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
  32 .\" or http://www.opensolaris.org/os/licensing.
  33 .\" See the License for the specific language governing permissions
  34 .\" and limitations under the License.
  35 .\"
  36 .\" When distributing Covered Code, include this CDDL HEADER in each
  37 .\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  38 .\" If applicable, add the following below this CDDL HEADER, with the
  39 .\" fields enclosed by brackets "[]" replaced with your own identifying
  40 .\" information: Portions Copyright [yyyy] [name of copyright owner]
  41 .\"
  42 .\"
  43 .\" Copyright 1989 AT&T.
  44 .\" Copyright (c) 2001, The IEEE and The Open Group.  All Rights Reserved.
  45 .\" Copyright (c) 2005, Sun Microsystems, Inc.  All Rights Reserved.
  46 .\" Copyright (c) 2014, Joyent, Inc.
  47 .\"
  48 .TH CHMOD 2 "Dec 22, 2014"
  49 .SH NAME
  50 chmod, fchmod, fchmodat \- change access permission mode of file
  51 .SH SYNOPSIS
  52 .LP
  53 .nf
  54 #include <sys/types.h>
  55 #include <sys/stat.h>

  57 \fBint\fR \fBchmod\fR(\fBconst char *\fR\fIpath\fR, \fBmode_t\fR \fImode\fR);
  58 .fi

  60 .LP
  61 .nf
```



```
  62 \fBint\fR \fBfchmod\fR(\fBint\fR \fIfildes\fR, \fBmode_t\fR \fImode\fR);
  63 .fi

  65 .LP
  66 .nf
  67 \fBint\fR \fBfchmodat\fR(\fBint\fR \fIfildes\fR, \fBconst char *\fR\fIpath\fR, \
  68 .fi

  70 .SH DESCRIPTION
  71 .LP
  72 The \fBchmod()\fR, \fBfchmod()\fR, and \fBfchmodat()\fR functions set the access
  73 permission portion of the mode of the file whose name is given by \fIpath\fR or
  74 referenced by the open file descriptor \fIfildes\fR to the bit pattern contained
  75 in \fImode\fR. Access permission bits are interpreted as follows:
  76 .sp

  78 .sp
  79 .TS
  80 l l l
  81 l l l .
  82 \fBS_ISUID\fR   04000   Set user ID on execution.
  83 \fBS_ISGID\fR   020#0   T{
  84 Set group ID on execution if # is \fB7\fR, \fB5\fR, \fB3\fR, or \fB1\fR. Enable
  85 T}
  86 \fBS_ISVTX\fR   01000   Sticky bit.
  87 \fBS_IRWXU\fR   00700   Read, write, execute by owner.
  88 \fBS_IRUSR\fR   00400   Read by owner.
  89 \fBS_IWUSR\fR   00200   Write by owner.
  90 \fBS_IXUSR\fR   00100   T{
  91 Execute (search if a directory) by owner.
  92 T}
  93 \fBS_IRWXG\fR   00070   Read, write, execute by group.
  94 \fBS_IRGRP\fR   00040   Read by group.
  95 \fBS_IWGRP\fR   00020   Write by group.
  96 \fBS_IXGRP\fR   00010   Execute by group.
  97 \fBS_IRWXO\fR   00007   Read, write, execute (search) by others.
  98 \fBS_IROTH\fR   00004   Read by others.
  99 \fBS_IWOTH\fR   00002   Write by others.
 100 \fBS_IXOTH\fR   00001   Execute by others.
 101 .TE

 103 .sp
 104 .LP
 105 Modes are constructed by the bitwise OR operation of the access permission
 106 bits.
 107 .sp
 108 .LP
 109 The effective user ID of the process must match the owner of the file or the
 110 process must have the appropriate privilege to change the mode of a file.
 111 .sp
 112 .LP
 113 If the process is not a privileged process and the file is not a directory,
 114 mode bit 01000 (save text image on execution) is cleared.
 115 .sp
 116 .LP
 117 If neither the process is privileged nor the file’s group is a member of the
 118 process’s  supplementary group list, and the effective group ID of the process
 119 does not match the group ID of the file, mode bit 02000 (set group ID on
 120 execution) is cleared.
 121 .sp
 122 .LP
 123 If a directory is writable and has \fBS_ISVTX\fR (the sticky bit) set, files
 124 within that directory can be removed or renamed only if one or more of the
 125 following is true (see \fBunlink\fR(2) and \fBrename\fR(2)):
 126 .RS +4
 127 .TP
```

```
.ie t \(bu
.el o
the user owns the file
.RE
.RS +4
.TP
.ie t \(bu
.el o
the user owns the directory
.RE
.RS +4
.TP
.ie t \(bu
.el o
the file is writable by the user
.RE
.RS +4
.TP
.ie t \(bu
.el o
the user is a privileged user
.RE
.sp
.LP
If a regular file is not executable and has \fBS_ISVTX\fR set, the file is
assumed to be a swap file. In this case, the system's page cache will not be
used to hold the file's data. If the \fBS_ISVTX\fR bit is set on any other
file, the results are unspecified.
.sp
.LP
If a directory has the set group ID bit set, a given file created within that
directory will have  the same group ID as the directory.  Otherwise, the newly
created file's group ID will be set to the effective group ID of the creating
process.
.sp
.LP
If the mode bit 02000 (set group ID on execution) is set and the mode bit 00010
(execute or search by group) is not set, mandatory file/record locking will
exist on a regular file, possibly affecting future calls to \fBopen\fR(2),
\fBcreat\fR(2), \fBread\fR(2), and \fBwrite\fR(2) on this file.
.sp
.LP
If \fIfildes\fR references a shared memory object, \fBfchmod()\fR need only
affect the \fBS_IRUSR\fR, \fBS_IRGRP\fR, \fBS_IROTH\fR, \fBS_IWUSR\fR,
\fBS_IWGRP\fR, \fBS_IWOTH\fR, \fBS_IXUSR\fR, \fBS_IXGRP\fR, and \fBS_IXOTH\fR
file permission bits.
.sp
.LP
If \fIfildes\fR refers to a socket, \fBfchmod()\fR does not fail but no action
is taken.
.sp
.LP
If \fIfildes\fR refers to a stream that is attached to an object in the file
system name space with \fBfattach\fR(3C), the \fBfchmod()\fR call performs no
action and returns successfully.
.sp
.LP
The \fBfchmodat()\fR function behaves similarly to \fBchmod()\fR, except when
\fIpath\fR is a relative path, it is resolved relative to the directory
specified by \fIfiledes\fR. If \fBfiledes\fR has the value \fBAT_FDCWD\fR, then
\fBpath\fR will be resolved relative to the current working directory. The
argument \fIflag\fR should be zero, but may include the value
\fBAT_SYMLINK_NOFOLLOW\fR, which indicates that if \fIpath\fR refers to a
symbolic link, then permissions should be changed on the symbolic link itself.
However, changing permissions of symbolic links is not supported on illumos, and
will result in an error.
```



```
.sp
.LP
Upon successful completion, \fBchmod()\fR, \fBfchmod()\fR, \fBfchmodat()\fR mark
for update the \fBst_ctime\fR field of the file.
.SH RETURN VALUES
.LP
Upon successful completion, \fB0\fR is returned. Otherwise, \fB\(mi1\fR is
returned, the file mode is unchanged, and \fBerrno\fR is set to indicate the
error.
.SH ERRORS
.LP
The \fBchmod()\fR, \fBfchmod()\fR, and \fBfchmodat()\fR functions will fail if:
.sp
.ne 2
.na
\fB\fBEIO\fR\fR
.ad
.RS 9n
An I/O error occurred while reading from or writing to the file system.
.RE

.sp
.ne 2
.na
\fB\fBEPERM\fR\fR
.ad
.RS 9n
The effective user ID does not match the owner of the file and the process does
not have appropriate privilege.
.sp
The {\fBPRIV_FILE_OWNER\fR} privilege overrides constraints on ownership when
changing permissions on a file.
.sp
The {\fBPRIV_FILE_SETID\fR} privilege overrides constraints on ownership when
adding the setuid or setgid bits to an executable file or a directory.  When
adding the setuid bit to a root owned executable, additional restrictions
apply. See \fBprivileges\fR(5).
.RE

.sp
.LP
The \fBchmod()\fR and \fBfchmodat()\fR functions will fail if:
.sp
.ne 2
.na
\fB\fBEACCES\fR\fR
.ad
.RS 16n
Search permission is denied on a component of the path prefix of \fIpath\fR and
for \fBfchmodat()\fR, \fBfiledes\fR was not opened with \fBO_SEARCH\fR requested
for \fBfchmodat()\fR, \fBfiledes\fR was not opened with \fBO_SEARCH\fR requsted.
The privilege {\fBFILE_DAC_SEARCH\fR} overrides file permissions restrictions
in that case.
.RE

.sp
.ne 2
.na
\fB\fBEFAULT\fR\fR
.ad
.RS 16n
The \fIpath\fR argument points to an illegal address.
.RE

.sp
.ne 2
```

```
.na
\fB\fBELOOP\fR\fR
.ad
.RS 16n
A loop exists in symbolic links encountered during the resolution of the
\fIpath\fR argument.
.RE

.sp
.ne 2
.na
\fB\fBENAMETOOLONG\fR\fR
.ad
.RS 16n
The length of the \fIpath\fR argument exceeds \fBPATH_MAX\fR, or the length of
a \fIpath\fR component exceeds \fBNAME_MAX\fR while \fB_POSIX_NO_TRUNC\fR is in
effect.
.RE

.sp
.ne 2
.na
\fB\fBENOENT\fR\fR
.ad
.RS 16n
Either a component of the path prefix or the file referred to by \fIpath\fR
does not exist or is a null pathname.
.RE

.sp
.ne 2
.na
\fB\fBENOLINK\fR\fR
.ad
.RS 16n
The \fIfildes\fR argument points to a remote machine and the link to that
machine is no longer active.
.RE

.sp
.ne 2
.na
\fB\fBENOTDIR\fR\fR
.ad
.RS 16n
A component of the prefix of \fIpath\fR is not a directory.
.RE

.sp
.ne 2
.na
\fB\fBEROFS\fR\fR
.ad
.RS 16n
The file referred to by \fIpath\fR resides on a read-only file system.
.RE

.sp
.LP
The \fBfchmod()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBEBADF\fR\fR
.ad
.RS 11n
```



```
The \fIfildes\fR argument is not an open file descriptor
.RE

.sp
.ne 2
.na
\fB\fBENOLINK\fR\fR
.ad
.RS 11n
The \fIpath\fR argument points to a remote machine and the link to that machine
is no longer active.
.RE

.sp
.ne 2
.na
\fB\fBEROFS\fR\fR
.ad
.RS 11n
The file referred to by \fIfildes\fR resides on a read-only file system.
.RE

.sp
.LP
The \fBchmod()\fR and \fBfchmod()\fR functions may fail if:
.sp
.ne 2
.na
\fB\fBEINTR\fR\fR
.ad
.RS 10n
A signal was caught during execution of the function.
.RE

.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 10n
The value of the \fImode\fR argument is invalid.
.RE

.sp
.LP
The \fBfchmodat()\fR will fail if:
.sp
.ne 2
.na
.B EBADF
.ad
.RS 16n
The argument \fIpath\fR is a relative path and \fIfiledes\fR is not an open file
descriptor or the value \fBAT_FDCWD\fR.
.RE

.sp
.ne 2
.na
.B EINVAL
.ad
.RS 16n
The argument \fIflags\fR has a non-zero value other than
\fBAT_SYMLINK_NOFOLLOW\fR.
.RE
```

```
391 .sp
392 .ne 2
393 .na
394 .B ENOTDIR
395 .ad
396 .RS 16n
397 The argument \fIpath\fR is a relative path and \fIfiledes\fR is a valid file
398 descriptor which does not refer to a file.
399 .RE

401 .sp
402 .ne 2
403 .na
404 .B EOPNOTSUPP
405 .ad
406 .RS 16n
407 The \fBAT_SYMLINK_NOFOLLOW\fR bit is set in the \fIflags\fR argument.
408 .RE

410 .sp
411 .LP
412 The \fBchmod()\fR and \fBfchmodat()\fR functions may fail if:
413 .sp
414 .ne 2
415 .na
416 \fB\fBELOOP\fR\fR
417 .ad
418 .RS 16n
419 More than {\fBSYMLOOP_MAX\fR} symbolic links were encountered during the
420 resolution of the \fIpath\fR argument.
421 .RE

423 .sp
424 .ne 2
425 .na
426 \fB\fBENAMETOOLONG\fR\fR
427 .ad
428 .RS 16n
429 As a result of encountering a symbolic link in resolution of the \fIpath\fR
430 argument, the length of the substituted pathname strings exceeds
431 {\fBPATH_MAX\fR}.
432 .RE

434 .sp
435 .LP
436 The \fBfchmod()\fR function may fail if:
437 .sp
438 .ne 2
439 .na
440 \fB\fBEINVAL\fR\fR
441 .ad
442 .RS 10n
443 The \fIfildes\fR argument refers to a pipe and the system disallows execution
444 of this function on a pipe.
445 .RE

447 .SH EXAMPLES
448 .LP
449 \fBExample 1 \fRSet Read Permissions for User, Group, and Others
450 .sp
451 .LP
452 The following example sets read permissions for the owner, group, and others.

454 .sp
455 .in +2
456 .nf
```



```
457 #include <sys/stat.h>
458 const char *path;
459 \&...
460 chmod(path, S_IRUSR|S_IRGRP|S_IROTH);
461 .fi
462 .in -2

464 .LP
465 \fBExample 2 \fRSet Read, Write, and Execute Permissions for the Owner Only
466 .sp
467 .LP
468 The following example sets read, write, and execute permissions for the owner,
469 and no permissions for group and others.

471 .sp
472 .in +2
473 .nf
474 #include <sys/stat.h>
475 const char *path;
476 \&...
477 chmod(path, S_IRWXU);
478 .fi
479 .in -2

481 .LP
482 \fBExample 3 \fRSet Different Permissions for Owner, Group, and Other
483 .sp
484 .LP
485 The following example sets owner permissions for CHANGEFILE to read, write, and
486 execute, group permissions to read and execute, and other permissions to read.

488 .sp
489 .in +2
490 .nf
491 #include <sys/stat.h>
492 #define CHANGEFILE "/etc/myfile"
493 \&...
494 chmod(CHANGEFILE, S_IRWXU|S_IRGRP|S_IXGRP|S_IROTH);
495 .fi
496 .in -2

498 .LP
499 \fBExample 4 \fRSet and Checking File Permissions
500 .sp
501 .LP
502 The following example sets the file permission bits for a file named
503 \fB/home/cnd/mod1\fR, then calls the \fBstat\fR(2) function to verify the
504 permissions.

506 .sp
507 .in +2
508 .nf
509 #include <sys/types.h>
510 #include <sys/stat.h>
511 int status;
512 struct stat buffer
513 \&...
514 chmod("home/cnd/mod1", S_IRWXU|S_IRWXG|S_IROTH|S_IWOTH);
515 status = stat("home/cnd/mod1", &buffer;);
516 .fi
517 .in -2

519 .SH USAGE
520 .LP
521 If \fBchmod()\fR or \fBfchmod()\fR is used to change the file group owner
522 permissions on a file with non-trivial ACL entries, only the ACL mask is set to
```

```
523 the new permissions and the group owner permission bits in the file’s mode
524 field (defined in \fBmknod\fR(2)) are unchanged.  A non-trivial ACL entry is
525 one whose meaning cannot be represented in the file’s mode field alone. The new
526 ACL mask permissions  might change the effective permissions for additional
527 users and groups that have ACL entries on the file.
528 .SH ATTRIBUTES
529 .LP
530 See \fBattributes\fR(5) for descriptions of the following attributes:
531 .sp

533 .sp
534 .TS
535 box;
536 c | c
537 l | l .
538 ATTRIBUTE TYPE  ATTRIBUTE VALUE
539 _
540 Interface Stability     Standard
541 _
542 MT-Level        Async-Signal-Safe
543 .TE

545 .SH SEE ALSO
546 .LP
547 \fBchmod\fR(1), \fBchown\fR(2), \fBcreat\fR(2), \fBfcntl\fR(2), \fBmknod\fR(2),
548 \fBopen\fR(2), \fBread\fR(2), \fBrename\fR(2), \fBstat\fR(2), \fBwrite\fR(2),
549 \fBfattach\fR(3C), \fBmkfifo\fR(3C), \fBstat.h\fR(3HEAD), \fBattributes\fR(5),
550 \fBprivileges\fR(5), \fBstandards\fR(5)
551 .sp
552 .LP
553 \fIProgramming Interfaces Guide\fR
```

```
**********************************************************
   21864 Sun Sep 16 19:22:54 2018
new/usr/src/man/man2/exec.2
9842 man page typos and spelling
**********************************************************
.\"
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for
.\" permission to reproduce portions of its copyrighted documentation.
.\" Original documentation from The Open Group can be obtained online at
.\" http://www.opengroup.org/bookstore/.
.\"
.\" The Institute of Electrical and Electronics Engineers and The Open
.\" Group, have given us permission to reprint portions of their
.\" documentation.
.\"
.\" In the following statement, the phrase ‘‘this text’’ refers to portions
.\" of the system documentation.
.\"
.\" Portions of this text are reprinted and reproduced in electronic form
.\" in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition,
.\" Standard for Information Technology -- Portable Operating System
.\" Interface (POSIX), The Open Group Base Specifications Issue 6,
.\" Copyright (C) 2001-2004 by the Institute of Electrical and Electronics
.\" Engineers, Inc and The Open Group.  In the event of any discrepancy
.\" between these versions and the original IEEE and The Open Group
.\" Standard, the original IEEE and The Open Group Standard is the referee
.\" document.  The original Standard can be obtained online at
.\" http://www.opengroup.org/unix/online.html.
.\"
.\" This notice shall appear on any product containing this material.
.\"
.\" The contents of this file are subject to the terms of the
.\" Common Development and Distribution License (the "License").
.\" You may not use this file except in compliance with the License.
.\"
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
.\" or http://www.opensolaris.org/os/licensing.
.\" See the License for the specific language governing permissions
.\" and limitations under the License.
.\"
.\" When distributing Covered Code, include this CDDL HEADER in each
.\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
.\" If applicable, add the following below this CDDL HEADER, with the
.\" fields enclosed by brackets "[]" replaced with your own identifying
.\" information: Portions Copyright [yyyy] [name of copyright owner]
.\"
.\"
.\" Copyright 1989 AT&T.
.\" Portions Copyright (c) 1992, X/Open Company Limited.  All Rights Reserved.
.\" Copyright (c) 2008, Sun Microsystems, Inc.  All Rights Reserved.
.\" Copyright 2015, Joyent, Inc.
.\"
.TH EXEC 2 "Oct 27, 2015"
.SH NAME
exec, execl, execle, execlp, execv, execve, execvp \- execute a file
.SH SYNOPSIS
.LP
.nf
#include <unistd.h>

\fBint\fR \fBexecl\fR(\fBconst char *\fR\fIpath\fR, \fBconst char *\fR\fIarg0\fR
     /* const char *\fR\fIargn\fR, \fB(char *)0 */);\fR
.fi

.LP
.nf
\fBint\fR \fBexecle\fR(\fBconst char *\fR\fIpath\fR, \fBconst char *\fR\fIarg0\f
     /* const char *\fR\fIargn\fR, \fB(char *)0\fR,\fBchar *const\fR \fIenvp\fR[
.fi

.LP
.nf
\fBint\fR \fBexecve\fR(\fBconst char *\fR\fIpath\fR, \fBchar *const\fR \fIargv[]
     \fBchar *const\fR \fIenvp[]\fR);
.fi

.LP
.nf
\fBint\fR \fBexeclp\fR(\fBconst char *\fR\fIfile\fR, \fBconst char *\fR\fIarg0\f
     /* const char *\fR\fIargn\fR, \fB(char *)0 */);\fR
.fi

.LP
.nf
\fBint\fR \fBexecvp\fR(\fBconst char *\fR\fIfile\fR, \fBchar *const\fR \fIargv[]
.fi

.SH DESCRIPTION
.LP
Each of the functions in the \fBexec\fR family replaces the current process
image with a new process image. The new image is constructed from a regular,
executable file called the \fBnew process image file\fR. This file is either an
executable object file or a file of data for an interpreter. There is no return
from a successful call to one of these functions because the calling process
image is overlaid by the new process image.
.sp
.LP
An interpreter file begins with a line of the form
.sp
.in +2
.nf
#! pathname [\fIarg\fR]
.fi
.in -2

.sp
.LP
where \fIpathname\fR is the path of the interpreter, and \fIarg\fR is an
optional argument. When an interpreter file is executed, the system invokes the
specified interpreter. The pathname specified in the interpreter file is passed
as \fIarg0\fR to the interpreter. If \fIarg\fR was specified in the interpreter
file, it is passed as \fIarg1\fR to the interpreter. The remaining arguments to
the interpreter are \fIarg0\fR through \fIargn\fR of the originally exec’d
file. The interpreter named by \fIpathname\fR may also be an interpreter file.
There can be up to four nested interpreter files before the final interpreter.
The setid bits on nested interpreters are silently ignored.
.sp
.LP
When a C-language program is executed as a result of this call, it is entered
as a C-language function call as follows:
.sp
.in +2
.nf
int main (int argc, char *argv[]);
.fi
.in -2
```

```
.sp
.LP
where \fIargc\fR is the argument count and \fIargv\fR is an array of character
pointers to the arguments themselves. In addition, the following variable:
.sp
.in +2
.nf
extern char **environ;
.fi
.in -2

.sp
.LP
is initialized as a pointer to an array of character pointers to the
environment strings. The \fIargv\fR and \fIenviron\fR arrays are each
terminated by a null pointer. The null pointer terminating the \fIargv\fR array
is not counted in \fIargc\fR.
.sp
.LP
The value of \fIargc\fR is non-negative, and if greater than 0, \fIargv\fR[0]
points to a string containing the name of the file. If \fIargc\fR is 0,
\fIargv\fR[0] is a null pointer, in which case there are no arguments.
Applications should verify that \fIargc\fR is greater than 0 or that
\fIargv\fR[0] is not a null pointer before dereferencing \fIargv\fR[0].
.sp
.LP
The arguments specified by a program with one of the \fBexec\fR functions are
passed on to the new process image in the \fBmain()\fR arguments.
.sp
.LP
The \fIpath\fR argument points to a path name that identifies the new process
image file.
.sp
.LP
The \fIfile\fR argument is used to construct a pathname that identifies the new
process image file. If the \fIfile\fR argument contains a slash character, it
is used as the pathname for this file. Otherwise, the path prefix for this file
is obtained by a search of the directories passed in the \fBPATH\fR environment
variable (see \fBenviron\fR(5)). The environment is supplied typically by the
shell. If the process image file is not a valid executable object file,
\fBexeclp()\fR and \fBexecvp()\fR use the contents of that file as standard
input to the shell. In this case, the shell becomes the new process image. The
standard to which the caller conforms determines which shell is used. See
\fBstandards\fR(5).
.sp
.LP
The arguments represented by \fIarg0\fR\&.\|.\|. are pointers to
null-terminated character strings. These strings constitute the argument list
available to the new process image. The list is terminated by a null pointer.
The \fIarg0\fR argument should point to a filename that is associated with the
process being started by one of the \fBexec\fR functions.
.sp
.LP
The \fIargv\fR argument is an array of character pointers to null-terminated
strings. The last member of this array must be a null pointer. These strings
constitute the argument list available to the new process image. The value in
\fIargv\fR[0] should point to a filename that is associated with the process
being started by one of the \fBexec\fR functions.
.sp
.LP
The \fIenvp\fR argument is an array of character pointers to null-terminated
strings. These strings constitute the environment for the new process image.
The \fIenvp\fR array is terminated by a null pointer. For \fBexecl()\fR,
\fBexecv()\fR, \fBexecvp()\fR, and \fBexeclp()\fR, the C-language run-time
start-off routine places a pointer to the environment of the calling process in
the global object \fBextern char **environ\fR, and it is used to pass the
```



```
environment of the calling process to the new process image.
.sp
.LP
The number of bytes available for the new process's combined argument and
environment lists is \fBARG_MAX\fR. It is implementation-dependent whether null
terminators, pointers, and/or any alignment bytes are included in this total.
.sp
.LP
File descriptors open in the calling process image remain open in the new
process image, except for those whose close-on-exec flag \fBFD_CLOEXEC\fR is
set; see \fBfcntl\fR(2). For those file descriptors that remain open, all
attributes of the open file description, including file locks, remain
unchanged.
.sp
.LP
The preferred hardware address translation size (see \fBmemcntl\fR(2)) for the
stack and heap of the new process image are set to the default system page
size.
.sp
.LP
Directory streams open in the calling process image are closed in the new
process image.
.sp
.LP
The state of conversion descriptors and message catalogue descriptors in the
new process image is undefined. For the new process, the equivalent of:
.sp
.in +2
.nf
setlocale(LC_ALL, "C")
.fi
.in -2

.sp
.LP
is executed at startup.
.sp
.LP
Signals set to the default action (\fBSIG_DFL\fR) in the calling process image
are set to the default action in the new process image (see \fBsignal\fR(3C)).
Signals set to be ignored (\fBSIG_IGN\fR) by the calling process image are set
to be ignored by the new process image. Signals set to be caught by the calling
process image are set to the default action in the new process image (see
\fBsignal.h\fR(3HEAD)). After a successful call to any of the \fBexec\fR
functions, alternate signal stacks are not preserved and the \fBSA_ONSTACK\fR
flag is cleared for all signals.
.sp
.LP
After a successful call to any of the \fBexec\fR functions, any functions
previously registered by \fBatexit\fR(3C) are no longer registered.
.sp
.LP
The saved resource limits in the new process image are set to be a copy of the
process's corresponding hard and soft resource limits.
.sp
.LP
If the \fBST_NOSUID\fR bit is set for the file system containing the new
process image file, then the effective user \fBID\fR and effective group
\fBID\fR are unchanged in the new process image. If the set-user-\fBID\fR mode
bit of the new process image file is set (see \fBchmod\fR(2)), the effective
user \fBID\fR of the new process image is set to the owner \fBID\fR of the new
process image file. Similarly, if the set-group-\fBID\fR mode bit of the new
process image file is set, the effective group \fBID\fR of the new process
image is set to the group \fBID\fR of the new process image file. The real user
\fBID\fR and real group \fBID\fR of the new process image remain the same as
those of the calling process image. The effective user ID and effective group
```

ID of the new process image are saved (as the saved set-user-ID and the saved set-group-ID for use by \fBsetuid\fR(2).

.sp

.LP

The privilege sets are changed according to the following rules:

.RS +4

.TP

1.

The inheritable set, I, is intersected with the limit set, L. This mechanism enforces the limit set for processes.

.RE

.RS +4

.TP

2.

The effective set, E, and the permitted set, P, are made equal to the new inheritable set.

.RE

.sp

.LP

The system attempts to set the privilege-aware state to non-PA both before performing any modifications to the process IDs and privilege sets as well as after completing the transition to new UIDs and privilege sets, following the rules outlined in \fBprivileges\fR(5).

.sp

.LP

If the {\fBPRIV_PROC_OWNER\fR} privilege is asserted in the effective set, the set-user-ID and set-group-ID bits will be honored when the process is being controlled by \fBptrace\fR(3C). Additional restriction can apply when the traced process has an effective UID of 0. See \fBprivileges\fR(5).

.sp

.LP

Any shared memory segments attached to the calling process image will not be attached to the new process image (see \fBshmop\fR(2)). Any mappings established through \fBmmap()\fR are not preserved across an \fBexec\fR. Memory mappings created in the process are unmapped before the address space is rebuilt for the new process image. See \fBmmap\fR(2).

.sp

.LP

Memory locks established by the calling process via calls to \fBmlockall\fR(3C) or \fBmlock\fR(3C) are removed. If locked pages in the address space of the calling process are also mapped into the address spaces the locks established by the other processes will be unaffected by the call by this process to the \fBexec\fR function. If the \fBexec\fR function fails, the effect on memory locks is unspecified.

.sp

.LP

If \fB_XOPEN_REALTIME\fR is defined and has a value other than \(mi1, any named semaphores open in the calling process are closed as if by appropriate calls to \fBsem_close\fR(3C)

.sp

.LP

Profiling is disabled for the new process; see \fBprofil\fR(2).

.sp

.LP

Timers created by the calling process with \fBtimer_create\fR(3C) are deleted before replacing the current process image with the new process image.

.sp

.LP

For the \fBSCHED_FIFO\fR and \fBSCHED_RR\fR scheduling policies, the policy and priority settings are not changed by a call to an \fBexec\fR function.

.sp

.LP

All open message queue descriptors in the calling process are closed, as described in \fBmq_close\fR(3C).

.sp

.LP

Any outstanding asynchronous I/O operations may be cancelled. Those asynchronous I/O operations that are not canceled will complete as if the \fBexec\fR function had not yet occurred, but any associated signal notifications are suppressed. It is unspecified whether the \fBexec\fR function itself blocks awaiting such I/O completion. In no event, however, will the new process image created by the \fBexec\fR function be affected by the presence of outstanding asynchronous I/O operations at the time the \fBexec\fR function is called.

.sp

.LP

All active contract templates are cleared (see \fBcontract\fR(4)).

.sp

.LP

The new process also inherits the following attributes from the calling process:

.RS +4

.TP

.ie t \(bu

.el o

controlling terminal

.RE

.RS +4

.TP

.ie t \(bu

.el o

current working directory

.RE

.RS +4

.TP

.ie t \(bu

.el o

file-locks (see \fBfcntl\fR(2) and \fBlockf\fR(3C))

.RE

.RS +4

.TP

.ie t \(bu

.el o

file mode creation mask (see \fBumask\fR(2))

.RE

.RS +4

.TP

.ie t \(bu

.el o

file size limit (see \fBulimit\fR(2))

.RE

.RS +4

.TP

.ie t \(bu

.el o

limit privilege set

.RE

.RS +4

.TP

.ie t \(bu

.el o

nice value (see \fBnice\fR(2))

.RE

.RS +4

.TP

.ie t \(bu

.el o

parent process \fBID\fR

.RE

.RS +4

.TP

.ie t \(bu

```
.el o
pending signals (see \fBsigpending\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
privilege debugging flag (see \fBprivileges\fR(5) and \fBgetpflags\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
process \fBID\fR
.RE
.RS +4
.TP
.ie t \(bu
.el o
process contract (see \fBcontract\fR(4) and \fBprocess\fR(4))
.RE
.RS +4
.TP
.ie t \(bu
.el o
process group \fBID\fR
.RE
.RS +4
.TP
.ie t \(bu
.el o
process signal mask (see \fBsigprocmask\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
processor bindings (see \fBprocessor_bind\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
processor set bindings (see \fBpset_bind\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
project \fBID\fR
.RE
.RS +4
.TP
.ie t \(bu
.el o
real group \fBID\fR
.RE
.RS +4
.TP
.ie t \(bu
.el o
real user \fBID\fR
.RE
.RS +4
.TP
.ie t \(bu
```



```
.el o
resource limits (see \fBgetrlimit\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
root directory
.RE
.RS +4
.TP
.ie t \(bu
.el o
scheduler class and priority (see \fBpriocntl\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fBsemadj\fR values (see \fBsemop\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
session membership (see \fBexit\fR(2) and \fBsignal\fR(3C))
.RE
.RS +4
.TP
.ie t \(bu
.el o
supplementary group \fBIDs\fR
.RE
.RS +4
.TP
.ie t \(bu
.el o
task \fBID\fR
.RE
.RS +4
.TP
.ie t \(bu
.el o
time left until an alarm clock signal (see \fBalarm\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fBtms_utime\fR, \fBtms_stime\fR, \fBtms_cutime\fR, and \fBtms_cstime\fR (see
\fBtimes\fR(2))
.RE
.RS +4
.TP
.ie t \(bu
.el o
trace flag (see \fBptrace\fR(3C) request 0)
.RE
.sp
.LP
A call to any \fBexec\fR function from a process with more than one thread
results in all threads being terminated and the new executable image being
loaded and executed. No destructor functions will be called.
.sp
.LP
Upon successful completion, each of the functions in the \fBexec\fR family
```

```
524 marks for update the \fBst_atime\fR field of the file.  If an \fBexec\fR
525 function failed but was able to locate the \fBprocess image file\fR, whether
526 the \fBst_atime\fR field is marked for update is unspecified. Should the
527 function succeed, the process image file is considered to have been opened with
528 \fBopen\fR(2). The corresponding \fBclose\fR(2) is considered to occur at a
529 time after this open, but before process termination or successful completion
530 of a subsequent call to one of the \fBexec\fR functions. The \fIargv\fR[\|] and
531 \fIenvp\fR[\|] arrays of pointers and the strings to which those arrays point
532 will not be modified by a call to one of the \fBexec\fR functions, except as a
533 consequence of replacing the process image.
534 .sp
535 .LP
536 The saved resource limits in the new process image are set to be a copy of the
537 process's corresponding hard and soft limits.
538 .SH RETURN VALUES
539 .LP
540 If a function in the \fBexec\fR family returns to the calling process image, an
541 error has occurred; the return value is \fB\(mi1\fR and \fBerrno\fR is set to
542 indicate the error.
543 .SH ERRORS
544 .LP
545 The \fBexec\fR functions will fail if:
546 .sp
547 .ne 2
548 .na
549 \fB\fBE2BIG\fR\fR
550 .ad
551 .RS 16n
552 The number of bytes in the new process's argument list is greater than the
553 system-imposed limit of {\fBARG_MAX\fR} bytes. The argument list limit is sum
554 of the size of the argument list plus the size of the environment's exported
555 shell variables.
556 .RE

558 .sp
559 .ne 2
560 .na
561 \fB\fBEACCES\fR\fR
562 .ad
563 .RS 16n
564 Search permission is denied for a directory listed in the new process file's
565 path prefix.
566 .sp
567 The new process file is not an ordinary file.
568 .sp
569 The new process file mode denies execute permission.
570 .sp
571 The {\fBFILE_DAC_SEARCH\fR} privilege overrides the restriction on directory
572 searches.
573 .sp
574 The {\fBFILE_DAC_EXECUTE\fR} privilege overrides the lack of execute
575 permission.
576 .RE

578 .sp
579 .ne 2
580 .na
581 \fB\fBEAGAIN\fR\fR
582 .ad
583 .RS 16n
584 Total amount of system memory available when reading using raw I/O is
585 temporarily insufficient.
586 .RE

588 .sp
589 .ne 2
```



```
590 .na
591 \fB\fBEFAULT\fR\fR
592 .ad
593 .RS 16n
594 An argument points to an illegal address.
595 .RE

597 .sp
598 .ne 2
599 .na
600 \fB\fBEINVAL\fR\fR
601 .ad
602 .RS 16n
603 The new process image file has the appropriate permission and has a recognized
604 executable binary format, but the system does not support execution of a file
605 with this format.
606 .RE

608 .sp
609 .ne 2
610 .na
611 \fB\fBEINTR\fR\fR
612 .ad
613 .RS 16n
614 A signal was caught during the execution of one of the functions in the
615 \fIexec\fR family.
616 .RE

618 .sp
619 .ne 2
620 .na
621 \fB\fBELOOP\fR\fR
622 .ad
623 .RS 16n
624 Too many symbolic links were encountered in translating \fIpath\fR or
625 \fIfile\fR, or too many nested interpreter files.
626 .RE

628 .sp
629 .ne 2
630 .na
631 \fB\fBENAMETOOLONG\fR\fR
632 .ad
633 .RS 16n
634 The length of the \fIfile\fR or \fIpath\fR argument exceeds {\fBPATH_MAX\fR},
635 or the length of a \fIfile\fR or \fIpath\fR component exceeds {\fBNAME_MAX\fR}
636 while {\fB_POSIX_NO_TRUNC\fR} is in effect.
637 .RE

639 .sp
640 .ne 2
641 .na
642 \fB\fBENOENT\fR\fR
643 .ad
644 .RS 16n
645 One or more components of the new process path name of the file do not exist or
646 is a null pathname.
647 .RE

649 .sp
650 .ne 2
651 .na
652 \fB\fBENOLINK\fR\fR
653 .ad
654 .RS 16n
655 The \fIpath\fR argument points to a remote machine and the link to that machine
```

```
656 is no longer active.
657 .RE

659 .sp
660 .ne 2
661 .na
662 \fB\fBENOTDIR\fR\fR
663 .ad
664 .RS 16n
665 A component of the new process path of the file prefix is not a directory.
666 .RE

668 .sp
669 .LP
670 The \fBexec\fR functions, except for \fBexeclp()\fR and \fBexecvp()\fR, will
671 fail if:
672 .sp
673 .ne 2
674 .na
675 \fB\fBENOEXEC\fR\fR
676 .ad
677 .RS 11n
678 The new process image file has the appropriate access permission but is not in
679 the proper format.
680 .RE

682 .sp
683 .LP
684 The \fBexec\fR functions may fail if:
685 .sp
686 .ne 2
687 .na
688 \fB\fBENAMETOOLONG\fR\fR
689 .ad
690 .RS 16n
691 Pathname resolution of a symbolic link produced an intermediate result whose
692 length exceeds {\fBPATH_MAX\fR}.
693 .RE

695 .sp
696 .ne 2
697 .na
698 \fB\fBENOMEM\fR\fR
699 .ad
700 .RS 16n
701 The new process image requires more memory than is allowed by the hardware or
702 system-imposed by memory management constraints. See \fBbrk\fR(2).
703 .RE

705 .sp
706 .ne 2
707 .na
708 \fB\fBETXTBSY\fR\fR
709 .ad
710 .RS 16n
711 The new process image file is a pure procedure (shared text) file that is
712 currently open for writing by some process.
713 .RE

715 .SH USAGE
716 .LP
717 As the state of conversion descriptors and message catalogue descriptors in the
718 new process image is undefined, portable applications should not rely on their
719 use and should close them prior to calling one of the \fBexec\fR functions.
720 .sp
721 .LP
```



```
722 Applications that require other than the default POSIX locale should call
723 \fBsetlocale\fR(3C) with the appropriate parameters to establish the locale of
724 thenew process.
725 .sp
726 .LP
727 The \fIenviron\fR array should not be accessed directly by the application.
728 .SH ATTRIBUTES
729 .LP
730 See \fBattributes\fR(5) for descriptions of the following attributes:
731 .sp

733 .sp
734 .TS
735 box;
736 c | c
737 l | l .
738 ATTRIBUTE TYPE  ATTRIBUTE VALUE
739 _
740 Interface Stability     Committed
741 _
742 MT-Level        See below.
743 _
744 Standard        See \fBstandards\fR(5).
745 .TE

747 .sp
748 .LP
749 The \fBexecle()\fR and \fBexecve()\fR functions are Async-Signal-Safe.
749 The \fBexecle()\fR and \fBexecve()\fR fucntions are Async-Signal-Safe.
750 .SH SEE ALSO
751 .LP
752 \fBksh\fR(1), \fBps\fR(1), \fBsh\fR(1), \fBalarm\fR(2), \fBbrk\fR(2),
753 \fBchmod\fR(2), \fBexit\fR(2), \fBfcntl\fR(2), \fBfork\fR(2),
754 \fBgetpflags\fR(2), \fBgetrlimit\fR(2), \fBmemcntl\fR(2), \fBmmap\fR(2),
755 \fBnice\fR(2), \fBpriocntl\fR(2), \fBprofil\fR(2), \fBsemop\fR(2),
756 \fBshmop\fR(2), \fBsigpending\fR(2), \fBsigprocmask\fR(2), \fBtimes\fR(2),
757 \fBumask\fR(2), \fBlockf\fR(3C), \fBptrace\fR(3C), \fBsetlocale\fR(3C),
758 \fBsignal\fR(3C), \fBsystem\fR(3C), \fBtimer_create\fR(3C), \fBa.out\fR(4),
759 \fBcontract\fR(4), \fBprocess\fR(4), \fBattributes\fR(5), \fBenviron\fR(5),
760 \fBprivileges\fR(5), \fBstandards\fR(5)
761 .SH WARNINGS
762 .LP
763 If a program is \fBsetuid\fR to a user \fBID\fR other than the superuser, and
764 the program is executed when the real user \fBID\fR is super-user, then the
765 program has some of the powers of a super-user as well.
```

```
**********************************************************
    2494 Sun Sep 16 19:22:54 2018
new/usr/src/man/man2/getisax.2
9842 man page typos and spelling
**********************************************************
   1 '\" te
   2 .\" Copyright (c) 2007, Sun Microsystems, Inc. All Rights Reserved.
   3 .\" The contents of this file are subject to the terms of the Common Development
   4 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
   5 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
   6 .TH GETISAX 2 "Nov 7, 2007"
   7 .SH NAME
   8 getisax \- extract valid instruction set extensions
   9 .SH SYNOPSIS
  10 .LP
  11 .nf
  12 #include <sys/auxv.h>

  14 \fBuint_t\fR \fBgetisax\fR(\fBuint32_t *\fR\fIarray\fR, \fBuint_t\fR \fIn\fR);
  15 .fi

  17 .SH DESCRIPTION
  18 .sp
  18 .LP
  19 The \fBgetisax()\fR function sets the vector \fIarray\fR of \fIn\fR 32-bit
  20 integers to contain the bits from the \fBAV\fR_\fIxxx\fR_\fIyyy\fR namespace of
  21 the given instruction set architecture.
  22 .sp
  23 .LP
  24 Values for \fBAV\fR_\fIxxx\fR_\fIyyy\fR for SPARC and SPARCV9, and their
  25 associated descriptions, can be found in \fB<sys/auxv_SPARC.h>\fR.
  26 .sp
  27 .LP
  28 Values for \fBAV\fR_\fIxxx\fR_\fIyyy\fR for i386 and AMD64, and their
  29 associated descriptions, can be found in \fB<sys/auxv_386.h>\fR.
  30 .SH RETURN VALUES
  32 .sp
  31 .LP
  32 The \fBgetisax()\fR function returns the number of array elements that contain
  33 non-zero values.
  34 .SH EXAMPLES
  35 .LP
  36 \fBExample 1 \fRUse \fBgetisax()\fR to determine if the SSE2 instruction set is
  37 present.
  38 .sp
  39 .LP
  40 In the following example, if the message is written, the SSE2 instruction set
  41 is present and fully supported by the operating system.
  43 is present and fully supportred by the operating system.

  43 .sp
  44 .in +2
  45 .nf
  46 uint_t ui;

  48 (void) getisax(&ui, 1);

  50 if (ui & AV_386_SSE2)
  51         printf("SSE2 instruction set extension is present.\en");
  52 .fi
  53 .in -2

  55 .SH ATTRIBUTES
  58 .sp
  56 .LP
  57 See \fBattributes\fR(5) for descriptions of the following attributes:
```



```
  58 .sp

  60 .sp
  61 .TS
  62 box;
  63 c | c
  64 l | l .
  65 ATTRIBUTE TYPE  ATTRIBUTE VALUE
  66 _
  67 Interface Stability     Committed
  68 _
  69 MT-Level        Safe
  70 .TE

  72 .SH SEE ALSO
  76 .sp
  73 .LP
  74 \fBisainfo\fR(1), \fBld\fR(1), \fBpargs\fR(1), \fBattributes\fR(5)
  75 .sp
  76 .LP
  77 \fILinker and Libraries Guide\fR
  78 .sp
  79 .LP
  80 \fISPARC Assembly Language Reference Manual\fR
  81 .sp
  82 .LP
  83 \fIx86 Assembly Language Reference Manual\fR
```

```
**********************************************************
   26146 Sun Sep 16 19:22:54 2018
new/usr/src/man/man2/mmap.2
9842 man page typos and spelling
**********************************************************
.\"
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for
.\" permission to reproduce portions of its copyrighted documentation.
.\" Original documentation from The Open Group can be obtained online at
.\" http://www.opengroup.org/bookstore/.
.\"
.\" The Institute of Electrical and Electronics Engineers and The Open
.\" Group, have given us permission to reprint portions of their
.\" documentation.
.\"
.\" In the following statement, the phrase ‘‘this text’’ refers to portions
.\" of the system documentation.
.\"
.\" Portions of this text are reprinted and reproduced in electronic form
.\" in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition,
.\" Standard for Information Technology -- Portable Operating System
.\" Interface (POSIX), The Open Group Base Specifications Issue 6,
.\" Copyright (C) 2001-2004 by the Institute of Electrical and Electronics
.\" Engineers, Inc and The Open Group.  In the event of any discrepancy
.\" between these versions and the original IEEE and The Open Group
.\" Standard, the original IEEE and The Open Group Standard is the referee
.\" document.  The original Standard can be obtained online at
.\" http://www.opengroup.org/unix/online.html.
.\"
.\" This notice shall appear on any product containing this material.
.\"
.\" The contents of this file are subject to the terms of the
.\" Common Development and Distribution License (the "License").
.\" You may not use this file except in compliance with the License.
.\"
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
.\" or http://www.opensolaris.org/os/licensing.
.\" See the License for the specific language governing permissions
.\" and limitations under the License.
.\"
.\" When distributing Covered Code, include this CDDL HEADER in each
.\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
.\" If applicable, add the following below this CDDL HEADER, with the
.\" fields enclosed by brackets "[]" replaced with your own identifying
.\" information: Portions Copyright [yyyy] [name of copyright owner]
.\"
.\"
.\" Copyright 1989 AT&T
.\" Portions Copyright (c) 1992, X/Open Company Limited.  All Rights Reserved.
.\" Copyright (c) 2009, Sun Microsystems, Inc.  All Rights Reserved.
.\" Copyright 2013 OmniTI Computer Consulting, Inc. All Rights Reserved.
.\" Copyright 2016 James S Blachly, MD. All Rights Reserved.
.\"
.TH MMAP 2 "August 29, 2016"
.SH NAME
mmap \- map pages of memory
.SH SYNOPSIS
.LP
.nf
#include <sys/mman.h>

\fBvoid *\fR\fBmmap\fR(\fBvoid *\fR\fIaddr\fR, \fBsize_t\fR \fIlen\fR, \fBint\fR
     \fBint\fR \fIfildes\fR, \fBoff_t\fR \fIoff\fR);
.fi

.SH DESCRIPTION
```



```
.LP
The \fBmmap()\fR function establishes a mapping between a process’s address
space and a file or shared memory object. The format of the call is as follows:
.sp
.LP
\fIpa\fR = \fBmmap(\fR\fIaddr\fR\fB, \fR\fIlen\fR\fB, \fR\fIprot\fR\fB,
\fR\fIflags\fR\fB, \fR\fIfildes\fR\fB, \fR\fIoff\fR\fB);\fR
.sp
.LP
The \fBmmap()\fR function establishes a mapping between the address space of
the process at an address \fIpa\fR for \fIlen\fR bytes to the memory object
represented by the file descriptor \fIfildes\fR at offset \fIoff\fR for
\fIlen\fR bytes. The value of \fIpa\fR is a function of the  \fIaddr\fR
argument and values of \fIflags\fR, further described below. A successful
\fBmmap()\fR call returns \fIpa\fR as its result. The address range starting at
\fIpa\fR and continuing for \fIlen\fR bytes will be legitimate for the possible
(not necessarily current) address space of the process. The range of bytes
starting at \fIoff\fR and continuing for \fIlen\fR bytes will be legitimate for
the possible (not necessarily current) offsets in the file or shared memory
object represented by \fIfildes\fR.
.sp
.LP
The \fBmmap()\fR function allows [\fIpa, pa + len\fR) to extend beyond the end
of the object both at the time of the \fBmmap()\fR and while the mapping
persists, such as when the file is created prior to the \fBmmap()\fR call and
has no contents, or when the file is truncated. Any reference to addresses
beyond the end of the object, however, will result in the delivery of a
\fBSIGBUS\fR or \fBSIGSEGV\fR signal. The \fBmmap()\fR function cannot be used
to implicitly extend the length of files.
.sp
.LP
The mapping established by \fBmmap()\fR replaces any previous mappings for
those whole pages containing any part of the address space of the process
starting at \fIpa\fR and continuing for \fIlen\fR bytes.
.sp
.LP
If the size of the mapped file changes after the call to \fBmmap()\fR as a
result of some other operation on the mapped file, the effect of references to
portions of the mapped region that correspond to added or removed portions of
the file is unspecified.
.sp
.LP
The \fBmmap()\fR function is supported for regular files and shared memory
objects. Support for any other type of file is unspecified.
.sp
.LP
The  \fIprot\fR argument determines whether read, write, execute, or some
combination of accesses are permitted to the data being mapped. The \fIprot\fR
argument should be either \fBPROT_NONE\fR or the bitwise inclusive \fBOR\fR of
one or more of the other flags in the following table, defined in the header
<\fBsys/mman.h\fR>.
.sp
.ne 2
.na
\fB\fBPROT_READ\fR\fR
.ad
.RS 14n
Data can be read.
.RE

.sp
.ne 2
.na
\fB\fBPROT_WRITE\fR\fR
.ad
.RS 14n
```

```
128 Data can be written.
129 .RE

131 .sp
132 .ne 2
133 .na
134 \fB\fBPROT_EXEC\fR\fR
135 .ad
136 .RS 14n
137 Data can be executed.
138 .RE

140 .sp
141 .ne 2
142 .na
143 \fB\fBPROT_NONE\fR\fR
144 .ad
145 .RS 14n
146 Data cannot be accessed.
147 .RE

149 .sp
150 .LP
151 If an implementation of \fBmmap()\fR for a specific platform cannot support the
152 combination of access types specified by \fIprot\fR, the call to \fBmmap()\fR
153 fails. An implementation may permit accesses other than those specified by
154 \fIprot\fR; however, the implementation will not permit a write to succeed
155 where \fBPROT_WRITE\fR has not been set or permit any access where
156 \fBPROT_NONE\fR alone has been set. Each platform-specific implementation of
157 \fBmmap()\fR supports the following values of \fIprot\fR: \fBPROT_NONE\fR,
158 \fBPROT_READ\fR, \fBPROT_WRITE\fR, and the inclusive \fBOR\fR of
159 \fBPROT_READ\fR and \fBPROT_WRITE\fR. On some platforms, the \fBPROT_WRITE\fR
160 protection option is implemented as \fBPROT_READ|PROT_WRITE\fR and
161 \fBPROT_EXEC\fR as \fBPROT_READ|PROT_EXEC\fR. The file descriptor \fIfildes\fR
162 is opened with read permission, regardless of the protection options specified.
163 If \fBPROT_WRITE\fR is specified, the application must have opened the file
164 descriptor \fIfildes\fR with write permission unless \fBMAP_PRIVATE\fR is
165 specified in the \fIflags\fR argument as described below.
166 .sp
167 .LP
168 The  \fIflags\fR argument provides other information about the handling of the
169 mapped data. The value of \fIflags\fR is the bitwise inclusive \fBOR\fR of
170 these options, defined in <\fBsys/mman.h\fR>:
171 .sp
172 .ne 2
173 .na
174 \fB\fBMAP_SHARED\fR\fR
175 .ad
176 .RS 17n
177 Changes are shared.
178 .RE

180 .sp
181 .ne 2
182 .na
183 \fB\fBMAP_PRIVATE\fR\fR
184 .ad
185 .RS 17n
186 Changes are private.
187 .RE

189 .sp
190 .ne 2
191 .na
192 \fB\fBMAP_FIXED\fR\fR
193 .ad
```



```
194 .RS 17n
195 Interpret \fIaddr\fR exactly.
196 .RE

198 .sp
199 .ne 2
200 .na
201 \fB\fBMAP_NORESERVE\fR\fR
202 .ad
203 .RS 17n
204 Do not reserve swap space.
205 .RE

207 .sp
208 .ne 2
209 .na
210 \fB\fBMAP_ANON\fR\fR
211 .ad
212 .RS 17n
213 Map anonymous memory.
214 .RE

216 .sp
217 .ne 2
218 .na
219 \fB\fBMAP_ALIGN\fR\fR
220 .ad
221 .RS 17n
222 Interpret \fIaddr\fR as required alignment.
222 Interpret \fIaddr\fR as required aligment.
223 .RE

225 .sp
226 .ne 2
227 .na
228 \fB\fBMAP_TEXT\fR\fR
229 .ad
230 .RS 17n
231 Map text.
232 .RE

234 .sp
235 .ne 2
236 .na
237 \fB\fBMAP_INITDATA\fR\fR
238 .ad
239 .RS 17n
240 Map initialized data segment.
241 .RE

243 .sp
244 .ne 2
245 .na
246 \fB\fBMAP_32BIT\fR\fR
247 .ad
248 .RS 17n
249 Map to the lower 32 bits of address space.
250 .RE

252 .sp
253 .ne 2
254 .na
255 \fB\fBMAP_FILE\fR\fR
256 .ad
257 .RS 17n
258 Map a regular file. This is the default behavior;
```

specifying this flag is not required. It is provided
for compatibility with other systems and should not be
included in new code.
.RE

.sp
.LP
The \fBMAP_SHARED\fR and \fBMAP_PRIVATE\fR options describe the disposition of write references to the underlying object. If \fBMAP_SHARED\fR is specified, write references will change the memory object. If \fBMAP_PRIVATE\fR is specified, the initial write reference will create a private copy of the memory object page and redirect the mapping to the copy. The private copy is not created until the first write; until then, other users who have the object mapped \fBMAP_SHARED\fR can change the object. Either \fBMAP_SHARED\fR or \fBMAP_PRIVATE\fR must be specified, but not both. The mapping type is retained across \fBfork\fR(2).
.sp
.LP
When \fBMAP_FIXED\fR is set in the \fIflags\fR argument, the system is informed that the value of \fIpa\fR must be \fIaddr\fR, exactly. If \fBMAP_FIXED\fR is set, \fBmmap()\fR may return (\fBvoid *\fR)\(mi1 and set \fBerrno\fR to \fBEINVAL\fR. If a \fBMAP_FIXED\fR request is successful, the mapping established by \fBmmap()\fR replaces any previous mappings for the process's pages in the range [\fIpa, pa + len\fR). The use of \fBMAP_FIXED\fR is discouraged, since it may prevent a system from making the most effective use of its resources.
.sp
.LP
When \fBMAP_FIXED\fR is set and the requested address is the same as previous mapping, the previous address is unmapped and the new mapping is created on top of the old one.
.sp
.LP
When \fBMAP_FIXED\fR is not set, the system uses \fIaddr\fR to arrive at \fIpa\fR. The \fIpa\fR so chosen will be an area of the address space that the system deems suitable for a mapping of \fIlen\fR bytes to the file. The \fBmmap()\fR function interprets an \fIaddr\fR value of 0 as granting the system complete freedom in selecting \fIpa\fR, subject to constraints described below. A non-zero value of \fIaddr\fR is taken to be a suggestion of a process address near which the mapping should be placed. When the system selects a value for \fIpa\fR, it will never place a mapping at address 0, nor will it replace any extant mapping, nor map into areas considered part of the potential data or stack "segments".
.sp
.LP
When \fBMAP_ALIGN\fR is set, the system is informed that the alignment of \fIpa\fR must be the same as \fIaddr\fR. The alignment value in \fIaddr\fR must be 0 or some power of two multiple of page size as returned by \fBsysconf\fR(3C). If addr is 0, the system will choose a suitable alignment.
.sp
.LP
The \fBMAP_NORESERVE\fR option specifies that no swap space be reserved for a mapping. Without this flag, the creation of a writable \fBMAP_PRIVATE\fR mapping reserves swap space equal to the size of the mapping; when the mapping is written into, the reserved space is employed to hold private copies of the data. A write into a \fBMAP_NORESERVE\fR mapping produces results which depend on the current availability of swap space in the system. If space is available, the write succeeds and a private copy of the written page is created; if space is not available, the write fails and a \fBSIGBUS\fR or \fBSIGSEGV\fR signal is delivered to the writing process. \fBMAP_NORESERVE\fR mappings are inherited across \fBfork()\fR; at the time of the \fBfork()\fR, swap space is reserved in the child for all private pages that currently exist in the parent; thereafter the child's mapping behaves as described above.
.sp
.LP
When \fBMAP_ANON\fR is set in \fIflags\fR, and \fIfildes\fR is set to -1,

\fBmmap()\fR provides a direct path to return anonymous pages to the caller. This operation is equivalent to passing \fBmmap()\fR an open file descriptor on \fB/dev/zero\fR with \fBMAP_ANON\fR elided from the \fIflags\fR argument.
.sp
.LP
The \fBMAP_TEXT\fR option informs the system that the mapped region will be used primarily for executing instructions. This information can help the system better utilize MMU resources on some platforms. This flag is always passed by the dynamic linker when it maps text segments of shared objects. When the \fBMAP_TEXT\fR option is used for regular file mappings on some platforms, the system can choose a mapping size larger than the page size returned by \fBsysconf\fR(3C). The specific page sizes that are used depend on the platform and the alignment of the addr and len arguments. Several different mapping sizes can be used to map the region with larger page sizes used in the parts of the region that meet alignment and size requirements for those page sizes.
.sp
.LP
The \fBMAP_INITDATA\fR option informs the system that the mapped region is an initialized data segment of an executable or shared object. When the \fBMAP_INITDATA\fR option is used for regular file mappings on some platforms, the system can choose a mapping size larger than the page size returned by \fBsysconf()\fR. The \fBMAP_INITDATA\fR option should be used only by the dynamic linker for mapping initialized data of shared objects.
.sp
.LP
The \fBMAP_32BIT\fR option informs the system that the search space for mapping assignment should be limited to the first 32 bits (4 Gbytes) of the caller's address space. This flag is accepted in both 32-bit and 64-bit process models, but does not alter the mapping strategy when used in a 32-bit process model.
.sp
.LP
The \fIoff\fR argument is constrained to be aligned and sized according to the value returned by \fBsysconf()\fR when passed \fB_SC_PAGESIZE\fR or \fB_SC_PAGE_SIZE\fR. When \fBMAP_FIXED\fR is specified, the \fIaddr\fR argument must also meet these constraints. The system performs mapping operations over whole pages. Thus, while the \fIlen\fR argument need not meet a size or alignment constraint, the system will include, in any mapping operation, any partial page specified by the range [\fIpa, pa + len\fR).
.sp
.LP
The system will always zero-fill any partial page at the end of an object. Further, the system will never write out any modified portions of the last page of an object which are beyond its end. References to whole pages following the end of an object will result in the delivery of a \fBSIGBUS\fR or \fBSIGSEGV\fR signal. \fBSIGBUS\fR signals may also be delivered on various file system conditions, including quota exceeded errors.
.sp
.LP
The \fBmmap()\fR function adds an extra reference to the file associated with the file descriptor \fIfildes\fR which is not removed by a subsequent \fBclose\fR(2) on that file descriptor. This reference is removed when there are no more mappings to the file by a call to the \fBmunmap\fR(2) function.
.sp
.LP
The \fBst_atime\fR field of the mapped file may be marked for update at any time between the \fBmmap()\fR call and the corresponding \fBmunmap\fR(2) call. The initial read or write reference to a mapped region will cause the file's \fBst_atime\fR field to be marked for update if it has not already been marked for update.
.sp
.LP
The \fBst_ctime\fR and \fBst_mtime\fR fields of a file that is mapped with \fBMAP_SHARED\fR and \fBPROT_WRITE\fR, will be marked for update at some point in the interval between a write reference to the mapped region and the next call to \fBmsync\fR(3C) with \fBMS_ASYNC\fR or \fBMS_SYNC\fR for that portion

```
391 of the file by any process.  If there is no such call, these fields may be
392 marked for update at any time after a write reference if the underlying file is
393 modified as a result.
394 .sp
395 .LP
396 If the process calls \fBmlockall\fR(3C) with the \fBMCL_FUTURE\fR flag, the
397 pages mapped by all future calls to \fBmmap()\fR will be locked in memory. In
398 this case, if not enough memory could be locked, \fBmmap()\fR fails and sets
399 \fBerrno\fR to \fBEAGAIN\fR.
400 .sp
401 .LP
402 The \fBmmap()\fR function aligns based on the length of the mapping. When
403 determining the amount of space to add to the address space, \fBmmap()\fR
404 includes two 8-Kbyte pages, one at each end of the mapping that are not mapped
405 and are therefore used as "red-zone" pages. Attempts to reference these pages
406 result in access violations.
407 .sp
408 .LP
409 The size requested is incremented by the 16 Kbytes for these pages and is then
410 subject to rounding constraints. The constraints are:
411 .RS +4
412 .TP
413 .ie t \(bu
414 .el o
415 For 32-bit processes:
416 .sp
417 .in +2
418 .nf
419 If length > 4 Mbytes
420         round to 4-Mbyte multiple
421 elseif length > 512 Kbytes
422         round to 512-Kbyte multiple
423 else
424         round to 64-Kbyte multiple
425 .fi
426 .in -2

428 .RE
429 .RS +4
430 .TP
431 .ie t \(bu
432 .el o
433 For 64-bit processes:
434 .sp
435 .in +2
436 .nf
437 If length > 4 Mbytes
438         round to 4-Mbyte multiple
439 else
440         round to 1-Mbyte multiple
441 .fi
442 .in -2

444 .RE
445 .sp
446 .LP
447 The net result is that for a 32-bit process:
448 .RS +4
449 .TP
450 .ie t \(bu
451 .el o
452 If an \fBmmap()\fR request is made for 4 Mbytes, it results in 4 Mbytes + 16
453 Kbytes and is rounded up to 8 Mbytes.
454 .RE
455 .RS +4
456 .TP
```



```
457 .ie t \(bu
458 .el o
459 If an \fBmmap()\fR request is made for 512 Kbytes, it results in 512 Kbytes +
460 16 Kbytes and is rounded up to 1 Mbyte.
461 .RE
462 .RS +4
463 .TP
464 .ie t \(bu
465 .el o
466 If an \fBmmap()\fR request is made for 1 Mbyte, it results in 1 Mbyte + 16
467 Kbytes and is rounded up to 1.5 Mbytes.
468 .RE
469 .RS +4
470 .TP
471 .ie t \(bu
472 .el o
473 Each 8-Kbyte mmap request "consumes" 64 Kbytes of virtual address space.
474 .RE
475 .sp
476 .LP
477 To obtain maximal address space usage for a 32-bit process:
478 .RS +4
479 .TP
480 .ie t \(bu
481 .el o
482 Combine 8-Kbyte requests up to a limit of 48 Kbytes.
483 .RE
484 .RS +4
485 .TP
486 .ie t \(bu
487 .el o
488 Combine amounts over 48 Kbytes into 496-Kbyte chunks.
489 .RE
490 .RS +4
491 .TP
492 .ie t \(bu
493 .el o
494 Combine amounts over 496 Kbytes into 4080-Kbyte chunks.
495 .RE
496 .sp
497 .LP
498 To obtain maximal address space usage for a 64-bit process:
499 .RS +4
500 .TP
501 .ie t \(bu
502 .el o
503 Combine amounts < 1008 Kbytes into chunks <= 1008 Kbytes.
504 .RE
505 .RS +4
506 .TP
507 .ie t \(bu
508 .el o
509 Combine amounts over 1008 Kbytes into 4080-Kbyte chunks.
510 .RE
511 .sp
512 .LP
513 The following is the output from a 32-bit program demonstrating this:
514 .sp
515 .ne 2
516 .na
517 \fBmap 8192 bytes: \fB0xff390000\fR\fR
518 .ad
519 .br
520 .na
521 \fBmap 8192 bytes: \fB0xff380000\fR\fR
522 .ad
```

```
523 .sp .6
524 .RS 4n
525 64-Kbyte delta between starting addresses.
526 .RE

528 .sp
529 .ne 2
530 .na
531 \fBmap 512 Kbytes: \fB0xff180000\fR\fR
532 .ad
533 .br
534 .na
535 \fBmap 512 Kbytes: \fB0xff080000\fR\fR
536 .ad
537 .sp .6
538 .RS 4n
539 1-Mbyte delta between starting addresses.
540 .RE

542 .sp
543 .ne 2
544 .na
545 \fBmap 496 Kbytes: \fB0xff000000\fR\fR
546 .ad
547 .br
548 .na
549 \fBmap 496 Kbytes: \fB0xfef80000\fR\fR
550 .ad
551 .sp .6
552 .RS 4n
553 512-Kbyte delta between starting addresses
554 .RE

556 .sp
557 .ne 2
558 .na
559 \fBmap 1 Mbyte: \fB0xfee00000\fR\fR
560 .ad
561 .br
562 .na
563 \fBmap 1 Mbyte: \fB0xfec80000\fR\fR
564 .ad
565 .sp .6
566 .RS 4n
567 1536-Kbyte delta between starting addresses
568 .RE

570 .sp
571 .ne 2
572 .na
573 \fBmap 1008 Kbytes: \fB0xfeb80000\fR\fR
574 .ad
575 .br
576 .na
577 \fBmap 1008 Kbytes: \fB0xfea80000\fR\fR
578 .ad
579 .sp .6
580 .RS 4n
581 1-Mbyte delta between starting addresses
582 .RE

584 .sp
585 .ne 2
586 .na
587 \fBmap 4 Mbytes: \fB0xfe400000\fR\fR
588 .ad
```



```
589 .br
590 .na
591 \fBmap 4 Mbytes: \fB0xfdc00000\fR\fR
592 .ad
593 .sp .6
594 .RS 4n
595 8-Mbyte delta between starting addresses
596 .RE

598 .sp
599 .ne 2
600 .na
601 \fBmap 4080 Kbytes: \fB0xfd800000\fR\fR
602 .ad
603 .br
604 .na
605 \fBmap 4080 Kbytes: \fB0xfd400000\fR\fR
606 .ad
607 .sp .6
608 .RS 4n
609 4-Mbyte delta between starting addresses
610 .RE

612 .sp
613 .LP
614 The following is the output of the same program compiled as a 64-bit
615 application:
616 .sp
617 .ne 2
618 .na
619 \fBmap 8192 bytes: \fB0xfffffff7f000000\fR\fR
620 .ad
621 .br
622 .na
623 \fBmap 8192 bytes: \fB0xfffffff7ef00000\fR\fR
624 .ad
625 .sp .6
626 .RS 4n
627 1-Mbyte delta between starting addresses
628 .RE

630 .sp
631 .ne 2
632 .na
633 \fBmap 512 Kbytes: \fB0xfffffff7ee00000\fR\fR
634 .ad
635 .br
636 .na
637 \fBmap 512 Kbytes: \fB0xfffffff7ed00000\fR\fR
638 .ad
639 .sp .6
640 .RS 4n
641 1-Mbyte delta between starting addresses
642 .RE

644 .sp
645 .ne 2
646 .na
647 \fBmap 496 Kbytes: \fB0xfffffff7ec00000\fR\fR
648 .ad
649 .br
650 .na
651 \fBmap 496 Kbytes: \fB0xfffffff7eb00000\fR\fR
652 .ad
653 .sp .6
654 .RS 4n
```

```
655 1-Mbyte delta between starting addresses
656 .RE

658 .sp
659 .ne 2
660 .na
661 \fBmap 1 Mbyte: \fB0xffffffff7e900000\fR\fR
662 .ad
663 .br
664 .na
665 \fBmap 1 Mbyte: \fB0xffffffff7e700000\fR\fR
666 .ad
667 .sp .6
668 .RS 4n
669 2-Mbyte delta between starting addresses
670 .RE

672 .sp
673 .ne 2
674 .na
675 \fBmap 1008 Kbytes: \fB0xffffffff7e600000\fR\fR
676 .ad
677 .br
678 .na
679 \fBmap 1008 Kbytes: \fB0xffffffff7e500000\fR\fR
680 .ad
681 .sp .6
682 .RS 4n
683 1-Mbyte delta between starting addresses
684 .RE

686 .sp
687 .ne 2
688 .na
689 \fBmap 4 Mbytes: \fB0xffffffff7e000000\fR\fR
690 .ad
691 .br
692 .na
693 \fBmap 4 Mbytes: \fB0xffffffff7d800000\fR\fR
694 .ad
695 .sp .6
696 .RS 4n
697 8-Mbyte delta between starting addresses
698 .RE

700 .sp
701 .ne 2
702 .na
703 \fBmap 4080 Kbytes: \fB0xffffffff7d400000\fR\fR
704 .ad
705 .br
706 .na
707 \fBmap 4080 Kbytes: \fB0xffffffff7d000000\fR\fR
708 .ad
709 .sp .6
710 .RS 4n
711 4-Mbyte delta between starting addresses
712 .RE

714 .SH RETURN VALUES
715 .LP
716 Upon successful completion, the \fBmmap()\fR function returns the address at
717 which the mapping was placed (\fIpa\fR); otherwise, it returns a value of
718 \fBMAP_FAILED\fR and sets \fBerrno\fR to indicate the error. The symbol
719 \fBMAP_FAILED\fR is defined in the header <\fBsys/mman.h\fR>. No successful
720 return from \fBmmap()\fR will return the value \fBMAP_FAILED\fR.
```



```
721 .sp
722 .LP
723 If \fBmmap()\fR fails for reasons other than \fBEBADF\fR, \fBEINVAL\fR or
724 \fBENOTSUP\fR, some of the mappings in the address range starting at \fIaddr\fR
725 and continuing for \fIlen\fR bytes may have been unmapped.
726 .SH ERRORS
727 .LP
728 The \fBmmap()\fR function will fail if:
729 .sp
730 .ne 2
731 .na
732 \fB\fBEACCES\fR\fR
733 .ad
734 .RS 13n
735 The \fIfildes\fR file descriptor is not open for read, regardless of the
736 protection specified; or \fIfildes\fR is not open for write and
737 \fBPROT_WRITE\fR was specified for a \fBMAP_SHARED\fR type mapping.
738 .RE

740 .sp
741 .ne 2
742 .na
743 \fB\fBEAGAIN\fR\fR
744 .ad
745 .RS 13n
746 The mapping could not be locked in memory.
747 .sp
748 There was insufficient room to reserve swap space for the mapping.
749 .RE

751 .sp
752 .ne 2
753 .na
754 \fB\fBEBADF\fR\fR
755 .ad
756 .RS 13n
757 The \fIfildes\fR file descriptor is not open (and \fBMAP_ANON\fR was not
758 specified).
759 .RE

761 .sp
762 .ne 2
763 .na
764 \fB\fBEINVAL\fR\fR
765 .ad
766 .RS 13n
767 The arguments \fIaddr\fR (if \fBMAP_FIXED\fR was specified) or \fIoff\fR are
768 not multiples of the page size as returned by \fBsysconf()\fR.
769 .sp
770 The argument \fIaddr\fR (if \fBMAP_ALIGN\fR was specified) is not 0 or some
771 power of two multiple of page size as returned by \fBsysconf\fR(3C).
772 .sp
773 \fBMAP_FIXED\fR and \fBMAP_ALIGN\fR are both specified.
774 .sp
775 The field in \fIflags\fR is invalid (neither \fBMAP_PRIVATE\fR or
776 \fBMAP_SHARED\fR is set).
777 .sp
778 The argument \fIlen\fR has a value equal to 0.
779 .sp
780 \fBMAP_ANON\fR was specified, but the file descriptor was not \(mi1.
781 .sp
782 \fBMAP_TEXT\fR was specified but \fBPROT_EXEC\fR was not.
783 .sp
784 \fBMAP_TEXT\fR and \fBMAP_INITDATA\fR were both specified.
785 .RE
```

```
787 .sp
788 .ne 2
789 .na
790 \fB\fBEMFILE\fR\fR
791 .ad
792 .RS 13n
793 The number of mapped regions would exceed an implementation-dependent limit
794 (per process or per system).
795 .RE

797 .sp
798 .ne 2
799 .na
800 \fB\fBENODEV\fR\fR
801 .ad
802 .RS 13n
803 The \fIfildes\fR argument refers to an object for which \fBmmap()\fR is
804 meaningless, such as a terminal.
805 .RE

807 .sp
808 .ne 2
809 .na
810 \fB\fBENOMEM\fR\fR
811 .ad
812 .RS 13n
813 The \fBMAP_FIXED\fR option was specified and the range [\fIaddr, addr + len\fR)
814 exceeds that allowed for the address space of a process.
815 .sp
816 The \fBMAP_FIXED\fR option was \fInot\fR specified and there is insufficient
817 room in the address space to effect the mapping.
818 .sp
819 The mapping could not be locked in memory, if required by \fBmlockall\fR(3C),
820 because it would require more space than the system is able to supply.
821 .sp
822 The composite size of \fIlen\fR plus the lengths obtained from all previous
823 calls to \fBmmap()\fR exceeds \fBRLIMIT_VMEM\fR (see  \fBgetrlimit\fR(2)).
824 .RE

826 .sp
827 .ne 2
828 .na
829 \fB\fBENOTSUP\fR\fR
830 .ad
831 .RS 13n
832 The system does not support the combination of accesses requested in the
833 \fIprot\fR argument.
834 .RE

836 .sp
837 .ne 2
838 .na
839 \fB\fBENXIO\fR\fR
840 .ad
841 .RS 13n
842 Addresses in the range [\fIoff, off + len\fR) are invalid for the object
843 specified by \fIfildes\fR.
844 .sp
845 The \fBMAP_FIXED\fR option was specified in \fIflags\fR and the combination of
846 \fIaddr\fR, \fIlen\fR and \fIoff\fR is invalid for the object specified by
847 \fIfildes\fR.
848 .RE

850 .sp
851 .ne 2
852 .na
```



```
853 \fB\fBEOVERFLOW\fR\fR
854 .ad
855 .RS 13n
856 The file is a regular file and the value of \fIoff\fR plus \fIlen\fR exceeds
857 the offset maximum establish in the open file description associated with
858 \fIfildes\fR.
859 .RE

861 .sp
862 .LP
863 The \fBmmap()\fR function may fail if:
864 .sp
865 .ne 2
866 .na
867 \fB\fBEAGAIN\fR\fR
868 .ad
869 .RS 10n
870 The file to be mapped is already locked using advisory or mandatory record
871 locking. See \fBfcntl\fR(2).
872 .RE

874 .SH USAGE
875 .LP
876 Use of \fBmmap()\fR may reduce the amount of memory available to other memory
877 allocation functions.
878 .sp
879 .LP
880 \fBMAP_ALIGN\fR is useful to assure a properly aligned value of \fIpa\fR for
881 subsequent use with \fBmemcntl\fR(2) and the \fBMC_HAT_ADVISE\fR command. This
882 is best used for large, long-lived, and heavily referenced regions.
883 \fBMAP_FIXED\fR and \fBMAP_ALIGN\fR are always mutually-exclusive.
884 .sp
885 .LP
886 Use of \fBMAP_FIXED\fR may result in unspecified behavior in further use of
887 \fBbrk\fR(2), \fBsbrk\fR(2), \fBmalloc\fR(3C), and \fBshmat\fR(2). The use of
888 \fBMAP_FIXED\fR is discouraged, as it may prevent an implementation from making
889 the most effective use of resources.
890 .sp
891 .LP
892 The application must ensure correct synchronization when using \fBmmap()\fR in
893 conjunction with any other file access method, such as \fBread\fR(2) and
894 \fBwrite\fR(2), standard input/output, and \fBshmat\fR(2).
895 .sp
896 .LP
897 The \fBmmap()\fR function has a transitional interface for 64-bit file offsets.
898 See \fBlf64\fR(5).
899 .sp
900 .LP
901 The \fBmmap()\fR function allows access to resources using address space
902 manipulations instead of the \fBread()\fR/\fBwrite()\fR interface. Once a file
903 is mapped, all a process has to do to access it is use the data at the address
904 to which the object was mapped.
905 .sp
906 .LP
907 Consider the following pseudo-code:
908 .sp
909 .in +2
910 .nf
911 fildes = open(\|.\|.\|.)
912 lseek(fildes, offset, whence)
913 read(fildes, buf, len)
914 /* use data in buf */
915 .fi
916 .in -2

918 .sp
```

```
 919 .LP
 920 The following is a rewrite using  \fBmmap()\fR:
 921 .sp
 922 .in +2
 923 .nf
 924 fildes = open(\|.\|.\|.)
 925 address = mmap((caddr_t) 0, len, (PROT_READ | PROT_WRITE),
 926           MAP_PRIVATE, fildes, offset)
 927 /* use data at address */
 928 .fi
 929 .in -2

 931 .SH ATTRIBUTES
 932 .LP
 933 See \fBattributes\fR(5) for descriptions of the following attributes:
 934 .sp

 936 .sp
 937 .TS
 938 box;
 939 c | c
 940 l | l .
 941 ATTRIBUTE TYPE  ATTRIBUTE VALUE
 942 _
 943 Interface Stability     Standard
 944 _
 945 MT-Level        Async-Signal-Safe
 946 .TE

 948 .SH SEE ALSO
 949 .LP
 950 \fBclose\fR(2), \fBexec\fR(2), \fBfcntl\fR(2), \fBfork\fR(2),
 951 \fBgetrlimit\fR(2), \fBmemcntl\fR(2), \fBmmapobj\fR(2), \fBmprotect\fR(2),
 952 \fBmunmap\fR(2), \fBshmat\fR(2), \fBlockf\fR(3C), \fBmlockall\fR(3C),
 953 \fBmsync\fR(3C), \fBplock\fR(3C), \fBsysconf\fR(3C), \fBattributes\fR(5),
 954 \fBlf64\fR(5), \fBstandards\fR(5), \fBnull\fR(7D), \fBzero\fR(7D)
```

```
**********************************************************
    5703 Sun Sep 16 19:22:54 2018
new/usr/src/man/man2/pset_bind.2
9842 man page typos and spelling
**********************************************************
   1 '\" te
   2 .\" Copyright (c) 2009, Sun Microsystems, Inc.  All Rights Reserved.
   3 .\" The contents of this file are subject to the terms of the Common Development
   4 .\"  See the License for the specific language governing permissions and limitat
   5 .\" the fields enclosed by brackets "[]" replaced with your own identifying info
   6 .TH PSET_BIND 2 "Mar 13, 2009"
   7 .SH NAME
   8 pset_bind \- bind LWPs to a set of processors
   9 .SH SYNOPSIS
  10 .LP
  11 .nf
  12 #include <sys/pset.h>

  14 \fBint\fR \fBpset_bind\fR(\fBpsetid_t\fR \fIpset\fR, \fBidtype_t\fR \fIidtype\fR
  15 .fi

  17 .SH DESCRIPTION
  18 .sp
  18 .LP
  19 The \fBpset_bind()\fR function binds the \fBLWP\fR or set of \fBLWPs\fR
  20 specified by \fIidtype\fR and \fIid\fR to the processor set specified by
  21 \fIpset\fR. If \fIopset\fR is not \fINULL\fR, \fBpset_bind()\fR sets the
  22 \fBpsetid_t\fR variable pointed to by \fIopset\fR to the previous processor set
  23 binding of one of the specified \fBLWP\fR, or to \fBPS_NONE\fR if the selected
  24 \fBLWP\fR was not bound.
  25 .sp
  26 .LP
  27 If \fIidtype\fR is \fBP_PID\fR, the binding affects all \fBLWP\fRs of the
  28 process with process \fBID\fR (PID) \fIid\fR.
  29 .sp
  30 .LP
  31 If \fIidtype\fR is \fBP_LWPID\fR, the binding affects the \fBLWP\fR of the
  32 current process with \fBLWP ID\fR \fIid\fR.
  33 .sp
  34 .LP
  35 If \fIidtype\fR is \fBP_TASKID\fR, the binding affects all LWPs of all
  36 processes with task ID \fIid\fR.
  37 .sp
  38 .LP
  39 If \fIidtype\fR is \fBP_PROJID\fR, the binding affects all LWPs of all
  40 processes with project ID \fIid\fR.
  41 .sp
  42 .LP
  43 If \fIidtype\fR is \fBP_ZONEID\fR, the binding affects all LWPs of all
  44 processes with zone ID \fIid\fR.
  45 .sp
  46 .LP
  47 If \fIidtype\fR is \fBP_CTID\fR, the binding affects all LWPs of all processes
  48 with process contract ID \fIid\fR.
  49 .sp
  50 .LP
  51 If \fIid\fR is \fBP_MYID\fR, the specified LWP, process, task, process, zone,
  52 or process contract is the current one.
  53 .sp
  54 .LP
  55 If \fIpset\fR is \fBPS_NONE\fR, the processor set bindings of the specified
  56 LWPs are cleared.
  57 .sp
  58 .LP
  59 If \fIpset\fR is \fBPS_QUERY\fR, the processor set bindings are not changed.
  60 .sp
```



```
  61 .LP
  62 If \fIpset\fR is \fBPS_MYID\fR, the specified LWPs are bound to the same
  63 processor set as the caller. If the caller is not bound to a processor set, the
  64 processor set bindings are cleared.
  65 .sp
  66 .LP
  67 The {\fBPRIV_SYS_RES_CONFIG\fR} privilege must be asserted in the effective set
  68 of the calling process or \fIpset\fR must be \fBPS_QUERY\fR.
  69 .sp
  70 .LP
  71 LWPs that have been bound to a processor with \fBprocessor_bind\fR(2) may also
  72 be bound to a processor set if the processor is part of the processor set. If
  73 this occurs, the binding to the processor remains in effect. If the processor
  74 binding is later removed, the processor set binding becomes effective.
  75 .sp
  76 .LP
  77 Processor set bindings are inherited across \fBfork\fR(2) and \fBexec\fR(2).
  78 .SH RETURN VALUES
  80 .sp
  79 .LP
  80 Upon successful completion, 0 is returned. Otherwise, \(mi1 is returned and
  81 \fBerrno\fR is set to indicate the error.
  82 .SH ERRORS
  85 .sp
  83 .LP
  84 The \fBpset_bind()\fR function will fail if:
  85 .sp
  86 .ne 2
  87 .na
  88 \fB\fBEBUSY\fR\fR
  89 .ad
  90 .RS 11n
  91 One of the \fBLWP\fRs is bound to a processor, and the specified processor set
  92 does not include that processor.
  93 .RE

  95 .sp
  96 .ne 2
  97 .na
  98 \fB\fBEFAULT\fR\fR
  99 .ad
 100 .RS 11n
 101 The location pointed to by \fIopset\fR was not \fINULL\fR and not writable by
 102 the user.
 103 .RE

 105 .sp
 106 .ne 2
 107 .na
 108 \fB\fBEINVAL\fR\fR
 109 .ad
 110 .RS 11n
 111 An invalid processor set \fBID\fR was specified;  or \fIidtype\fR was not
 112 \fBP_PID\fR, \fBP_LWPID\fR, \fBP_PROJID\fR, \fBP_TASKID\fR, \fBP_ZONEID\fR, or
 113 \fBP_CTID\fR.
 114 .RE

 116 .sp
 117 .ne 2
 118 .na
 119 \fB\fBENOTSUP\fR\fR
 120 .ad
 121 .RS 11n
 122 The pools facility is active.  See \fBpooladm\fR(1M) and
 123 \fBpool_set_status\fR(3POOL) for information about enabling and disabling the
 124 pools facility. Processes can be bound to pools using the \fBpoolbind\fR(1M)
```

```
125 utility or the \fBpool_set_binding\fR(3POOL) function.
126 .sp
127 Binding a system process to a processor set is not supported.
128 .RE

130 .sp
131 .ne 2
132 .na
133 \fB\fBEPERM\fR\fR
134 .ad
135 .RS 11n
136 The {\fBPRIV_PROC_OWNER\fR} is not asserted in the effective set of the calling
139 The {\fBPRIV_PROC_OWNER\fR} is not asserted in the effecive set of the calling
137 process and either the real or effective user ID of the calling process does
138 not match the real or effective user \fBID\fR of one of the LWPs being bound,
139 or the processor set from which one or more of the LWPs are being unbound has
140 the \fBPSET_NOESCAPE\fR attribute set and {\fBPRIV_SYS_RES_CONFIG\fR) is not
141 asserted in the effective set of the calling process. See \fBpset_setattr\fR(2)
142 for more information about processor set attributes.
143 .RE

145 .sp
146 .ne 2
147 .na
148 \fB\fBESRCH\fR\fR
149 .ad
150 .RS 11n
151 No processes, \fBLWP\fRs, or tasks were found to match the criteria specified
152 by \fIidtype\fR and \fIid\fR.
153 .RE

155 .SH ATTRIBUTES
159 .sp
156 .LP
157 See \fBattributes\fR(5) for descriptions of the following attributes:
158 .sp

160 .sp
161 .TS
162 box;
163 c | c
164 l | l .
165 ATTRIBUTE TYPE  ATTRIBUTE VALUE
166 _
167 Interface Stability     Committed
168 _
169 MT-Level        Async-Signal-Safe
170 .TE

172 .SH SEE ALSO
177 .sp
173 .LP
174 \fBpbind\fR(1M), \fBpooladm\fR(1M), \fBpoolbind\fR(1M), \fBpsrset\fR(1M),
175 \fBexec\fR(2), \fBfork\fR(2), \fBprocessor_bind\fR(2), \fBpset_create\fR(2),
176 \fBpset_info\fR(2), \fBpset_setattr\fR(2), \fBpool_set_binding\fR(3POOL),
177 \fBpool_set_status\fR(3POOL), \fBpset_getloadavg\fR(3C), \fBprocess\fR(4),
178 \fBproject\fR(4), \fBattributes\fR(5), \fBprivileges\fR(5)
```

```
**********************************************************
    3012 Sun Sep 16 19:22:54 2018
new/usr/src/man/man3c/crypt_genhash_impl.3c
9842 man page typos and spelling
**********************************************************
'\" te
.\" Copyright (c) 2002, Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH CRYPT_GENHASH_IMPL 3C "Jun 10, 2002"
.SH NAME
crypt_genhash_impl \- generate encrypted password
.SH SYNOPSIS
.LP
.nf
#include <crypt.h>

\fBchar *\fR\fBcrypt_genhash_impl\fR(\fBchar *\fR\fIctbuffer\fR, \fBsize_t\fR \f
     \fBconst char *\fR\fIplaintext\fR, \fBconst char *\fR\fIsalt\fR, \fBconst c
.fi

.SH DESCRIPTION
.sp
.LP
The \fBcrypt_genhash_impl()\fR function is called by \fBcrypt\fR(3C) to
generate the encrypted password \fIplaintext\fR.
.sp
.LP
The \fIctbuffer\fR argument is a pointer to an MT-safe buffer of
\fIctbufflen\fR size that is used to return the result.
.sp
.LP
The \fIsalt\fR argument is the salt used in encoding.
.sp
.LP
The \fIparams\fR argument is an \fIargv\fR-like null-terminated vector of type
\fBchar *\fR. The first element of \fIparams\fR represents the mechanism token
name from \fBcrypt.conf\fR(4). The remaining elements of \fIparams\fR represent
strings of the form <\fBparameter\fR>[=<\fBvalue\fR>] to allow passing in
additional information from the \fBcrypt.conf\fR entry, such as specifying
*additional information from the \fBcrypt.conf\fR entry, such as specifing*
rounds information "\fBrounds=4096\fR".
.sp
.LP
The \fBcrypt_genhash_impl()\fR function must not \fBfree\fR(3C) \fIctbufflen\fR
on error.
.SH RETURN VALUES
.sp
.LP
Upon successful completion, \fBcrypt_genhash_impl()\fR returns a pointer to the
encoded version of \fIplaintext\fR. Otherwise a null pointer is returned and
\fBerrno\fR is set to indicate the error.
.SH ERRORS
.sp
.LP
The \fBcrypt_genhash_impl()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 11n
The configuration file \fBcrypt.conf\fR contains an invalid entry.
.RE
```



```
.sp
.ne 2
.na
\fB\fBELIBACC\fR\fR
.ad
.RS 11n
The required shared library was not found.
.RE

.sp
.ne 2
.na
\fB\fBENOMEM\fR\fR
.ad
.RS 11n
There is insufficient memory to perform hashing.
.RE

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
Interface Stability     Evolving
_
MT-Level        MT-Safe
.TE

.SH SEE ALSO
.sp
.LP
\fBpasswd\fR(1), \fBcrypt\fR(3C), \fBcrypt_gensalt_impl\fR(3C), \fBfree\fR(3C),
\fBgetpassphrase\fR(3C), \fBcrypt.conf\fR(4), \fBpasswd\fR(4),
\fBattributes\fR(5)
```

```
**********************************************************
    5835 Sun Sep 16 19:22:54 2018
new/usr/src/man/man3c/sem_timedwait.3c
9842 man page typos and spelling
**********************************************************
```

```
.\"
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for
.\" permission to reproduce portions of its copyrighted documentation.
.\" Original documentation from The Open Group can be obtained online at
.\" http://www.opengroup.org/bookstore/.
.\"
.\" The Institute of Electrical and Electronics Engineers and The Open
.\" Group, have given us permission to reprint portions of their
.\" documentation.
.\"
.\" In the following statement, the phrase ‘‘this text’’ refers to portions
.\" of the system documentation.
.\"
.\" Portions of this text are reprinted and reproduced in electronic form
.\" in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition,
.\" Standard for Information Technology -- Portable Operating System
.\" Interface (POSIX), The Open Group Base Specifications Issue 6,
.\" Copyright (C) 2001-2004 by the Institute of Electrical and Electronics
.\" Engineers, Inc and The Open Group.  In the event of any discrepancy
.\" between these versions and the original IEEE and The Open Group
.\" Standard, the original IEEE and The Open Group Standard is the referee
.\" document.  The original Standard can be obtained online at
.\" http://www.opengroup.org/unix/online.html.
.\"
.\" This notice shall appear on any product containing this material.
.\"
.\" The contents of this file are subject to the terms of the
.\" Common Development and Distribution License (the "License").
.\" You may not use this file except in compliance with the License.
.\"
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
.\" or http://www.opensolaris.org/os/licensing.
.\" See the License for the specific language governing permissions
.\" and limitations under the License.
.\"
.\" When distributing Covered Code, include this CDDL HEADER in each
.\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
.\" If applicable, add the following below this CDDL HEADER, with the
.\" fields enclosed by brackets "[]" replaced with your own identifying
.\" information: Portions Copyright [yyyy] [name of copyright owner]
.\"
.\"
.\" Copyright (c) 2001, The IEEE and The Open Group.  All Rights Reserved.
.\" Portions Copyright (c) 2008, Sun Microsystems, Inc. All Rights Reserved.
.\"
.TH SEM_TIMEDWAIT 3C "Feb 5, 2008"
.SH NAME
sem_timedwait, sem_reltimedwait_np \- lock a semaphore
.SH SYNOPSIS
.LP
.nf
#include <semaphore.h>
#include <time.h>

\fBint\fR \fBsem_timedwait\fR(\fBsem_t *restrict\fR \fIsem\fR,
     \fBconst struct timespec *restrict\fR \fIabs_timeout\fR);
.fi

.LP
.nf
\fBint\fR \fBsem_reltimedwait_np\fR(\fBsem_t *restrict\fR \fIsem\fR,
```



```
     \fBconst struct timespec *restrict\fR \fIrel_timeout\fR);
.fi

.SH DESCRIPTION
.sp
.LP
The \fBsem_timedwait()\fR function locks the semaphore referenced by \fIsem\fR
as in the \fBsem_wait\fR(3C) function. However, if the semaphore cannot be
locked without waiting for another process or thread to unlock the semaphore by
performing a \fBsem_post\fR(3C) function, this wait is terminated when the
specified timeout expires.
.sp
.LP
The \fBsem_reltimedwait_np()\fR function is identical to the
\fBsem_timedwait()\fR function, except that the timeout is specified as a
relative time interval.
.sp
.LP
For \fBsem_timedwait()\fR, the timeout expires when the absolute time specified
by \fIabs_timeout\fR passes, as measured by the \fBCLOCK_REALTIME\fR clock
(that is, when the value of that clock equals or exceeds \fIabs_timeout\fR), or
if the absolute time specified by \fIabs_timeout\fR has already been passed at
the time of the call.
.sp
.LP
For \fBsem_reltimedwait_np()\fR, the timeout expires when the time interval
specified by \fIrel_timeout\fR passes, as measured by the \fBCLOCK_REALTIME\fR
clock, or if the time interval specified by \fIrel_timeout\fR is negative at
the time of the call.
.sp
.LP
The resolution of the timeout is the resolution of the \fBCLOCK_REALTIME\fR
clock. The \fBtimespec\fR data type is defined as a structure in the
<\fBtime.h\fR> header.
.sp
.LP
Under no circumstance does the function fail with a timeout if the semaphore
can be locked immediately. The validity of the \fIabs_timeout\fR need not be
checked if the semaphore can be locked immediately.
.SH RETURN VALUES
.sp
.LP
The \fBsem_timedwait()\fR and \fBsem_reltimedwait_np()\fR functions return 0 if
the calling process successfully performed the semaphore lock operation on the
semaphore designated by \fIsem\fR. If the call was unsuccessful, the state of
the semaphore is be unchanged and the function returns -1 and sets \fBerrno\fR
to indicate the error.
.SH ERRORS
.sp
.LP
The \fBsem_timedwait()\fR and \fBsem_reltimedwait_np()\fR functions will fail
if:
.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 13n
The \fIsem\fR argument does not refer to a valid semaphore.
.RE

.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
```

```
.RS 13n
The process or thread would have blocked, and the timeout parameter specified a
nanoseconds field value less than zero or greater than or equal to 1,000
million.
.RE

.sp
.ne 2
.na
\fB\fBETIMEDOUT\fR\fR
.ad
.RS 13n
The semaphore could not be locked before the specified timeout expired.
.RE

.sp
.LP
The \fBsem_timedwait()\fR and \fBsem_reltimedwait_np()\fR functions may fail
if:
.sp
.ne 2
.na
\fB\fBEDEADLK\fR\fR
.ad
.RS 11n
A deadlock condition was detected.
.RE

.sp
.ne 2
.na
\fB\fBEINTR\fR\fR
.ad
.RS 11n
A signal interrupted this function.
.RE

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
Interface Stability     Committed
Interface Stability     Commmitted
_
MT-Level        MT-Safe
_
Standard        See below.
.TE

.sp
.LP
For \fBsem_timedwait()\fR, see \fBstandards\fR(5).
.SH SEE ALSO
.sp
.LP
\fBsemctl\fR(2), \fBsemget\fR(2), \fBsemop\fR(2), \fBtime\fR(2),
\fBsem_post\fR(3C), \fBsem_trywait\fR(3C), \fBsem_wait\fR(3C),
```



```
\fBattributes\fR(5), \fBstandards\fR(5)
```

```
**********************************************************
   24315 Sun Sep 16 19:22:54 2018
new/usr/src/man/man3c/string.3c
9842 man page typos and spelling
**********************************************************
```

```
.\"
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for
.\" permission to reproduce portions of its copyrighted documentation.
.\" Original documentation from The Open Group can be obtained online at
.\" http://www.opengroup.org/bookstore/.
.\"
.\" The Institute of Electrical and Electronics Engineers and The Open
.\" Group, have given us permission to reprint portions of their
.\" documentation.
.\"
.\" In the following statement, the phrase ‘‘this text’’ refers to portions
.\" of the system documentation.
.\"
.\" Portions of this text are reprinted and reproduced in electronic form
.\" in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition,
.\" Standard for Information Technology -- Portable Operating System
.\" Interface (POSIX), The Open Group Base Specifications Issue 6,
.\" Copyright (C) 2001-2004 by the Institute of Electrical and Electronics
.\" Engineers, Inc and The Open Group.  In the event of any discrepancy
.\" between these versions and the original IEEE and The Open Group
.\" Standard, the original IEEE and The Open Group Standard is the referee
.\" document.  The original Standard can be obtained online at
.\" http://www.opengroup.org/unix/online.html.
.\"
.\" This notice shall appear on any product containing this material.
.\"
.\" The contents of this file are subject to the terms of the
.\" Common Development and Distribution License (the "License").
.\" You may not use this file except in compliance with the License.
.\"
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
.\" or http://www.opensolaris.org/os/licensing.
.\" See the License for the specific language governing permissions
.\" and limitations under the License.
.\"
.\" When distributing Covered Code, include this CDDL HEADER in each
.\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
.\" If applicable, add the following below this CDDL HEADER, with the
.\" fields enclosed by brackets "[]" replaced with your own identifying
.\" information: Portions Copyright [yyyy] [name of copyright owner]
.\"
.\"
.\" Copyright 1989 AT&T
.\" Portions Copyright (c) 1992, X/Open Company Limited.  All Rights Reserved.
.\" Portions Copyright (c) 1994 Man-cgi 1.15, Panagiotis Christias (christia@sof
.\" Portions Copyright (c) 1996-2008 Modified for NetBSD by Kimmo Suominen (kimm
.\" Copyright (c) 2008, Sun Microsystems, Inc.  All Rights Reserved.
.\" Copyright 2014 Garrett D’Amore <garrett@damore.org>
.\" Copyright (c) 2014, Joyent, Inc.
.\"
.TH STRING 3C "Mar 23, 2016"
.SH NAME
string, strcasecmp, strcasecmp_l, strncasecmp, strncasecmp_l, strcat, strncat,
strlcat, strchr, strchrnul, strrchr, strcmp, strncmp, stpcpy, stpncpy, strcpy,
strncpy, strlcpy, strcspn, strspn, strdup, strndup, strdupa, strndupa, strlen,
strnlen, strpbrk, strsep, strstr, strnstr, strcasestr, strtok, strtok_r \-
string operations
.SH SYNOPSIS
.LP
.nf
#include <strings.h>
```



```
\fBint\fR \fBstrcasecmp\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fIs2\f
.fi
.LP
.nf
\fBint\fR \fBstrcasecmp_l\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fIs2
.fi
.LP
.nf
\fBint\fR \fBstrncasecmp\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fIs2\
.fi
.LP
.nf
\fBint\fR \fBstrncasecmp_l\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fIs
.fi
.LP
.nf
#include <string.h>

\fBchar *\fR\fBstrcat\fR(\fBchar *restrict\fR \fIs1\fR, \fBconst char *restrict\
.fi
.LP
.nf
\fBchar *\fR\fBstrncat\fR(\fBchar *restrict\fR \fIs1\fR, \fBconst char *restrict
.fi
.LP
.nf
\fBsize_t\fR \fBstrlcat\fR(\fBchar *\fR\fIdst\fR, \fBconst char *\fR\fIsrc\fR, \
.fi
.LP
.nf
\fBchar *\fR\fBstrchr\fR(\fBconst char *\fR\fIs\fR, \fBint\fR \fIc\fR);
.fi
.LP
.nf
\fBchar *\fR\fBstrrchr\fR(\fBconst char *\fR\fIs\fR, \fBint\fR \fIc\fR);
.fi
.LP
.nf
\fBint\fR \fBstrcmp\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fIs2\fR);
.fi
.LP
.nf
\fBint\fR \fBstrncmp\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fIs2\fR,
.fi
.LP
.nf
\fBchar *\fR\fBstpcpy\fR(\fBchar *restrict\fR \fIs1\fR, \fBconst char *restrict\
.fi
.LP
.nf
\fBchar *\fR\fBstpncpy\fR(\fBchar *restrict\fR \fIs1\fR, \fBconst char *restrict
.fi
.LP
.nf
\fBchar *\fR\fBstrcpy\fR(\fBchar *restrict\fR \fIs1\fR, \fBconst char *restrict\
.fi
.LP
.nf
\fBchar *\fR\fBstrncpy\fR(\fBchar *restrict\fR \fIs1\fR, \fBconst char *restrict
.fi
.LP
.nf
\fBsize_t\fR \fBstrlcpy\fR(\fBchar *\fR\fIdst\fR, \fBconst char *\fR\fIsrc\fR, \
.fi
.LP
```

```
.nf
\fBsize_t\fR \fBstrcspn\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fIs2\f
.fi
.LP
.nf
\fBsize_t\fR \fBstrspn\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fIs2\fR
.fi
.LP
.nf
\fBchar *\fR\fBstrdup\fR(\fBconst char *\fR\fIs1\fR);
.fi
.LP
.nf
\fBchar *\fR\fBstrndup\fR(\fBconst char *\fR\fIs1\fR, \fBsize_t\fR \fIn\fR);
.fi
.LP
.nf
\fBchar *\fR\fBstrdupa\fR(\fBconst char *\fR\fIs1\fR);
.fi
.LP
.nf
\fBchar *\fR\fBstrndupa\fR(\fBconst char *\fR\fIs1\fR, \fBsize_t\fR \fIn\fR);
.fi
.LP
.nf
\fBsize_t\fR \fBstrlen\fR(\fBconst char *\fR\fIs\fR);
.fi
.LP
.nf
\fBsize_t\fR \fBstrnlen\fR(\fBconst char *\fR\fIs\fR, \fBsize_t\fR \fIn\fR);
.fi
.LP
.nf
\fBchar *\fR\fBstrpbrk\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fIs2\fR
.fi
.LP
.nf
\fBchar *\fR\fBstrsep\fR(\fBchar **\fR\fIstringp\fR, \fBconst char *\fR\fIdelim\
.fi
.LP
.nf
\fBchar *\fR\fBstrstr\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fIs2\fR)
.fi
.LP
.nf
\fBchar *\fR\fBstrnstr\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fIs2\fR
.fi
.LP
.nf
\fBchar *\fR\fBstrcasestr\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fIs2
.fi
.LP
.nf
\fBchar *\fR\fBstrcasestr_l\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fI
.fi
.LP
.nf
\fBchar *\fR\fBstrtok\fR(\fBchar *restrict\fR \fIs1\fR, \fBconst char *restrict\
.fi
.LP
.nf
\fBchar *\fR\fBstrtok_r\fR(\fBchar *restrict\fR \fIs1\fR, \fBconst char *restric
     \fBchar **restrict\fR \fIlasts\fR);
.fi
.SS "ISO C++"
.LP
```



```
.nf
#include <string.h>

\fBconst char *\fR\fBstrchr\fR(\fBconst char *\fR\fIs\fR, \fBint\fR \fIc\fR);
.fi
.LP
.nf
\fBconst char *\fR\fBstrchrnul\fR(\fBconst char *\fR\fIs\fR, \fBint\fR \fIc\fR);
.fi
.LP
.nf
\fBconst char *\fR\fBstrpbrk\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\f
.fi
.LP
.nf
\fBconst char *\fR\fBstrrchr\fR(\fBconst char *\fR\fIs\fR, \fBint\fR \fIc\fR);
.fi
.LP
.nf
\fBconst char *\fR\fBstrstr\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fI
.fi
.LP
.nf
#include <cstring>

\fBchar *std::\fR\fBstrchr\fR(\fBchar *\fR\fIs\fR, \fBint\fR \fIc\fR);
.fi
.LP
.nf
\fBchar *std::\fR\fBstrpbrk\fR(\fBchar *\fR\fIs1\fR, \fBconst char *\fR\fIs2\fR)
.fi
.LP
.nf
\fBchar *std::\fR\fBstrrchr\fR(\fBchar *\fR\fIs\fR, \fBint\fR \fIc\fR);
.fi
.LP
.nf
\fBchar *std::\fR\fBstrstr\fR(\fBchar *\fR\fIs1\fR, \fBconst char *\fR\fIs2\fR);
.fi
.SH DESCRIPTION
.LP
The arguments \fIs\fR, \fIs1\fR, and \fIs2\fR point to strings (arrays of
characters terminated by a null character). The \fBstrcat()\fR, \fBstrncat()\fR,
\fBstrlcat()\fR, \fBstrcpy()\fR, \fBstpcpy()\fR, \fBstpncpy()\fR,
\fBstrncpy()\fR, \fBstrlcpy()\fR, \fBstrsep()\fR, \fBstrtok()\fR, and
\fBstrtok_r()\fR functions all alter their first argument. Additionally, the
\fBstrcat()\fR, \fBstpcpy()\fR, and \fBstrcpy()\fR functions do not check for
overflow of the array.
.SS "\fBstrcasecmp()\fR, \fBstrncasecmp()\fR"
.LP
The \fBstrcasecmp()\fR and \fBstrncasecmp()\fR functions are case-insensitive
versions of  \fBstrcmp()\fR and \fBstrncmp()\fR respectively, described below.
.LP
The \fBstrcasecmp()\fR and \fBstrncasecmp()\fR functions compare two strings
byte-by-byte, after
converting each upper-case character to lower-case (as determined by the
\fBLC_CTYPE\fR category of the current locale).  Note that neither the contents
pointed to by \fIs1\fR nor \fIs2\fR are modified.
.LP
The functions return an integer
greater than, equal to, or less than 0, if the string pointed to by \fIs1\fR
is greater than, equal to, or less than the string pointed to by \fIs2\fR
respectively. The sign of a non-zero return value is determined  by the sign of
the difference between the values of the first pair of bytes that differ in the
.LP
The \fBstrncasecmp()\fR function examines at most \fIn\fR bytes from each
```

string.
.SS "\fBstrcasecmp_l()\fR, \fBstrncasecmp_l()\fR"
.LP
The \fBstrcasecmp_l()\fR and \fBstrncasecmp_l()\fR functions behave identically to \fBstrcasecmp()\fR and \fBstrncasecmp()\fR, except instead of operating in the current locale, they instead operate in the locale specified by \fIloc\fR.
.SS "\fBstrcat()\fR, \fBstrncat()\fR, \fBstrlcat()\fR"
.LP
The \fBstrcat()\fR function appends a copy of string \fIs2\fR, including the terminating null character, to the end of string \fIs1\fR. The \fBstrncat()\fR function appends at most \fIn\fR characters of \fIs2\fR to \fIs1\fR, not including any terminating null character, and then appends a null character. Each returns a pointer to the null-terminated result. The initial character of \fIs2\fR overrides the null character at the end of \fIs1\fR. If copying takes place between objects that overlap, the behavior of \fBstrcat()\fR, \fBstrncat()\fR, and \fBstrlcat()\fR is undefined.
.LP
The \fBstrlcat()\fR function appends at most (\fIdstsize\fR-\fBstrlen\fR(\fIdst\fR)-1) characters of \fIsrc\fR to \fIdst\fR (\fIdstsize\fR being the size of the string buffer \fIdst\fR). If the string pointed to by \fIdst\fR contains a null-terminated string that fits into \fIdstsize\fR bytes when \fBstrlcat()\fR is called, the string pointed to by \fIdst\fR will be a null-terminated string that fits in \fIdstsize\fR bytes (including the terminating null character) when it completes, and the initial character of \fIsrc\fR will override the null character at the end of \fIdst\fR. If the string pointed to by \fIdst\fR is longer than \fIdstsize\fR bytes when \fBstrlcat()\fR is called, the string pointed to by \fIdst\fR will not be changed. The function returns \fBmin\fR{\fIdstsize\fR,\fBstrlen\fR(\fIdst\fR)}+\fBstrlen\fR(\fIsrc\fR). Buffer overflow can be checked as follows:
.sp
.in +2
.nf
if (strlcat(dst, src, dstsize) >= dstsize)
        return \(mi1;
.fi
.in -2
.SS "\fBstrchr()\fR, \fBstrrchr()\fR, \fBstrchrnul()\fR"
.LP
The \fBstrchr()\fR function returns a pointer to the first occurrence of \fIc\fR (converted to a \fBchar\fR) in string \fIs\fR, or a null pointer if \fIc\fR does not occur in the string. The \fBstrrchr()\fR function returns a pointer to the last occurrence of \fIc\fR. The null character terminating a string is considered to be part of the string. The \fBstrchrnul()\fR function behaves similarly to \fBstrchr()\fR, except when the character \fBc\fR is not found, it returns a pointer to the null terminator of the string \fBs\fR and not a null pointer.
.SS "\fBstrcmp()\fR, \fBstrncmp()\fR"
.LP
The \fBstrcmp()\fR function compares two strings byte-by-byte, according to the ordering of your machine’s character set. The function returns an integer greater than, equal to, or less than 0, if the string pointed to by \fIs1\fR is greater than, equal to, or less than the string pointed to by \fIs2\fR respectively. The sign of a non-zero return value is determined by the sign of the difference between the values of the first pair of bytes that differ in the strings being compared. The \fBstrncmp()\fR function makes the same comparison but looks at a maximum of \fIn\fR bytes. Bytes following a null byte are not compared.
.SS "\fBstrcpy()\fR, \fBstrncpy()\fR, \fBstrlcpy()\fR"
.LP
The \fBstrcpy()\fR function copies string \fIs2\fR to \fIs1\fR, including the terminating null character, stopping after the null character has been copied. The \fBstrncpy()\fR function copies exactly \fIn\fR bytes, truncating \fIs2\fR or adding null characters to \fIs1\fR if necessary. The result will not be null-terminated if the length of \fIs2\fR is \fIn\fR or more. Both the \fBstrcpy()\fR and \fBstrncpy()\fR functions return \fIs1\fR. If copying takes

place between objects that overlap, the behavior of \fBstrcpy()\fR, \fBstrncpy()\fR, and \fBstrlcpy()\fR is undefined.
.LP
The \fBstrlcpy()\fR function copies at most \fIdstsize\fR\(mi1 characters (\fIdstsize\fR being the size of the string buffer \fIdst\fR) from \fIsrc\fR to \fIdst\fR, truncating \fIsrc\fR if necessary. The result is always null-terminated. The function returns \fBstrlen\fR(\fIsrc\fR). Buffer overflow can be checked as follows:
.sp
.in +2
.nf
if (strlcpy(dst, src, dstsize) >= dstsize)
        return \(mi1;
.fi
.in -2
.SS "\fBstpcpy()\fR, \fBstpncpy()\fR"
.LP
The \fBstpcpy()\fR and \fBstpncpy()\fR functions behave identically to \fBstrcpy()\fR and \fBstrncpy()\fR respectively; however, instead of returning a pointer to the beginning of \fIs1\fR, they return a pointer to the terminating null character.
.SS "\fBstrcspn()\fR, \fBstrspn()\fR"
.LP
The \fBstrcspn()\fR function returns the length of the initial segment of string \fIs1\fR that consists entirely of characters not from string \fIs2\fR. The \fBstrspn()\fR function returns the length of the initial segment of string \fIs1\fR that consists entirely of characters from string \fIs2\fR.
.SS "\fBstrdup()\fR, \fBstrndup()\fR, \fBstrdupa()\fR, \fBstrndupa()\fR"
.LP
The \fBstrdup()\fR function returns a pointer to a new string that is a duplicate of the string pointed to by \fIs1\fR. The returned pointer can be passed to \fBfree()\fR. The space for the new string is obtained using \fBmalloc\fR(3C). If the new string cannot be created, a null pointer is returned and \fBerrno\fR may be set to \fBENOMEM\fR to indicate that the storage space available is insufficient. The \fBstrndup()\fR function is **identical to \fBstrdup()\fR, except it copies at most \fIn\fR bytes from \fBs1\fR and ensures the copied string is always null terminated.**
identical to \fBstrdup()\fR, execept it copies at most \fIn\fR bytes from \fBs1\fR and ensures the copied string is awlays null terminated.
.LP
The functions \fBstrdupa()\fR and \fBstrndupa()\fR behave identically to \fBstrdup()\fR and \fBstrndup()\fR respectively; however, instead of allocating memory using \fBmalloc\fR(3C), they use \fBalloca\fR(3C). These functions are provided for compatibility only, their use is strongly discouraged due to their use of \fBalloca\fR(3C).
.SS "\fBstrlen()\fR, \fBstrnlen()\fR"
.LP
The \fBstrlen()\fR function returns the number of bytes in \fIs\fR, not including the terminating null character.
.LP
The \fBstrnlen()\fR function returns the smaller of \fIn\fR or the number of bytes in \fIs\fR, not including the terminating null character. The \fBstrnlen()\fR function never examines more than \fIn\fR bytes of the string pointed to by \fIs\fR.
.SS "\fBstrpbrk()\fR"
.LP
The \fBstrpbrk()\fR function returns a pointer to the first occurrence in string \fIs1\fR of any character from string \fIs2\fR, or a null pointer if no character from \fIs2\fR exists in \fIs1\fR.
.SS "\fBstrsep()\fR"
.LP
The \fBstrsep()\fR function locates, in the null-terminated string referenced by *\fIstringp\fR, the first occurrence of any character in the string \fIdelim\fR (or the terminating ‘\e0’ character) and replaces it with a ‘\e0’. The location of the next character after the delimiter character (or \fINULL\fR, if the end of the string was reached) is stored in *\fIstringp\fR. The original value of *\fIstringp\fR is returned.

```
.LP
An ‘‘empty’’ field (one caused by two adjacent delimiter characters) can be
detected by comparing the location referenced by the pointer returned by
\fBstrsep()\fR to ‘\e0’.
.LP
If *\fIstringp\fR is initially \fINULL\fR, \fBstrsep()\fR returns \fINULL\fR.
.SS "\fBstrstr()\fR, \fBstrnstr()\fR, \fBstrcasestr()\fR, \fBstrcasestr_l()\fR"
.LP
The \fBstrstr()\fR function locates the first occurrence of the string \fIs2\fR
(excluding the terminating null character) in string \fIs1\fR and returns a
pointer to the located string, or a null pointer if the string is not found. If
\fIs2\fR points to a string with zero length (that is, the string \fB""\fR),
the function returns  \fIs1\fR. The \fBstrnstr()\fR function performs the same
search as \fBstrstr()\fR, but only considers up to \fIn\fR bytes of \fIs1\fR.
Bytes following a null byte are not compared.
.sp
.LP
The \fBstrcasestr()\fR and \fBstrcasestr_l()\fR functions are similar to
\fBstrstr()\fR, but both functions ignore the case of both \fBs1\fR and
\fBs2\fR. Where as the \fBstrcasestr()\fR function operates in the current
locale, the \fBstrcasestr_l()\fR function operates in the locale specified by
\fIloc\fR.
.SS "\fBstrtok()\fR"
.LP
A sequence of calls to \fBstrtok()\fR breaks the string pointed to by \fIs1\fR
into a sequence of tokens, each of which is delimited by a byte from the string
pointed to by \fIs2\fR. The first call in the sequence has \fIs1\fR as its
first argument, and is followed by calls with a null pointer as their first
argument. The separator string pointed to by \fIs2\fR can be different from
call to call.
.LP
The first call in the sequence searches the string pointed to by \fIs1\fR for
the first byte that is not contained in the current separator string pointed to
by \fIs2\fR. If no such byte is found, then there are no tokens in the string
pointed to by \fIs1\fR and \fBstrtok()\fR returns a null pointer. If such a
byte is found, it is the start of the first token.
.LP
The \fBstrtok()\fR function then searches from there for a byte that is
contained in the current separator string. If no such byte is found, the
current token extends to the end of the string pointed to by \fIs1\fR, and
subsequent searches for a token return a null pointer. If such a byte is found,
it is overwritten by a null byte that terminates the current token. The
\fBstrtok()\fR function saves a pointer to the following byte in
thread-specific data, from which the next search for a token starts.
.LP
Each subsequent call, with a null pointer as the value of the first argument,
starts searching from the saved pointer and behaves as described above.
.LP
See Example 1, 2, and 3 in the \fBEXAMPLES\fR section for examples of
\fBstrtok()\fR usage and the explanation in \fBNOTES\fR.
.SS "\fBstrtok_r()\fR"
.LP
The \fBstrtok_r()\fR function considers the null-terminated string \fIs1\fR as
a sequence of zero or more text tokens separated by spans of one or more
characters from the separator string \fIs2\fR. The argument \fIlasts\fR points
to a user-provided pointer which points to stored information necessary for
\fBstrtok_r()\fR to continue scanning the same string.
.LP
In the first call to \fBstrtok_r()\fR, \fIs1\fR points to a null-terminated
string, \fIs2\fR to a null-terminated string of separator characters, and the
value pointed to by \fIlasts\fR is ignored. The \fBstrtok_r()\fR function
returns a pointer to the first character of the first token, writes a null
character into \fIs1\fR immediately following the returned token, and updates
the pointer to which \fIlasts\fR points.
.LP
In subsequent calls, \fIs1\fR is a null pointer and \fIlasts\fR is unchanged
```



```
from the previous call so that subsequent calls move through the string
\fIs1\fR, returning successive tokens until no tokens remain. The separator
string \fIs2\fR can be different from call to call. When no token remains in
\fIs1\fR, a null pointer is returned.
.LP
See Example 3 in the \fBEXAMPLES\fR section for an example of \fBstrtok_r()\fR
usage and the explanation in \fBNOTES\fR.
.SH EXAMPLES
.LP
\fBExample 1 \fRSearch for word separators.
.LP
The following example searches for tokens separated by space characters.

.sp
.in +2
.nf
#include <string.h>
\&...
char *token;
char line[] = "LINE TO BE SEPARATED";
char *search = " ";

/* Token will point to "LINE". */
token = strtok(line, search);

/* Token will point to "TO". */
token = strtok(NULL, search);
.fi
.in -2

.LP
\fBExample 2 \fRBreak a Line.
.LP
The following example uses strtok to break a line into two character strings
separated by any combination of SPACEs, TABs, or NEWLINEs.

.sp
.in +2
.nf
#include <string.h>
\&...
struct element {
        char *key;
        char *data;
};
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    7159 Sun Sep 16 19:22:55 2018
new/usr/src/man/man3dat/dat_ep_post_recv.3dat
9842 man page typos and spelling
**********************************************************
'\" te
.\" This manual page is derived from the DAT/uDAPL 1.2 specification.
.\" Portions Copyright (c) 2007, Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH DAT_EP_POST_RECV 3DAT "Jul 16, 2004"
.SH NAME
dat_ep_post_recv \- receive data over the connection of the Endpoint
.SH SYNOPSIS
.LP
.nf
cc [ \fIflag\fR\&.\|.\|. ] \fIfile\fR\&.\|.\|. \fB-ldat\fR [ \fIlibrary\fR\&.\|.
#include <\fBdat/udat.h\fR>

DAT_RETURN
    dat_ep_post_recv (
    IN    DAT_EP_HANDLE        ep_handle,
    IN    DAT_COUNT            num_segments,
    IN    DAT_LMR_TRIPLET      *local_iov,
    IN    DAT_DTO_COOKIE       user_cookie,
    IN    DAT_COMPLETION_FLAGS completion_flags
    )
.fi

.SH PARAMETERS
.sp
.ne 2
.na
\fB\fIep_handle\fR\fR
.ad
.RS 20n
Handle for an instance of the Endpoint.
.RE

.sp
.ne 2
.na
\fB\fInum_segments\fR\fR
.ad
.RS 20n
Number of \fIlmr_triplets\fR in \fIlocal_iov\fR. Can be 0 for receiving a 0
size message.
.RE

.sp
.ne 2
.na
\fB\fIlocal_iov\fR\fR
.ad
.RS 20n
I/O Vector that specifies the local buffer to be filled. Can be \fINULL\fR for
receiving a 0 size message.
.RE

.sp
.ne 2
.na
\fB\fIuser_cookie:\fR\fR
.ad
.RS 20n
```



```
User-provided cookie that is returned to the Consumer at the completion of the
Receive DTO. Can be \fINULL\fR.
.RE

.sp
.ne 2
.na
\fB\fIcompletion_flags\fR\fR
.ad
.RS 20n
Flags for posted Receive. The default \fBDAT_COMPLETION_DEFAULT_FLAG\fR is
0x00. Other values are as follows:
.sp
.ne 2
.na
\fBNotification of Completion\fR
.ad
.RS 30n
\fBDAT_COMPLETION_UNSIGNALLED_FLAG\fR
.sp
.ne 2
.na
\fB0x04\fR
.ad
.RS 8n
Non-notification completion. Local Endpoint must be configured for Unsignaled
CompletionNotification Suppression.
.RE

.RE

.RE

.SH DESCRIPTION
.sp
.LP
The \fBdat_ep_post_recv()\fR function requests the receive of the data over the
connection of the \fIep_handle\fR Endpoint of the incoming message into the
\fIlocal_iov\fR.
.sp
.LP
The \fInum_segments\fR parameter specifies the number of segments in the
\fIlocal_iov\fR. The \fIlocal_iov\fR segments are filled in the I/O Vector
order until the whole message is received. This ensures that all the "front"
segments of the \fIlocal_iov\fR I/O Vector are completely filled, only one
segment is partially filled, if needed, and all segments that follow it are not
filled at all.
.sp
.LP
The \fIuser_cookie\fR allows Consumers to have unique identifiers for each DTO.
These identifiers are completely under user control and are opaque to the
Provider. There is no requirement on the Consumer that the value
\fIuser_cookie\fR should be unique for each DTO. The \fIuser_cookie\fR is
returned to the Consumer in the Completion event for the posted Receive.
.sp
.LP
The completion of the posted Receive is reported to the Consumer asynchronously
through a DTO Completion event based on the configuration of the connection for
Solicited Wait and the specified \fIcompletion_flags\fR value for the matching
Send. The value of \fBDAT_COMPLETION _UNSIGNALLED_FLAG\fR is only valid if the
Endpoint Recv Completion Flags \fBDAT_COMPLETION_UNSIGNALLED_FLAG\fR.
Otherwise, \fBDAT_INVALID_PARAMETER\fR is returned.
.sp
.LP
A Consumer must not modify the \fIlocal_iov\fR or its content until the DTO is
completed. When a Consumer does not adhere to this rule, the behavior of the
```

Provider and the underlying Transport is not defined. Providers that allow Consumers to get ownership of the \fIlocal_iov\fR but not the memory it specified back after the \fBdat_ep_post_recv()\fR returns should document this behavior and also specify its support in Provider attributes. This behavior allows Consumer full control of the \fIlocal_iov\fR content after \fBdat_ep_post_recv()\fR returns. Because this behavior is not guaranteed by all Providers, portable Consumers should not rely on this behavior. Consumers **should not rely on the Provider copying \fIlocal_iov\fR information.**
shouldnot rely on the Provider copying \fIlocal_iov\fR information.
.sp
.LP
The \fBDAT_SUCCESS\fR return of the \fBdat_ep_post_recv()\fR is at least the equivalent of posting a Receive operation directly by native Transport. Providers should avoid resource allocation as part of \fBdat_ep_post_recv()\fR to ensure that this operation is nonblocking and thread safe for an UpCall.
.sp
.LP
If the size of an incoming message is larger than the size of the \fIlocal_iov\fR, the reported status of the posted Receive DTO in the corresponding Completion DTO event is \fBDAT_DTO_LENGTH_ERROR\fR. If the reported status of the Completion DTO event corresponding to the posted Receive DTO is not \fBDAT_DTO_SUCCESS\fR, the content of the \fIlocal_iov\fR is not defined.
.sp
.LP
The operation is valid for all states of the Endpoint. The actual data transfer does not take place until the Endpoint is in the \fBDAT_EP_STATE_CONNECTED\fR state. The operation on the Endpoint in \fBDAT_EP_STATE_DISCONNECTED\fR is allowed. If the operation returns successfully, the posted Recv is immediately flushed to \fIrecv_evd_handle\fR.
.SH RETURN VALUES
.sp
.ne 2
.na
\fB\fBDAT_SUCCESS\fR\fR
.ad
.RS 30n
The operation was successful.
.RE

.sp
.ne 2
.na
\fB\fBDAT_INSUFFICIENT_RESOURCES\fR\fR
.ad
.RS 30n
The operation failed due to resource limitations.
.RE

.sp
.ne 2
.na
\fB\fBDAT_INVALID_PARAMETER\fR\fR
.ad
.RS 30n
Invalid parameter. For example, one of the IOV segments pointed to a memory outside its LMR.
.RE

.sp
.ne 2
.na
\fB\fBDAT_INVALID_HANDLE\fR\fR
.ad
.RS 30n
The \fIep_handle\fR parameter is invalid.

.RE

.sp
.ne 2
.na
\fB\fBDAT_PROTECTION_VIOLATION\fR\fR
.ad
.RS 30n
Protection violation for local or remote memory access. Protection Zone mismatch between an LMR of one of the \fIlocal_iov\fR segments and the local Endpoint.
.RE

.sp
.ne 2
.na
\fB\fBDAT_PRIVILEGES_VIOLATION\fR\fR
.ad
.RS 30n
Privileges violation for local or remote memory access. One of the LMRs used in \fIlocal_iov\fR was either invalid or did not have the local read privileges.
.RE

.SH USAGE
.sp
.LP
For best Recv operation performance, the Consumer should align each buffer segment of \fIlocal_iov\fR to the Optimal Buffer Alignment attribute of the Provider. For portable applications, the Consumer should align each buffer segment of \fIlocal_iov\fR to the \fBDAT_OPTIMAL_ALIGNMENT\fR.
.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE ATTRIBUTE VALUE
_
Interface Stability   Standard: uDAPL, 1.1, 1.2
_
MT-Level   Unsafe
.TE

.SH SEE ALSO
.sp
.LP
\fBlibdat\fR(3LIB), \fBattributes\fR(5)

```
**********************************************************
   12908 Sun Sep 16 19:22:55 2018
new/usr/src/man/man3dat/dat_ia_query.3dat
9842 man page typos and spelling
**********************************************************
```

```
'\" te
.\" This manual page is derived from the DAT/uDAPL 1.2 specification.
.\" Portions Copyright (c) 2007, Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH DAT_IA_QUERY 3DAT "Jul 16, 2004"
.SH NAME
dat_ia_query \- query an IA
.SH SYNOPSIS
.LP
.nf
cc [ \fIflag\fR\&.\|.\|. ] \fIfile\fR\&.\|.\|. \fB-ldat\fR [ \fIlibrary\fR\&.\|.
#include <\fBdat/udat.h\fR>

DAT_RETURN
    dat_ia_query (
    IN    DAT_IA_HANDLE           \fIia_handle\fR,
    OUT   DAT_EVD_HANDLE          *\fIasync_evd_handle\fR,
    IN    DAT_IA_ATTR_MASK        \fIia_attr_mask\fR,
    OUT   DAT_IA_ATTR             *\fIia_attributes\fR,
    IN    DAT_PROVIDER_ATTR_MASK  \fIprovider_attr_mask\fR,
    OUT   DAT_PROVIDER_ATTR       *\fIprovider_attributes\fR
    )
.fi

.SH PARAMETERS
.sp
.ne 2
.na
\fB\fIia_handle\fR\fR
.ad
.RS 23n
Handle for an open instance of an IA.
.RE

.sp
.ne 2
.na
\fB\fIasync_evd_handle\fR\fR
.ad
.RS 23n
Handle for an Event Dispatcher for asynchronous events generated by the IA.
.RE

.sp
.ne 2
.na
\fB\fIia_attr_mask\fR\fR
.ad
.RS 23n
Mask for the \fIia_attributes\fR.
.RE

.sp
.ne 2
.na
\fB\fIia_attributes\fR\fR
.ad
.RS 23n
Pointer to a Consumer-allocated structure that the Provider fills with IA
```



```
attributes.
.RE

.sp
.ne 2
.na
\fB\fIprovider_attr_mask\fR\fR
.ad
.RS 23n
Mask for the \fIprovider_attributes\fR.
.RE

.sp
.ne 2
.na
\fB\fIprovider_attributes\fR\fR
.ad
.RS 23n
Pointer to a Consumer-allocated structure that the Provider fills with Provider
attributes.
.RE

.SH DESCRIPTION
.sp
.LP
The \fBdat_ia_query()\fR functions provides the Consumer with the IA
parameters, as well as the IA and Provider attributes. Consumers pass in
pointers to Consumer-allocated structures for the IA and Provider attributes
that the Provider fills.
.sp
.LP
The \fIia_attr_mask\fR and \fIprovider_attr_mask\fR parameters allow the
Consumer to specify which attributes to query. The Provider returns values for
requested attributes. The Provider can also return values for any of the other
attributes.
.SS "Interface Adapter Attributes"
.sp
.LP
The IA attributes are common to all open instances of the IA. DAT defines a
method to query the IA attributes but does not define a method to modify them.
.sp
.LP
If IA is multiported, each port is presented to a Consumer as a separate IA.
.sp
.ne 2
.na
\fBAdapter name:\fR
.ad
.br
.na
\fB\fR
.ad
.sp .6
.RS 4n
The name of the IA controlled by the Provider. The same as \fIia_name_ptr\fR.
.RE

.sp
.ne 2
.na
\fBVendor name:\fR
.ad
.sp .6
.RS 4n
Vendor if IA hardware.
.RE
```

```
.sp
.ne 2
.na
\fBHW version major:\fR
.ad
.sp .6
.RS 4n
Major version of IA hardware.
.RE

.sp
.ne 2
.na
\fBHW version minor:\fR
.ad
.sp .6
.RS 4n
Minor version of IA hardware.
.RE

.sp
.ne 2
.na
\fBFirmware version major:\fR
.ad
.sp .6
.RS 4n
Major version of IA firmware.
.RE

.sp
.ne 2
.na
\fBFirmware version minor:\fR
.ad
.sp .6
.RS 4n
Minor version of IA firmware.
.RE

.sp
.ne 2
.na
\fBIA_address_ptr:\fR
.ad
.sp .6
.RS 4n
An address of the interface Adapter.
.RE

.sp
.ne 2
.na
\fBMax EPs:\fR
.ad
.sp .6
.RS 4n
Maximum number of Endpoints that the IA can support. This covers all Endpoints
in all states, including the ones used by the Providers, zero or more
applications, and management.
.RE

.sp
.ne 2
.na
```



```
\fBMax DTOs per EP:\fR
.ad
.sp .6
.RS 4n
Maximum number of DTOs and RMR_binds that any Endpoint can support for a single
direction. This means the maximum number of outstanding and in-progress Send,
RDMA Read, RDMA Write DTOs, and RMR Binds at any one time for any Endpoint; and
maximum number of outstanding and in-progress Receive DTOs at any one time for
any Endpoint.
.RE

.sp
.ne 2
.na
\fBMax incoming RDMA Reads per EP:\fR
.ad
.sp .6
.RS 4n
Maximum number of RDMA Reads that can be outstanding per (connected) Endpoint
with the IA as the target.
.RE

.sp
.ne 2
.na
\fBMax outgoing RDMA Reads per EP:\fR
.ad
.sp .6
.RS 4n
Maximum number of RDMA Reads that can be outstanding per (connected) Endpoint
with the IA as the originator.
.RE

.sp
.ne 2
.na
\fBMax EVDs:\fR
.ad
.sp .6
.RS 4n
Maximum number of Event Dispatchers that an IA can support. An IA cannot
support an Event Dispatcher directly, but indirectly by Transport-specific
Objects, for example, Completion Queues for Infiniband\(tm and VI. The Event
Dispatcher Objects can be shared among multiple Providers and similar Objects
from other APIs, for example, Event Queues for uDAPL.
.RE

.sp
.ne 2
.na
\fBMax EVD queue size:\fR
.ad
.sp .6
.RS 4n
Maximum size of the EVD queue supported by an IA.
.RE

.sp
.ne 2
.na
\fBMax IOV segments per DTO:\fR
.ad
.sp .6
.RS 4n
Maximum entries in an IOV list that an IA supports. Notice that this number
cannot be explicit but must be implicit to transport-specific Object entries.
```

```
257 For example, for IB, it is the maximum number of scatter/gather entries per
258 Work Request, and for VI it is the maximum number of data segments per VI
259 Descriptor.
260 .RE

262 .sp
263 .ne 2
264 .na
265 \fBMax LMRs:\fR
266 .ad
267 .sp .6
268 .RS 4n
269 Maximum number of Local Memory Regions IA supports among all Providers and
270 applications of this IA.
271 .RE

273 .sp
274 .ne 2
275 .na
276 \fBMax LMR block size:\fR
277 .ad
278 .sp .6
279 .RS 4n
280 Maximum contiguous block that can be registered by the IA.
281 .RE

283 .sp
284 .ne 2
285 .na
286 \fBMac LMR VA:\fR
287 .ad
288 .sp .6
289 .RS 4n
290 Highest valid virtual address within the context of an LMR. Frequently, IAs on
291 32-bit architectures support only 32-bit local virtual addresses.
292 .RE

294 .sp
295 .ne 2
296 .na
297 \fBMax PZs:\fR
298 .ad
299 .sp .6
300 .RS 4n
301 Maximum number of Protection Zones that the IA supports.
302 .RE

304 .sp
305 .ne 2
306 .na
307 \fBMax MTU size:\fR
308 .ad
309 .sp .6
310 .RS 4n
311 Maximum message size supported by the IA
312 .RE

314 .sp
315 .ne 2
316 .na
317 \fBMax RDMA size:\fR
318 .ad
319 .sp .6
320 .RS 4n
321 Maximum RDMA size supported by the IA
322 .RE
```



```
324 .sp
325 .ne 2
326 .na
327 \fBMax RMRs:\fR
328 .ad
329 .sp .6
330 .RS 4n
331 Maximum number of RMRs an IA supports among all Providers and applications of
332 this IA.
333 .RE

335 .sp
336 .ne 2
337 .na
338 \fBMax RMR target address:\fR
339 .ad
340 .sp .6
341 .RS 4n
342 Highest valid target address with the context of a local RMR. Frequently, IAs
343 on 32-bit architectures support only 32-bit local virtual addresses.
344 .RE

346 .sp
347 .ne 2
348 .na
349 \fBNum transport attributes:\fR
350 .ad
351 .sp .6
352 .RS 4n
353 Number of transport-specific attributes.
354 .RE

356 .sp
357 .ne 2
358 .na
359 \fBTransport-specific attributes:\fR
360 .ad
361 .sp .6
362 .RS 4n
363 Array of transport-specific attributes. Each entry has the format of
364 \fBDAT_NAMED_ATTR\fR, which is a structure with two elements. The first element
365 is the name of the attribute. The second element is the value of the attribute
366 as a string.
367 .RE

369 .sp
370 .ne 2
371 .na
372 \fBNum vendor attributes:\fR
373 .ad
374 .sp .6
375 .RS 4n
376 Number of vendor-specific attributes.
377 .RE

379 .sp
380 .ne 2
381 .na
382 \fBVendor-specific attributes:\fR
383 .ad
384 .sp .6
385 .RS 4n
386 Array of vendor-specific attributes. Each entry has the format of
387 \fBDAT_NAMED_ATTR\fR, which is a structure with two elements. The first element
388 is the name of the attribute. The second element is the value of the attribute
```

```
389 as a string.
390 .RE

392 .SS "DAPL Provider Attributes"
396 .sp
393 .LP
394 The provider attributes are specific to the open instance of the IA. DAT
395 defines a method to query Provider attributes but does not define a method to
396 modify them.
397 .sp
398 .ne 2
399 .na
400 \fBProvider name:\fR
401 .ad
402 .sp .6
403 .RS 4n
404 Name of the Provider vendor.
405 .RE

407 .sp
408 .ne 2
409 .na
410 \fBProvider version major:\fR
411 .ad
412 .sp .6
413 .RS 4n
414 Major Version of uDAPL Provider.
415 .RE

417 .sp
418 .ne 2
419 .na
420 \fBProvider version minor:\fR
421 .ad
422 .sp .6
423 .RS 4n
424 Minor Version of uDAPL Provider.
425 .RE

427 .sp
428 .ne 2
429 .na
430 \fBDAPL API version major:\fR
431 .ad
432 .sp .6
433 .RS 4n
434 Major Version of uDAPL API supported.
435 .RE

437 .sp
438 .ne 2
439 .na
440 \fBDAPL API version minor:\fR
441 .ad
442 .sp .6
443 .RS 4n
444 Minor Version of uDAPL API supported.
445 .RE

447 .sp
448 .ne 2
449 .na
450 \fBLMR memory types supported:\fR
451 .ad
452 .sp .6
453 .RS 4n
```



```
454 Memory types that LMR Create supports for memory registration. This value is a
455 union of LMR Memory Types \fBDAT_MEM_TYPE_VIRTUAL\fR, \fBDAT_MEM_TYPE_LMR\fR,
456 and \fBDAT_MEM_TYPE_SHARED_VIRTUAL\fR that the Provider supports. All Providers
457 must support the following Memory Types: \fBDAT_MEM_TYPE_VIRTUAL\fR,
458 \fBDAT_MEM_TYPE_LMR\fR, and \fBDAT_MEM_TYPE_SHARED_VIRTUAL\fR.
459 .RE

461 .sp
462 .ne 2
463 .na
464 \fBIOV ownership:\fR
465 .ad
466 .sp .6
467 .RS 4n
468 An enumeration flag that specifies the ownership of the local buffer
469 description (IOV list) after post DTO returns. The three values are as follows:
470 .RS +4
471 .TP
472 .ie t \(bu
473 .el o
474 \fBDAT_IOV_CONSUMER\fR indicates that the Consumer has the ownership of the
475 local buffer description after a post returns.
476 .RE
477 .RS +4
478 .TP
479 .ie t \(bu
480 .el o
481 \fBDAT_IOV_PROVIDER_NOMOD\fR indicates that the Provider still has ownership of
482 the local buffer description of the DTO when the post DTO returns, but the
483 Provider does not modify the buffer description.
484 .RE
485 .RS +4
486 .TP
487 .ie t \(bu
488 .el o
489 \fBDAT_IOV_PROVIDER_MOD\fR indicates that the Provider still has ownership of
490 the local buffer description of the DTO when the post DTO returns and can
491 modify the buffer description.
492 .RE
493 In any case, the Consumer obtains ownership of the local buffer description
494 after the DTO transfer is completed and the Consumer is notified through a DTO
495 completion event.
496 .RE

498 .sp
499 .ne 2
500 .na
501 \fBQOS supported:\fR
502 .ad
503 .sp .6
504 .RS 4n
505 The union of the connection QOS supported by the Provider.
506 .RE

508 .sp
509 .ne 2
510 .na
511 \fBCompletion flags supported:\fR
512 .ad
513 .sp .6
514 .RS 4n
515 The following values for the completion flag \fBDAT_COMPLETION_FLAGS\fR are
516 supported by the Provider: \fBDAT_COMPLETION_SUPPRESS_FLAG\fR,
517 \fBDAT_COMPLETION_UNSIGNALLED_FLAG\fR,
518 \fBDAT_COMPLETION_SOLICITED_WAIT_FLAG\fR, and
519 \fBDAT_COMPLETION_BARRIER_FENCE_FLAG\fR.
```

```
.RE

.sp
.ne 2
.na
\fBThread safety:\fR
.ad
.sp .6
.RS 4n
Provider Library thread safe or not. The Provider Library is not required to be
thread safe.
.RE

.sp
.ne 2
.na
\fBMax private data size:\fR
.ad
.sp .6
.RS 4n
Maximum size of private data the Provider supports. This value is at least 64
bytes.
.RE

.sp
.ne 2
.na
\fBMultipathing support:\fR
.ad
.sp .6
.RS 4n
Capability of the Provider to support Multipathing for connection
establishment.
.RE

.sp
.ne 2
.na
\fBEP creator for PSP:\fR
.ad
.sp .6
.RS 4n
Indicator for who can create an Endpoint for a Connection Request. For the
Consumer it is \fBDAT_PSP_CREATES_EP_NEVER\fR. For the Provider it is
\fBDAT_PSP_CREATES_EP_ALWAYS\fR. For both it is
\fBDAT_PSP_CREATES_EP_IFASKED\fR. This attribute is used for Public Service
Point creation.
.RE

.sp
.ne 2
.na
\fBPZ support:\fR
.ad
.sp .6
.RS 4n
Indicator of what kind of protection the Provider’s PZ provides.
.RE

.sp
.ne 2
.na
\fBOptimal Buffer Alignment:\fR
.ad
.sp .6
.RS 4n
```



```
Local and remote DTO buffer alignment for optimal performance on the Platform.
The \fBDAT_OPTIMAL_ALIGNMENT\fR must be divisible by this attribute value. The
maximum allowed value is \fBDAT_OPTIMAL_ALIGNMENT\fR, or 256.
The \fBDAT_OPTIMAL_ALIGMNEMT\fR must be divisible by this attribute value. The
maximum allowed value is \fBDAT_OPTIMAL_ALIGMNEMT\fR, or 256.
.RE

.sp
.ne 2
.na
\fBEVD stream merging support:\fR
.ad
.sp .6
.RS 4n
A 2D binary matrix where each row and column represent an event stream type.
Each binary entry is 1 if the event streams of its row and column can be fed to
the same EVD, and 0 otherwise.
.sp
More than two different event stream types can feed the same EVD if for each
pair of the event stream types the entry is 1.
.sp
The Provider should support merging of all event stream types.
.sp
The Consumer should check this attribute before requesting an EVD that merges
multiple event stream types.
.RE

.sp
.ne 2
.na
\fBNum provider attributes:\fR
.ad
.sp .6
.RS 4n
Number of Provider-specific attributes.
.RE

.sp
.ne 2
.na
\fBProvider-specific attributes:\fR
.ad
.sp .6
.RS 4n
Array of Provider-specific attributes. Each entry has the format of
\fBDAT_NAMED_ATTR\fR, which is a structure with two elements. The first element
is the name of the attribute. The second element is the value of the attribute
as a string.
.RE

.SH RETURN VALUES
.sp
.ne 2
.na
\fB\fBDAT_SUCCESS\fR\fR
.ad
.RS 25n
The operation was successful.
.RE

.sp
.ne 2
.na
\fB\fBDAT_INVALID_PARAMETER\fR\fR
.ad
.RS 25n
```

```
649 Invalid parameter;
650 .RE

652 .sp
653 .ne 2
654 .na
655 \fB\fBDAT_INVALID_HANDLE\fR\fR
656 .ad
657 .RS 25n
658 Invalid DAT handle; ia_handle is invalid.
659 .RE

661 .SH ATTRIBUTES
667 .sp
662 .LP
663 See \fBattributes\fR(5) for descriptions of the following attributes:
664 .sp

666 .sp
667 .TS
668 box;
669 c | c
670 l | l .
671 ATTRIBUTE TYPE  ATTRIBUTE VALUE
672 _
673 Interface Stability     Standard: uDAPL, 1.1, 1.2
674 _
675 MT-Level        Safe
676 .TE

678 .SH SEE ALSO
685 .sp
679 .LP
680 \fBlibdat\fR(3LIB), \fBattributes\fR(5)
```

```
**********************************************************
   11174 Sun Sep 16 19:22:55 2018
new/usr/src/man/man3project/getprojent.3project
9842 man page typos and spelling
**********************************************************
```

```
   1 '\" te
   2 .\" Copyright (c) 2004, Sun Microsystems, Inc.  All Rights Reserved.
   3 .\" The contents of this file are subject to the terms of the Common Development
   4 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
   5 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
   6 .TH GETPROJENT 3PROJECT "Apr 5, 2004"
   7 .SH NAME
   8 getprojent, getprojbyname, getprojbyid, getdefaultproj, inproj,
   9 getprojidbyname, setprojent, endprojent, fgetprojent \- project database entry
  10 operations
  11 .SH SYNOPSIS
  12 .LP
  13 .nf
  14 cc [ \fIflag\fR... ] \fIfile\fR... \(milproject [ \fIlibrary\fR... ]
  15 #include <project.h>

  17 \fBstruct project *\fR\fBgetprojent\fR(\fBstruct project *\fR\fIproj\fR, \fBvoid
  18      \fBsize_t\fR \fIbufsize\fR);
  19 .fi

  21 .LP
  22 .nf
  23 \fBstruct project *\fR\fBgetprojbyname\fR(\fBconst char *\fR\fIname\fR,
  24      \fBstruct project *\fR\fIproj\fR, \fBvoid *\fR\fIbuffer\fR, \fBsize_t\fR \f
  25 .fi

  27 .LP
  28 .nf
  29 \fBstruct project *\fR\fBgetprojbyid\fR(\fBprojid_t\fR \fIprojid\fR,
  30      \fBstruct project *\fR\fIproj\fR, \fBvoid *\fR\fIbuffer\fR, \fBsize_t\fR \f
  31 .fi

  33 .LP
  34 .nf
  35 \fBstruct project *\fR\fBgetdefaultproj\fR(\fBconst char *\fR\fIusername\fR,
  36      \fBstruct project *\fR\fIproj\fR, \fBvoid *\fR\fIbuffer\fR, \fBsize_t\fR \f
  37 .fi

  39 .LP
  40 .nf
  41 \fBint\fR \fBinproj\fR(\fBconst char *\fR\fIusername\fR, \fBconst char *\fR\fIpr
  42      \fBvoid *\fR\fIbuffer\fR, \fBsize_t\fR \fIbufsize\fR);
  43 .fi

  45 .LP
  46 .nf
  47 \fBprojid_t\fR \fBgetprojidbyname\fR(\fBconst char *\fR\fIname\fR);
  48 .fi

  50 .LP
  51 .nf
  52 \fBvoid\fR \fBsetprojent\fR(\fBvoid\fR);
  53 .fi

  55 .LP
  56 .nf
  57 \fBvoid\fR \fBendprojent\fR(\fBvoid\fR);
  58 .fi

  60 .LP
  61 .nf
```



```
  62 \fBstruct project *\fR\fBfgetprojent\fR(\fBFILE *\fR\fIf\fR, \fBstruct project *
  63      \fBvoid *\fR\fIbuffer\fR, \fBsize_t\fR \fIbufsize\fR);
  64 .fi

  66 .SH DESCRIPTION
  67 .sp
  67 .LP
  68 These functions are used to obtain entries describing user projects. Entries
  69 can come from any of the sources for a project specified in the
  70 \fB/etc/nsswitch.conf\fR file (see \fBnsswitch.conf\fR(4)).
  71 .sp
  72 .LP
  73 The \fBsetprojent()\fR, \fBgetprojent()\fR, and \fBendprojent()\fR functions
  74 are used to enumerate project entries from the database.
  75 .sp
  76 .LP
  77 The \fBsetprojent()\fR function effectively rewinds the project database to
  78 allow repeated searches. It sets (or resets) the enumeration to the beginning
  79 of the set of project entries. This function should be called before the first
  80 call to \fBgetprojent()\fR.
  81 .sp
  82 .LP
  83 The \fBgetprojent()\fR function returns a pointer to a structure containing the
  84 broken-out fields of an entry in the project database. When first called,
  85 \fBgetprojent()\fR returns a pointer to a project structure containing the
  86 first project structure in the project database. Successive calls can be used
  87 to read the entire database.
  88 .sp
  89 .LP
  90 The \fBendprojent()\fR function closes the project database and deallocates
  91 resources when processing is complete. It is permissible, though possibly less
  92 efficient, for the process to call more project functions after calling
  93 \fBendprojent()\fR.
  94 .sp
  95 .LP
  96 The \fBgetprojbyname()\fR function searches the project database for an entry
  97 with the project name specified by the character string \fIname\fR.
  98 .sp
  99 .LP
 100 The \fBgetprojbyid()\fR function searches the project database for an entry
 101 with the (numeric) project \fBID\fR specified by \fIprojid\fR.
 102 .sp
 103 .LP
 104 The \fBgetdefaultproj()\fR function first looks up the project key word in the
 105 \fBuser_attr\fR database used to define user attributes in restricted Solaris
 106 environments. If the database is available and the keyword is present, the
 107 function looks up the named project, returning \fINULL\fR if it cannot be found
 108 or if the user is not a member of the named project. If absent, the function
 109 looks for a match in the project database for the special project
 110 \fBuser\fR.\fIusername\fR. If no match is found, or if the user is excluded
 111 from project \fBuser\fR.\fIusername\fR, the function looks at the default group
 112 entry of the \fBpasswd\fR database for the user, and looks for a match in the
 113 project database for the special name \fBgroup\fR.\fIgroupname\fR, where
 114 \fIgroupname\fR is the default group associated with the password entry
 115 corresponding to the given \fIusername\fR. If no match is found, or if the user
 116 is excluded from project \fBgroup\fR.\fIgroupname\fR, the function returns
 117 \fINULL\fR. A special project entry called 'default' can be looked up and used
 118 as a last resort, unless the user is excluded from project 'default'. On
 119 successful lookup, this function returns a pointer to the valid \fBproject\fR
 120 structure. By convention, the user must have a default project defined on a
 121 system to be able to log on to that system.
 122 .sp
 123 .LP
 124 The \fBinproj()\fR function checks if the user specified by \fIusername\fR is
 125 able to use the project specified by \fIprojname\fR. This function returns 1 if
 126 the user belongs to the list of project's users, if there is a project's group
```

```
127 that contains the specified user, if project is a user's default project, or if
128 project's user or group list contains "*"  wildcard. In all other cases it
129 returns 0.
130 .sp
131 .LP
132 The \fBgetprojidbyname()\fR function searches the project database for an entry
133 with the project name specified by the character string name. This function
134 returns the project ID if the requested entry is found; otherwise it returns
135 \(mi1.
136 .sp
137 .LP
138 The \fBfgetprojent()\fR function, unlike the other functions described above,
139 does not use \fBnsswitch.conf\fR; it reads and parses the next line from the
140 stream \fIf\fR, which is assumed to have the format of the \fBproject\fR(4)
141 file. This function returns the same values as \fBgetprojent()\fR.
142 .sp
143 .LP
144 The \fBgetprojent()\fR, \fBgetprojbyname()\fR, \fBgetprojbyid()\fR,
145 \fBgetdefaultproj()\fR, and \fBinproj()\fR functions are reentrant interfaces
146 for operations with the \fBproject\fR database. These functions use buffers
147 supplied by the caller to store returned results and are safe for use in both
148 single-threaded and multithreaded applications.
149 .sp
150 .LP
151 Reentrant interfaces require the additional arguments \fIproj\fR, \fIbuffer\fR,
152 and \fIbufsize\fR. The \fIproj\fR argument must be a pointer to a \fBstruct
153 project\fR structure allocated by the caller. On successful completion, the
154 function returns the project entry in this structure. Storage referenced by the
155 \fBproject\fR structure is allocated from the memory provided with the
156 \fIbuffer\fR argument, which is \fIbufsize\fR bytes in size. The  content  of
157 the memory buffer could be lost in cases when these functions return errors.
158 .sp
159 .LP
160 For enumeration in multithreaded applications, the position within the
161 enumeration is a process-wide property shared by all threads. The
162 \fBsetprojent()\fR function can be used in a multithreaded application but
163 resets the enumeration position for all threads. If multiple threads interleave
164 calls to \fBgetprojent()\fR, the threads will enumerate disjoint subsets of the
165 project database. The \fBinproj()\fR, \fBgetprojbyname()\fR,
166 \fBgetprojbyid()\fR, and \fBgetdefaultproj()\fR functions leave the enumeration
167 position in an indeterminate state.
168 .SH RETURN VALUES
170 .sp
169 .LP
170 Project entries are represented by the \fBstruct project\fR structure defined
171 in <\fBproject.h\fR>.
172 .sp
173 .in +2
174 .nf
175 struct project {
176     char      *pj_name;     /* name of the project */
177     projid_t  pj_projid;    /* numerical project id */
178     char      *pj_comment;  /* project comment */
179     char      **pj_users;   /* vector of pointers to
180                                project user names */
181     char      **pj_groups;  /* vector of pointers to
182                                project group names */
183     char      *pj_attr;     /* project attributes */
184 };
185 .fi
186 .in -2

188 .sp
189 .LP
190 The \fBgetprojbyname()\fR and \fBgetprojbyid()\fR functions each return a
191 pointer to a \fBstruct project\fR if they successfully locate the requested
```



```
192 entry; otherwise they return \fINULL\fR.
193 .sp
194 .LP
195 The \fBgetprojent()\fR function returns a pointer to a \fBstruct project\fR if
196 it successfully enumerates an entry; otherwise it returns \fINULL\fR,
197 indicating the end of the enumeration.
198 .sp
199 .LP
200 The \fBgetprojidbyname()\fR function returns the project ID if the requested
202 The \fBgetprojidbyname()\fR function returns the project ID if the requsted
201 entry is found; otherwise it returns \(mi1 and sets errno to indicate the
202 error.
203 .sp
204 .LP
205 When the pointer returned by the reentrant functions \fBgetprojbyname()\fR,
206 \fBgetprojbyid()\fR, and \fBgetprojent()\fR is non-null, it is always equal to
207 the \fIproj\fR pointer that was supplied by the caller.
208 .sp
209 .LP
210 Upon failure, \fBNULL\fR is returned and errno is set to indicate the error.
211 .SH ERRORS
214 .sp
212 .LP
213 The \fBgetprojent()\fR, \fBgetprojbyname()\fR, \fBgetprojbyid()\fR,
214 \fBinproj()\fR, \fBgetprojidbyname()\fR, \fBfgetprojent()\fR, and
215 \fBgetdefaultproj()\fR functions will fail if:
216 .sp
217 .ne 2
218 .na
219 \fB\fBEINTR\fR\fR
220 .ad
221 .RS 10n
222 A signal was caught during the operation.
223 .RE

225 .sp
226 .ne 2
227 .na
228 \fB\fBEIO\fR\fR
229 .ad
230 .RS 10n
231 An I/O error has occurred.
232 .RE

234 .sp
235 .ne 2
236 .na
237 \fB\fBEMFILE\fR\fR
238 .ad
239 .RS 10n
240 There are \fBOPEN_MAX\fR file descriptors currently open in the calling
241 process.
242 .RE

244 .sp
245 .ne 2
246 .na
247 \fB\fBENFILE\fR\fR
248 .ad
249 .RS 10n
250 The maximum allowable number of files is currently open in the system.
251 .RE

253 .sp
254 .ne 2
255 .na
```

```
256 \fB\fBERANGE\fR\fR
257 .ad
258 .RS 10n
259 Insufficient storage was supplied by \fIbuffer\fR and \fIbufsize\fR to contain
260 the data to be referenced by the resulting \fBproject\fR structure.
261 .RE

263 .sp
264 .LP
265 These functions can also fail if the name service switch does not specify valid
266 \fBproject\fR(4) name service sources. In the case of an incompletely
267 configured name service switch configuration, \fBgetprojbyid()\fR and other
270 configurated name service switch configuration, \fBgetprojbyid()\fR and other
268 functions can return error values other than those documented above. These
269 conditions usually occur when the \fBnsswitch.conf\fR file indicates that one
270 or more name services is providing entries for the project database when that
271 name service does not actually make a project table available.
272 .SH USAGE
276 .sp
273 .LP
274 When compiling multithreaded applications, see \fBIntro\fR(3), Notes On
275 Multithreaded Applications.
276 .sp
277 .LP
278 Use of the enumeration interface \fBgetprojent()\fR is discouraged. Enumeration
279 is supported for the project file, NIS, and LDAP but in general is not
280 efficient. The semantics of enumeration are discussed further in
281 \fBnsswitch.conf\fR(4).
282 .SH ATTRIBUTES
287 .sp
283 .LP
284 See \fBattributes\fR(5) for descriptions of the following attributes:
285 .sp

287 .sp
288 .TS
289 box;
290 c | c
291 l | l .
292 ATTRIBUTE  TYPE ATTRIBUTE VALUE
293 _
294 Interface Stability     Evolving
295 _
296 MT-Level        See Description.
297 .TE

299 .SH SEE ALSO
305 .sp
300 .LP
301 \fBIntro\fR(3), \fBlibproject\fR(3LIB), \fBproject_walk\fR(3PROJECT),
302 \fBsysconf\fR(3C), \fBnsswitch.conf\fR(4), \fBproject\fR(4),
303 \fBattributes\fR(5)
```

```
**********************************************************
    2867 Sun Sep 16 19:22:55 2018
new/usr/src/man/man3volmgt/media_getid.3volmgt
9842 man page typos and spelling
**********************************************************
'\" te
.\" Copyright (c) 1998, Sun Microsystems, Inc.  All Rights Reserved
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH MEDIA_GETID 3VOLMGT "Mar 8, 2007"
.SH NAME
media_getid \- return the id of a piece of media
.SH SYNOPSIS
.LP
.nf
\fBcc \fR[flag \&.\|.\|.] \fIfile \fR\&.\|.\|.\fB\(mil\fRvolmgt [library \&.\|.\
\fBcc \fR[flag \&.\|.\|.] \fIfile \fR\&.\|.\|.\fB\(mil\fRvolgmt [library \&.\|.\

#include <volmgt.h>



\fBulonglong_t\fR \fBmedia_getid\fR(\fBchar *\fR\fIvol_path\fR);
.fi

.SH DESCRIPTION
.sp
.LP
This function is obsolete. The management of removable media by the Volume
Management feature, including \fBvold\fR, has been replaced by software that
supports the Hardware Abstraction Layer (HAL). Programmatic support for HAL is
through the HAL APIs, which are documented on the HAL web site. See
\fBhal\fR(5). The return value of this function is undefined.
.sp
.LP
\fBmedia_getid()\fR returns the \fIid\fR of a piece of media. Volume management
must be running. See \fBvolmgt_running\fR(3VOLMGT).
.SH PARAMETERS
.sp
.ne 2
.na
\fB\fIvol_path\fR\fR
.ad
.RS 12n
Path to the block or character special device.
.RE

.SH RETURN VALUES
.sp
.LP
The return from this function is undefined.
.SH EXAMPLES
.LP
\fBExample 1 \fRUsing \fBmedia_getid()\fR
.sp
.LP
The following example first checks if volume management is running, then checks
the volume management name space for \fIpath\fR, and then returns the \fIid\fR
for the piece of media.

.sp
.in +2
.nf
char *path;
```



```
        ...

if (volmgt_running()) {
     if (volmgt_ownspath(path)) {
          (void) printf("id of %s is %lld\en",
               path, media_getid(path));
     }
        }
.fi
.in -2

.sp
.LP
If a program using \fBmedia_getid()\fR does not check whether or not volume
management is running, then any \fBNULL\fR return value will be ambiguous, as
it could mean that either volume management does not have \fIpath\fR in its
name space, or volume management is not running.

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5)  for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
MT-Level        Safe
_
Interface Stability     Obsolete
.TE

.SH SEE ALSO
.sp
.LP
\fBvolmgt_ownspath\fR(3VOLMGT), \fBvolmgt_running\fR(3VOLMGT),
\fBattributes\fR(5), \fBhal\fR(5)
```

```
**********************************************************
    2671 Sun Sep 16 19:22:55 2018
new/usr/src/man/man3volmgt/volmgt_ownspath.3volmgt
9842 man page typos and spelling
**********************************************************
'\" te
.\" Copyright (c) 1998, Sun Microsystems, Inc.  All Rights Reserved
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH VOLMGT_OWNSPATH 3VOLMGT "Mar 8, 2007"
.SH NAME
volmgt_ownspath \- check volume management name space for path
.SH SYNOPSIS
.LP
.nf
\fBcc\fR [flag]\&.\|.\|. \fIfile\fR\&.\|.\|. \fB\(mil\fRvolmgt [library]\&.\|.\|
\fBcc\fR [flag]\&.\|.\|. \fIfile\fR\&.\|.\|. \fB\(mil\fRvolgmt [library]\&.\|.\|
#include <volmgt.h>

\fBint\fR \fBvolmgt_ownspath\fR(\fBchar *\fR\fIpath\fR);
.fi

.SH PARAMETERS
.sp
.ne 2
.na
\fB\fIpath\fR\fR
.ad
.RS 8n
A string containing the path.
.RE

.SH DESCRIPTION
.sp
.LP
This function is obsolete. The management of removable media by the Volume
Management feature, including \fBvold\fR, has been replaced by software that
supports the Hardware Abstraction Layer (HAL). Programmatic support for HAL is
through the HAL APIs, which are documented on the HAL web site. See
\fBhal\fR(5). The return value of this function is undefined.
.sp
.LP
The \fBvolmgt_ownspath()\fR function checks to see if a given \fIpath\fR is
contained in the volume management name space. This is achieved by comparing
the beginning of the supplied path name with the output from
\fBvolmgt_root\fR(3VOLMGT)
.SH RETURN VALUES
.sp
.LP
The return from this function is undefined.
.SH EXAMPLES
.LP
\fBExample 1 \fRUsing \fBvolmgt_ownspath()\fR
.sp
.LP
The following example first checks if volume management is running, then checks
the volume management name space for \fIpath\fR, and then returns the \fIid\fR
for the piece of media.

.sp
.in +2
.nf
char *path;

\&...
```



```
if (volmgt_running()) {
     if (volmgt_ownspath(path)) {
          (void) printf("id of %s is %lld\en",
               path, media_getid(path));
     }
}
.fi
.in -2

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5)  for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
MT-Level        Safe
_
Interface Stability     Obsolete
.TE

.SH SEE ALSO
.sp
.LP
\fBvolmgt_root\fR(3VOLMGT), \fBvolmgt_running\fR(3VOLMGT), \fBattributes\fR(5),
\fBhal\fR(5)
```

```
**********************************************************
   33837 Sun Sep 16 19:22:55 2018
new/usr/src/man/man4/NISLDAPmapping.4
9842 man page typos and spelling
**********************************************************
```

```
   1 '\" te
   2 .\" Copyright (C) 2006, Sun Microsystems, Inc. All Rights Reserved
   3 .\" The contents of this file are subject to the terms of the Common Development
   4 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
   5 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
   6 .TH NISLDAPMAPPING 4 "Feb 25, 2017"
   7 .SH NAME
   8 NISLDAPmapping \- mapping file used by the NIS server components
   9 .SH SYNOPSIS
  10 .LP
  11 .nf
  12 \fB/var/yp/NISLDAPmapping\fR
  13 .fi

  15 .SH DESCRIPTION
  16 .LP
  17 The \fBNISLDAPmapping\fR file specifies the mapping between NIS map entries and
  18 equivalent Directory Information Tree (DIT) entries.
  19 .sp
  20 .LP
  21 The presence of \fB/var/yp/NISLDAPmapping\fR on a NIS master server causes that
  22 server to obtain NIS data from LDAP. See \fBypserv\fR(4). If
  23 \fB/var/yp/NISLDAPmapping\fR is present but the connection configuration file
  24 that is defined in \fB/etc/default/ypserv\fR cannot be found, a warning is
  25 logged. See \fBypserv\fR(1M).
  26 .sp
  27 .LP
  28 NIS slave servers always obtain their data from a NIS master server, whether or
  29 not that server is getting data from LDAP, and ignore the
  30 \fB/var/yp/NISLDAPmapping\fR file.
  31 .sp
  32 .LP
  33 A simple \fBNISLDAPmapping\fR file is created using \fBinityp2l\fR(1M). You can
  34 customize your \fBNISLDAPmapping\fR file as you require.
  35 .sp
  36 .LP
  37 Each attribute defined below can be specified
  38 in\fB/var/yp/NISLDAPmappingLDAP\fR or as an LDAP attribute. If both are
  39 specified, then the attribute in \fB/var/yp/NISLDAPmapping\fR (including empty
  40 values) takes precedence.
  41 .sp
  42 .LP
  43 A continuation is indicated by a '\e' (backslash) in the last position,
  44 immediately before the newline of a line. Characters are escaped, that is,
  45 exempted from special interpretation, when preceded by a backslash character.
  46 .sp
  47 .LP
  48 The '#' (hash) character starts a comment. White space is either ASCII space or
  49 a horizontal tab. In general, lines consist of optional white space, an
  50 attribute name, at least one white space character, and an attribute value.
  51 .SH EXTENDED DESCRIPTION
  52 .SS "File Syntax"
  53 .LP
  54 Repeated fields, with separator characters, are described by the following
  55 syntax:
  56 .sp
  57 .ne 2
  58 .na
  59 \fBOne or more entries\fR
  60 .ad
  61 .RS 24n
```



```
  62 entry:entry:entry
  63 .sp
  64 .in +2
  65 .nf
  66 entry[":"...]
  67 .fi
  68 .in -2

  70 .RE

  72 .sp
  73 .ne 2
  74 .na
  75 \fBZero or more entries\fR
  76 .ad
  77 .RS 24n
  78 .sp
  79 .in +2
  80 .nf
  81 [entry":"...]
  82 .fi
  83 .in -2

  85 .RE

  87 .SS "Attributes"
  88 .LP
  89 Attributes generally apply to one more more NIS maps. Map names can be
  90 specified either on their own,that is in \fBpasswd.byname\fR, in which case
  91 they apply to all domains, or for individual NIS domains, for example, in
  92 \fBpasswd.byname,example.sun.uk\fR. Where a map is mentioned in more than one
  93 attribute, both versions are applied. If any parts of the attributes are in
  94 conflict, the domain specific version takes precedence over the non-domain
  95 specific version.
  96 .sp
  97 .LP
  98 Each domain specific attributes must appear in \fBNISLDAPmapping\fR before any
  99 related non-domain specific attribute. If non-domain specific attributes appear
 100 first, behavior may be unpredictable. Errors are logged when non-domain
 101 specific attributes are found first.
 102 .sp
 103 .LP
 104 You can associate a group of map names with a \fBdatabaseId\fR. In effect, a
 105 macro is expanded to the group of names. Use this mechanism where the same
 106 group of names is used in many attributes or where domain specific map names
 107 are used. Then, you can make any changes to the domain name in one place.
 108 .sp
 109 .LP
 110 Unless otherwise noted, all elements of the syntaxes below may be surrounded by
 111 white space. Separator characters and white space must be escaped if they are
 112 part of syntactic elements.
 113 .sp
 114 .LP
 115 The following attributes are recognized.
 116 .sp
 117 .ne 2
 118 .na
 119 \fB\fBnisLDAPdomainContext\fR\fR
 120 .ad
 121 .sp .6
 122 .RS 4n
 123 The context to use for a NIS domain.
 124 .sp
 125 The syntax for \fBnisLDAPdomainContext\fR is:
 126 .sp
 127 .in +2
```

```
.nf
NISDomainName ":" context
.fi
.in -2

The following is an example of the \fBnisLDAPdomainContext\fR attribute:
.sp
.in +2
.nf
domain.one : dc=site, dc=company, dc=com
.fi
.in -2

The mapping file should define the context for each domain before any other
attribute makes use of the \fBNISDomainName\fR specified for that domain.
.RE

.sp
.ne 2
.na
\fB\fBnisLDAPyppasswddDomains\fR\fR
.ad
.sp .6
.RS 4n
Lists the domains for which password changes should be made. NIS password
change requests do not specify the domains in which any given password should
be changed. In traditional NIS this information is effectively hard coded in
the NIS makefile.
.sp
The syntax for the \fBnisLDAPyppasswddDomains\fR attribute is:
.sp
.in +2
.nf
domainname
.fi
.in -2

If there are multiple domains, use multiple \fBnisLDAPyppasswddDomain\fR
entries with one domainname per entry.
.RE

.sp
.ne 2
.na
\fB\fBnisLDAPdatabaseIdMapping\fR\fR
.ad
.sp .6
.RS 4n
Sets up an alias for a group of NIS map names. There is no default value.
.sp
The syntax for the \fBnisLDAPdatabaseIdMapping\fR attribute is:
.sp
.in +2
.nf
databaseId ":" ["["indexlist"]"] mapname[" "...]
.fi
.in -2

where
.sp
.in +2
.nf
databaseId      = Label identifying a (subset of a) NIS
                  object for mapping purposes.
indexlist       = fieldspec[","...]
fieldspec       = fieldname "=" fieldvalue
```



```
fieldname       = The name of a entry field as defined in
                  nisLDAPnameFields.
fieldvalue      = fieldvaluestring | \e" fieldvaluestring \e"
.fi
.in -2

\fBindexlist\fR is used for those cases where it is necessary to select a
subset of entries from a NIS map. The subset are those NIS entries that match
the \fBindexlist\fR. If there are multiple specifications indexed for a
particular NIS map, they are tried in the order retrieved until one matches.
Note that retrieval order usually is unspecified for multi-valued LDAP
attributes. Hence, if using indexed specifications when
\fBnisLDAPdatabaseIdMapping\fR is retrieved from LDAP, make sure that the
subset match is unambiguous.
.sp
If the \fBfieldvaluestring\fR contains white space or commas, it must either be
surrounded by double quotes, or the special characters must be escaped.
Wildcards are allowed in the \fBfieldvaluestring\fR. See Wildcards
.sp
To associate the \fBpasswd.byname\fR and \fBpasswd.byuid\fR maps with the
\fBpasswd databaseId\fR:
.sp
.in +2
.nf
passwd:passwd.byname passwd.byuid
.fi
.in -2

The \fBpasswd\fR and \fBpasswd.adjunct\fR \fBdatabaseIds\fR receive special
handling. In addition to its normal usage, \fBpasswd\fR defines which maps
\fByppasswdd\fR is to update when a \fBpasswd\fR is changed. In addition to its
normal usage \fBpasswd.adjunct\fR defines which maps \fByppasswdd\fR is to
update when an adjunct \fBpasswd\fR is changed.
.sp
You may not alias a single map name to a different name, as the results are
unpredictable.
.RE

.sp
.ne 2
.na
\fB\fBnisLDAPentryTtl\fR\fR
.ad
.sp .6
.RS 4n
Establish TTLs for NIS entries derived from LDAP.
.sp
The syntax for the \fBnisLDAPentryTtl\fR attribute is:
.sp
.in +2
.nf
mapName[" "...]":"
        initialTTLlo ":" initialTTLhi ":" runningTTL
.fi
.in -2

where
.sp
.ne 2
.na
\fB\fBinitialTTLlo\fR\fR
.ad
.RS 16n
The lower limit for the initial \fBTTL\fR (in seconds) for data read from LDAP
when the \fBypserv\fR starts. If the \fBinitialTTLhi\fR also is specified, the
actual \fBinitialTTL\fR will be randomly selected from the interval
```

```
\fBinitialTTLlo\fR to \fBinitialTTLhi\fR , inclusive. Leaving the field empty
yields the default value of 1800 seconds.
.RE

.sp
.ne 2
.na
\fB\fBinitialTTLhi\fR\fR
.ad
.RS 16n
The upper limit for the initial TTL. If left empty, defaults to 5400.
.RE

.sp
.ne 2
.na
\fB\fBrunningTTL\fR\fR
.ad
.RS 16n
The TTL (in seconds) for data retrieved from LDAP while the ypserv is running.
Leave the field empty to obtain the default value of 3600 seconds.
.RE

If there is no specification of \fBTTL\fRs for a particular map, the default
values are used.
.sp
If the \fBinitialTTLlo\fR and \fBinitialTTLhi\fR have the same value, the
effect will be that all data known to the \fBypserv\fR at startup times out at
the same time. Depending on NIS data lookup patterns, this could cause spikes
in ypserv-to-LDAP traffic. In order to avoid that, you can specify different
\fBinitialTTLlo\fR and \fBinitialTTLhi\fR values, and obtain a spread in
initial TTLs.
.sp
The following is an example of the \fBnisLDAPentryTtl\fR attribute used to
specify that entries in the NIS host maps read from LDAP should be valid for
four hours. When \fBypserv\fR restarts, the disk database entries are valid for
between two and three hours.
.sp
.in +2
.nf
hosts.byname hosts.byaddr:7200:10800:14400
.fi
.in -2

.RE

.sp
.ne 2
.na
\fB\fBnisLDAPobjectDN\fR\fR
.ad
.sp .6
.RS 4n
Specifies the connection between a group of NIS maps and the LDAP directory.
This attribute also defines the 'order' of the NIS maps. When NIS maps are bulk
copied to or from the DIT, they are processed in the same order as related
\fBnisLDAPobjectDN\fR attributes appear in \fB/var/yp/NISLDAPmapping.\fR
.sp
The syntax for the \fBnisLDAPobjectDN\fR\ attribute is:
.sp
.in +2
.nf
mapName[" "...] ":" objectDN *( ";" objectDN )
.fi
.in -2
```



```
where
.sp
.in +2
.nf
objectDN            = readObjectSpec [":"[writeObjectSpec]]
readObjectSpec      = [baseAndScope [filterAttrValList]]
writeObjectSpec     = [baseAndScope [attrValList]]
baseAndScope        = [baseDN] ["?" [scope]]
filterAttrValList   = ["?" [filter | attrValList]]]
scope               = "base" | "one" | "sub"
attrValList         = attribute "=" value
                              *("," attribute "=" value)
.fi
.in -2

The \fBbaseDN\fR defaults to the value of the \fBnisLDAPdomainContext\fR
attribute for the accessed domain. If the \fBbaseDN\fR ends in a comma, the
\fBnisLDAPdomainContext\fR value is appended.
.sp
\fBscope\fR defaults to one. \fBscope\fR has no meaning and is ignored in a
\fBwriteObjectSpec\fR.
.sp
The \fBfilter\fR is an LDAP search filter and has no default value.
.sp
The \fBattrValList\fR is a list of attribute and value pairs. There is no
default value.
.sp
As a convenience, if an \fBattrValList\fR is specified in a
\fBreadObjectSpec\fR, it is converted to a search filter by ANDing together the
attributes and the values. For example, the attribute and value list:
.sp
.in +2
.nf
objectClass=posixAccount,objectClass=shadowAccount
.fi
.in -2

is converted to the filter:
.sp
.in +2
.nf
(&(objectClass=posixAccount)\e
        (objectClass=shadowAccount))
.fi
.in -2

Map entries are mapped by means of the relevant mapping rules in the
\fBnisLDAPnameFields\fR and \fBnisLDAPattributeFromField\fR .
.sp
If a \fBwriteObjectSpec\fR is omitted, the effect is one of the following:
.RS +4
.TP
.ie t \(bu
.el o
If there is no trailing colon after the \fBreadObjectSpec\fR, then there is no
write at all.
.RE
.RS +4
.TP
.ie t \(bu
.el o
If there is a colon after the \fBreadObjectSpec\fR, then \fBwriteObjectSpec\fR
equals \fBreadObjectSpec\fR.
.RE
The following is an example of a \fBnisLDAPobjectDN\fR attribute declaration
that gets the \fBhosts.byaddr\fR map entries from the \fBou=Hosts\fR container
```

```
392 under the default search base and writes to the same place.
393 .sp
394 .in +2
395 .nf
396 hosts.byaddr:ou=Hosts,?one?objectClass=ipHost:
397 .fi
398 .in -2

400 The following is an example of a \fBnisLDAPobjectDN\fR attribute declaration
401 that obtains \fBpasswd\fR map entries from the \fBou=People\fR containers under
402 the default search base, and also from \fBdc=another,dc=domain\fR.
403 .sp
404 .in +2
405 .nf
406 passwd:ou=People,?one?\e
407                 objectClass=shadowAccount,\e
408                 objectClass=posixAccount:;\e
409        ou=People,dc=another,dc=domain,?one?\e
410                 objectClass=shadowAccount,\e
411                 objectClass=posixAccount
412 .fi
413 .in -2

415 .RE

417 .sp
418 .ne 2
419 .na
420 \fB\fBnisLDAPnameFields\fR\fR
421 .ad
422 .sp .6
423 .RS 4n
424 Specifies the content of entries in a NIS map and how they should be broken
425 into named fields. \fBnisLDAPnameFields\fR is required because NIS
426 maps do not store information in named fields.
427 .sp
428 The syntax for the \fBnisLDAPnameFields\fR attribute is as follows:
429 .sp
430 .in +2
431 .nf
432 "nisLDAPnameFields" mapName ":" "(" matchspec "," fieldNames ")"
433 fieldName       = nameOrArrayName[","...]
434 nameOrArrayName = Name of field or 'array' of repeated fields.
435 matchspec       = \e" formatString \e"
436 .fi
437 .in -2

439 \fBformatString\fR may contains a list of \fB%s\fR and \fB%a\fR elements each
440 of which represents a single named field or a list of repeated fields. A
441 \fB%a\fR field is interpreted as an IPv4 address or an IPv6 address in
442 preferred format. If an IPv6 address in non preferred format is found, then it
443 is converted and a warning is logged.
444 .sp
445 Where there are a list of repeated fields, the entire list is stored as one
446 entry. The fields are broken up into individual entries, based on the internal
447 separator, at a latter stage. Other characters represent separators which must
448 be present. Any separator, including whitespace, specified by the
449 \fBformatString\fR, may be surrounded by a number of whitespace and tab
450 characters. The whitespace and tab characters are ignored.
451 .sp
452 Regardless of the content of this entry some \fBfieldNames\fR are reserved:
453 .sp
454 .ne 2
455 .na
456 \fB\fBrf_key\fR\fR
457 .ad
```



```
458 .RS 18n
459 The DBM key value
460 .RE

462 .sp
463 .ne 2
464 .na
465 \fB\fBrf_ipkey\fR\fR
466 .ad
467 .RS 18n
468 The DBM key value handled as an IP address. See the discussion of \fB%a\fR
469 fields.
470 .RE

472 .sp
473 .ne 2
474 .na
475 \fB\fBrf_comment\fR\fR
476 .ad
477 .RS 18n
478 Everything following the first occurrence of a symbol. \fBrf_comment\fR is
479 defined by \fBnisLDAPcommentChar\fR.
480 .RE

482 .sp
483 .ne 2
484 .na
485 \fB\fBrf_domain\fR\fR
486 .ad
487 .RS 18n
488 The name of the domain in which the current NIS operation is being carried out.
489 .RE

491 .sp
492 .ne 2
493 .na
494 \fB\fBrf_searchipkey\fR\fR
495 .ad
496 .RS 18n
497 The \fBrf_searchkey\fR value handled as an IP address. See the discussion of
498 \fB%a\fR fields above.
499 .RE

501 .sp
502 .ne 2
503 .na
504 \fB\fBrf_searchkey\fR\fR
505 .ad
506 .RS 18n
507 See the description under \fBnisLDAPattributeFromField\fR below.
508 .RE

510 For example, the \fBrpc.bynumber\fR map has the format:
511 .sp
512 .in +2
513 .nf
514 name number alias[" "...]
515 .fi
516 .in -2

518 The NIS to LDAP system is instructed to break it into a name, a number, and an
519 array of alias field by the following entry in the mapping file:
520 .sp
521 .in +2
522 .nf
523 nisLDAPnameFields rpc.bynumber : \e
```

```
524         "%s %s %s", name,number,aliases)
525 .fi
526 .in -2

528 .RE

530 .sp
531 .ne 2
532 .na
533 \fB\fBnisLDAPsplitFields\fR\fR
534 .ad
535 .sp .6
536 .RS 4n
537 Defines how a field, or list of fields, named by \fBnisLDAPnameFields\fR is
538 split into subfields. The original field is compared with each line of this
539 attribute until one matches. When a match is found named subfields are
540 generated. In latter operations subfield names can be used in the same way as
541 other field names.
542 .sp
543 The syntax for the \fBnisLDAPsplitFields\fR attribute is as follows:
544 .sp
545 .in +2
546 .nf
547 "nisLDAPsplitFields" fieldName ":" splitSpec[","...]
548 splitSpec       = "(" matchspec "," subFieldNames ")"
549 fieldName       = Name of a field from nisLDAPnameFields
550 subFieldNames   = subFieldname[","...]
551 matchspec       = \e" formatString \e"
552 .fi
553 .in -2

555 The netgroup \fBmemberTriples\fR can have format \fB(host, user, domain)\fR or
556 \fBgroupname\fR. The format is specified by the attribute:
557 .sp
558 .in +2
559 .nf
560 nisLDAPsplitField memberTriple: \e
561       ("(%s,%s,%s)", host, user, domain) , \e
562       ("%s", group)
563 .fi
564 .in -2

566 Later operations can then use field names \fBhost\fR, \fBuser\fR, \fBdomain\fR,
567 \fBgroup\fR or \fBmemberTriple\fR. Because lines are processed in order, if
568 \fBhost\fR, \fBuser\fR and \fBdomain\fR are found, \fBgroup\fR will not be
569 generated.
570 .sp
571 Several maps and databaseIds may contain fields that are to be split in the
572 same way. As a consequence, the names of fields to be split must be unique
573 across all maps and databaseIds.
574 .sp
575 Only one level of splitting is supported. That is, a subfield cannot be split
575 Only one level of spliting is supported.That is, a subfield cannot be split
576 into further subfields.
577 .RE

579 .sp
580 .ne 2
581 .na
582 \fB\fBnisLDAPrepeatedFieldSeparators\fR\fR
583 .ad
584 .sp .6
585 .RS 4n
586 Where there is a list of repeated, splittable fields,
586 Where there is a list of repeated, splitable fields,
587 \fBnisLDAPrepeatedFieldSeparators\fR specifies which characters separate
```



```
588 instances of the splittable field.
588 instances of the splitable field.
589 .sp
590 The syntax for the \fBnisLDAPrepeatedFieldSeparators\fR attribute is as
591 follows:
592 .sp
593 .in +2
594 .nf
595 "nisLDAPrepeatedFieldSeparators" fieldName \e"sepChar[...]\e"
596 sepChar = A separator character.
597 .fi
598 .in -2

600 The default value is space or tab. If repeated splittable fields are adjacent,
600 The default value is space or tab. If repeated splitable fields are adjacent,
601 that is, there is no separating character, then the following should be
602 specified:
603 .sp
604 .in +2
605 .nf
606 nisLDAPrepeatedFieldSeparators netIdEntry: ""
607 .fi
608 .in -2

610 .RE

612 .sp
613 .ne 2
614 .na
615 \fB\fBnisLDAPcommentChar\fR\fR
616 .ad
617 .sp .6
618 .RS 4n
619 Specifies which character represents the start of the special comment field in
620 a given NIS map. If this attribute is not present then the default comment
621 character \fB#\fR is used.
622 .sp
623 To specify that a map uses a asterix to mark the start of comments.
624 .sp
625 .in +2
626 .nf
627 nisLDAPcommentChar mapname : '*'
628 .fi
629 .in -2

631 If a map cannot contain comments, then the following attribute should be
632 specified.
633 .sp
634 .in +2
635 .nf
636 nisLDAPcommentChar mapname : ''
637 .fi
638 .in -2

640 .RE

642 .sp
643 .ne 2
644 .na
645 \fB\fBnisLDAPmapFlags\fR\fR
646 .ad
647 .sp .6
648 .RS 4n
649  Indicates if \fBYP_INTERDOMAIN\fR or \fBYP_SECURE\fR entries should be created
650 in a map. Using \fBnisLDAPmapFlags\fR is equivalent to running
651 \fBmakedbm\fR(1M) with the \fB-b\fR or the \fB-s\fR option. When a map is
```

```
652 created from the contents of the DIT, the mapping file attribute is the only
653 source for the \fBYP_INTERDOMAIN\fR or \fBYP_SECURE\fR entries.
654 .sp
655 The syntax for the \fBnisLDAPmapFlags\fR attribute is as follows:
656 .sp
657 .in +2
658 .nf
659 "nisLDAPmapFlags" mapname ":" ["b"]["s"]
660 .fi
661 .in -2

663 By default neither entry is created.
664 .RE

666 .sp
667 .ne 2
668 .na
669 \fB\fBnisLDAPfieldFromAttribute\fR\fR
670 .ad
671 .sp .6
672 .RS 4n
673 Specifies how a NIS entries field values are derived from LDAP attribute
674 values.
675 .sp
676 The syntax for the \fBnisLDAPfieldFromAttribute\fR attribute is as follows:
677 .sp
678 .in +2
679 .nf
680 mapName ":" fieldattrspec *("," fieldattrspec)
681 .fi
682 .in -2

684 The format of \fBfieldattrspec\fR is shown below at Field and Attribute
685 Conversion Syntax.
686 .sp
687 To map by direct copy and assignment the value of the \fBipHostNumber\fR
688 attribute to the \fBaddr\fR named field, for example:
689 .sp
690 .in +2
691 .nf
692 addr=ipHostNumber
693 .fi
694 .in -2

696 Formats for the named field and attribute conversion syntax are discussed
697 below, including examples of complex attribute to field conversions.
698 .RE

700 .sp
701 .ne 2
702 .na
703 \fB\fBnisLDAPattributeFromField\fR\fR
704 .ad
705 .sp .6
706 .RS 4n
707  Specifies how an LDAP attribute value is derived from a NIS entriy field
708 value.
709 .sp
710 The syntax for the \fBnisLDAPattributeFromField\fR attribute is as follows:
711 .sp
712 .in +2
713 .nf
714 mapName ":" fieldattrspec *("," fieldattrspec )
715 .fi
716 .in -2
```



```
718 The format of \fBfieldattrspec\fR is shown below at Field and Attribute
719 Conversion Syntax.
720 .sp
721 As a special case, if the \fBdn\fR attribute value derived from a
722 \fBfieldattrspec\fR ends in a comma ("\fB,\fR"), the domains context from
723 \fBnisLDAPdomainContext\fR is appended.
724 .sp
725 Use the following example to map the value of the \fBaddr\fR field to the
726 \fBipHostNumber\fR attribute by direct copy and assignment:
727 .sp
728 .in +2
729 .nf
730 ipHostNumber=addr
731 .fi
732 .in -2

734 All relevant attributes, including the \fBdn\fR, must be specified.
735 .sp
736 For every map it must be possible to rapidly find a DIT entry based on its key.
737 There are some maps for which a NIS to LDAP mapping for the key is not
738 desirable, so a key mapping cannot be specified. In these cases a mapping that
739 uses the reserved \fBrf_searchkey\fR must be specified. Mappings that use this
740 field name are ignored when information is mapped into the DIT.
741 .RE

743 .SS "Field and Attribute Conversion Syntax"
744 .LP
745 The general format of a \fBfieldattrspec\fR is:
746 .sp
747 .in +2
748 .nf
749 fieldattrspec     = lhs "=" rhs
750 lhs               = lval | namespeclist
751 rhs               = rval | [namespec]
752 namespeclist      = namespec | "(" namespec *("," namespec) ")"
753 .fi
754 .in -2

756 .sp
757 .LP
758 The \fBlval\fR and \fBrval\fR syntax are defined below at Values. The format of
759 a \fBnamespec\fR is:
760 .sp
761 .ne 2
762 .na
763 \fB\fBnamespec\fR\fR
764 .ad
765 .RS 16n
766 .sp
767 .in +2
768 .nf
769 ["ldap:"] attrspec [searchTriple] | ["yp:"] fieldname
770 [mapspec]
771 .fi
772 .in -2

774 .RE

776 .sp
777 .ne 2
778 .na
779 \fB\fBfieldname\fR\fR
780 .ad
781 .RS 16n
782 .sp
783 .in +2
```

```
.nf
field | "(" field ")"
.fi
.in -2

.RE

.sp
.ne 2
.na
\fB\fBattrspec\fR\fR
.ad
.RS 16n
.sp
.in +2
.nf
attribute | "(" attribute ")"
.fi
.in -2

.RE

.sp
.ne 2
.na
\fB\fBsearchTriple\fR\fR
.ad
.RS 16n
.sp
.in +2
.nf
":" [baseDN] ["?" [scope] ["?" [filter]]]
.fi
.in -2

.RE

.sp
.ne 2
.na
\fB\fBbaseDN\fR\fR
.ad
.RS 16n
Base DN for search
.RE

.sp
.ne 2
.na
\fB\fBfilter\fR\fR
.ad
.RS 16n
LDAP search filter
.RE

.sp
.ne 2
.na
\fB\fBmapspec\fR\fR
.ad
.RS 16n
Map name
.RE

.sp
.LP
```



```
The repository specification in a \fBnamespec\fR defaults is as follows:
.RS +4
.TP
.ie t \(bu
.el o
For assignments to a field:
.RS

.sp
.ne 2
.na
\fBon the \fBLHS\fR\fR
.ad
.RS 14n
yp
.RE

.sp
.ne 2
.na
\fBon the \fBRHS\fR\fR
.ad
.RS 14n
ldap
.RE

.RE

NIS field values on the \fBRHS\fR are those that exist before the NIS entry is
modified.
.RE
.RS +4
.TP
.ie t \(bu
.el o
For assignments to an attribute:
.RS

.sp
.ne 2
.na
\fBon the \fBLHS\fR\fR
.ad
.RS 14n
ldap
.RE

.sp
.ne 2
.na
\fBon the \fBRHS\fR\fR
.ad
.RS 14n
yp
.RE

.RE

Attribute values on the \fBRHS\fR are those that exist before the LDAP entry is
modified.
.RE
.sp
.LP
When the field or attribute name is enclosed in parenthesis, it denotes a list
of field or attribute values. For attributes, the meaning is the list of all
attributes of that name, and the interpretation depends on the context. See the
```

```
 916 discussion at Values. The list specification is ignored when a
 917 \fBsearchTriple\fR or \fBmapspec\fR is supplied.
 918 .sp
 919 .LP
 920 For fields, the \fBfieldname\fR syntax is used to map multiple attribute
 921 instances to multiple NIS entries.
 922 .sp
 923 .LP
 924 The \fBsearchTriple\fR can be used to specify an attribute from a location
 925 other than the read or write target. The defaultvalues are as follows:
 926 .sp
 927 .ne 2
 928 .na
 929 \fB\fBbaseDN\fR\fR
 930 .ad
 931 .RS 10n
 932 If \fBbaseDN\fR is omitted, the default is the current \fBobjectDN\fR. If the
 933 \fBbaseDN\fR ends in a comma, the context of the domain is appended from
 934 \fBnisLDAPdomainContext\fR .
 935 .RE

 937 .sp
 938 .ne 2
 939 .na
 940 \fB\fBscope\fR\fR
 941 .ad
 942 .RS 10n
 943 one
 944 .RE

 946 .sp
 947 .ne 2
 948 .na
 949 \fB\fBfilter\fR\fR
 950 .ad
 951 .RS 10n
 952 Empty
 953 .RE

 955 .sp
 956 .LP
 957 Similarly, the \fBmapspec\fR can be used to specify a field value from a NIS
 958 map other than the one implicitly indicated by the \fBmapName\fR. If
 959 \fBsearchTriple\fR or \fBmapspec\fR is explicitly specified in a
 960 \fBnamespec\fR, the retrieval or assignment, whether from or to LDAP or NIS, is
 961 performed without checking if read and write are enabled for the LDAP container
 962 or NIS map.
 963 .sp
 964 .LP
 965 The omission of the \fBnamespec\fR in an \fBrhs\fR is only allowed if the
 966 \fBlhs\fR is one or more attributes. The effect is to delete the specified
 967 attribute(s). In all other situations, an omitted \fBnamespec\fR means that the
 968 rule is ignored.
 969 .sp
 970 .LP
 971 The \fBfilter\fR can be a value. See Values. For example, to find the
 972 \fBipHostNumber\fRthat uses the \fBcn\fR, you specify the following in the
 973 \fBfilter\fR field:
 974 .sp
 975 .in +2
 976 .nf
 977 ldap:ipHostNumber:?one?("cn=%s", (cname, "%s.*"))
 978 .fi
 979 .in -2

 981 .sp
```



```
 982 .LP
 983 In order to remove ambiguity, the unmodified value of a single field or
 984 attribute must be specified as the following when used in the \fBfilter\fR
 985 field.
 986 .sp
 987 .in +2
 988 .nf
 989 ("%s", namespec)
 990 .fi
 991 .in -2

 993 .sp
 994 .LP
 995 If the \fBfilter\fR is not specified, the scope will be base, and the
 996 \fBbaseDN\fR is assumed to be the \fBDN\fR of the entry that contains the
 997 attribute to be retrieved or modified. To use previously existing field or
 998 attribute values in the mapping rules requires a lookup to find those values.
 999 Obviously, this adds to the time required to perform the modification. Also,
1000 there is a window between the time when a value is retrieved and then slightly
1001 later stored back. If the values have changed in the mean time, the change may
1002 be overwritten.
1003 .sp
1004 .LP
1005 When \fBfieldattrspecs\fR are grouped into rule sets, in the value of a
1006 \fBnisLDAPfieldFromAttribute\fR or \fBnisLDAPattributeFromField\fR attribute,
1007 the evaluation of the \fBfieldattrspecs\fR proceed in the listed order.
1008 However, evaluation may be done in parallel for multiple \fBfieldattrspecs\fR.
1009 If there is an error when evaluating a certain \fBfieldattrspec\fR, including
1010 retrieval or assignment of entry or field values, the extent to which the other
1011 \fBfieldattrspec\fR rules are evaluated is unspecified.
1012 .SS "Wildcards"
1013 .LP
1014 Where wildcard support is available, it is of the following limited form:
1015 .sp
1016 .ne 2
1017 .na
1018 \fB\fB*\fR\fR
1019 .ad
1020 .RS 9n
1021 Matches any number of characters
1022 .RE

1024 .sp
1025 .ne 2
1026 .na
1027 \fB\fB[x]\fR\fR
1028 .ad
1029 .RS 9n
1030 Matches the character x
1031 .RE

1033 .sp
1034 .ne 2
1035 .na
1036 \fB\fB[x-y]\fR\fR
1037 .ad
1038 .RS 9n
1039 Matches any character in the range x to y, inclusive
1040 .RE

1042 .sp
1043 .LP
1044 Combinations such as \fB[a-cA-C0123]\fR are also allowed, which would match any
1045 one of a, b, c, A, B, C, 0, 1, 2, or 3.
1046 .SS "Substring Extraction"
1047 .in +2
```

```
.nf
substringextract = "(" namespec "," matchspec ")"
name             = field or attribute name
matchspec        =
.fi
.in -2

.sp
.LP
The \fBmatchspec\fR is a string like the \fBsscanf\fR(3C) format string, except
that there may be at most one format specifier, a single \fB%s\fR. The output
value of the \fBsubstringextract\fR is the substring that matches the location
of the \fB%s\fR.
.sp
.LP
If there is no \fB%s\fR in the formatstring, it must instead be a single
character, which is assumed to be a field separator for the \fBnamespec\fR. The
output values are the field values. Wild cards are supported. If there is no
match, the output value is the empty string, " ".
.sp
.LP
For example, if the \fBfieldcname\fR has the value
\fBuser.some.domain.name.\fR, the value of the expression:
.sp
.in +2
.nf
(cname, "%s.*")
.fi
.in -2

.sp
.LP
is \fBuser\fR, which can be used to extract the user name from a NIS principal
name.
.sp
.LP
Similarly, use this expression to extract the third of the colon-separated
fields of the shadow field:
.sp
.in +2
.nf
(shadow, "*:*:%s:*")
.fi
.in -2

.sp
.LP
This form can be used to extract all of the shadow fields. However, a simpler
way to specify that special case is:
.sp
.in +2
.nf
(shadow, ":")
.fi
.in -2

.SS "Values"
.in +2
.nf
lval            = "(" formatspec "," namespec *("," namespec) ")"
rval            = "(" formatspec ["," namelist ["," elide] ] ")"

namelist        = name_or_sse *( "," name_or_sse)
name_or_sse     = namespec | removespec | substringextract
removespec      = list_or_name "-" namespec
list_or_name    = "(" namespec ")" | namespec
```



```
formatspec      =
formatstring    = A string combining text and % field specifications
elide           =
singlechar      = Any character
.fi
.in -2

.sp
.LP
The syntax above is used to produce \fBrval\fR values that incorporate field or
attribute values, in a manner like \fBsprintf\fR(3C), or to perform assignments
to \fBlval\fR like \fBsscanf\fR(3C). One important restriction is that the
format specifications,\fB%\fR plus a single character, use the designations
from \fBber_printf\fR(3LDAP). Thus, while \fB%s\fR is used to extract a string
value, \fB%i\fR causes BER conversion from an integer. Formats other than
\fB%s\fR, for instance, \fB%i\fR, are only meaningfully defined in simple
format strings without any other text.
.sp
.LP
The following \fBber_printf()\fR format characters are recognized:
.sp
.in +2
.nf
b  i  n  o  s
.fi
.in -2

.sp
.LP
If there are too few format specifiers, the format string may be repeated as
needed.
.sp
.LP
When used as an \fBlval\fR, there is a combination of pattern matching and
assignment, possibly to multiple fields or attributes.
.sp
.LP
In an assignment to an attribute, if the value of the \fBaddr\fR field is
\fB1.2.3.4\fR, the \fBrval\fR:
.sp
.in +2
.nf
("ipNetworkNumber=%s,", addr)
.fi
.in -2

.sp
.LP
produces the value \fBipNetworkNumber=1.2.3.4,\fR, while:
.sp
.in +2
.nf
("(%s,%s,%s)", host, user, domain)
.fi
.in -2

.sp
.LP
results in:
.sp
.in +2
.nf
(assuming host="xyzzy", user="-", domain="x.y.z")
"(xyzzy,-,x.y.z)"
.fi
.in -2
```

```
.sp
.LP
The elide character feature is used with attribute lists. So:
.sp
.in +2
.nf
("%s,", (mgrprfc822mailmember), ",")
.fi
.in -2

.sp
.LP
concatenates all \fBmgrprfc822mailmember\fR values into one comma-separated
string, and then elides the final trailing comma. Thus, for
.sp
.in +2
.nf
mgrprfc822mailmember=usera
mgrprfc822mailmember=userb
mgrprfc822mailmember=userc
.fi
.in -2

.sp
.LP
the value would be:
.sp
.in +2
.nf
usera,userb,userc
.fi
.in -2

.sp
.LP
As a special case, to combine an \fBLHS\fR extraction with an \fBRHS\fR
implicit list creates multiple entries and values. So
.sp
.in +2
.nf
("(%s,%s,%s)", host, user, domain)=(nisNetgroupTriple)
.fi
.in -2

.sp
.LP
creates one NIS entry for each \fBnisNetgroupTriple\fR value.
.sp
.LP
The \fB\&'removespec'\fR form is used to exclude previously assigned fields
values from a list. So, if an LDAP entry contains:
.sp
.in +2
.nf
name: foo
cn: foo
cn: foo1
cn: foo2
.fi
.in -2

.sp
.LP
and the mapping file specifies :
.sp
```



```
.in +2
.nf
myName = name, \e
myAliases = ("%s ", (cn) - yp:myName, " ")
.fi
.in -2

.sp
.LP
then the following assignments are carried out:
.RS +4
.TP
1.
Assign value \fBfoo\fR to \fBmyName\fR
.RE
.RS +4
.TP
2.
Assign value \fBfoo foo1 foo2\fR to \fBmyAliases\fR
.RE
.RS +4
.TP
3.
Remove value of \fBmyName\fR from value \fBmyAliases\fR
.RE
.sp
.LP
This results in the field values \fBmyName\fR is set to \fBfoo\fR, and
\fBmyAliases\fR is set to \fBfoo1 foo2\fR.
.SS "Assignments"
.LP
The assignment syntax, also found at Field and Attribute Conversion Syntax, is
as follows:
.sp
.in +2
.nf
fieldattrspec    = lhs "=" rhs
lhs              = lval | namespeclist
rhs              = rval | namespec
namespeclist     = namespec | "(" namespec *("," namespec) ")"
.fi
.in -2

.sp
.LP
The general form of a simple assignment, which is a one-to-one mapping of field
to attribute, is:
.sp
.in +2
.nf
("%s", fieldname)=("%s", attrname)
.fi
.in -2

.sp
.LP
As a convenient shorthand, this can also be written as:
.sp
.in +2
.nf
fieldname=attrname
.fi
.in -2

.sp
.LP
```

A list specification, which is a name enclosed in parenthesis, can be used to make many-to-many assignments. The expression:

```
.sp
.in +2
.nf
(fieldname)=(attrname)
.fi
.in -2
```

```
.sp
.LP
```

where there are multiple instances of \fBattrname\fR, creates one NIS entry for each such instance, differentiated by their \fBfieldname\fR values. The following combinations of lists are allowed, but they are not particularly useful:

```
.sp
.ne 2
.na
\fB\fB(attrname)=(fieldname)\fR\fR
.ad
.RS 26n
Equivalent to \fBattrname=fieldname\fR
.RE

.sp
.ne 2
.na
\fB\fBattrname=(fieldname)\fR\fR
.ad
.RS 26n
Equivalent to \fBattrname=fieldname\fR
.RE

.sp
.ne 2
.na
\fB\fB(fieldname)=attrname\fR\fR
.ad
.RS 26n
Equivalent to \fBfieldname=attrname\fR
.RE

.sp
.ne 2
.na
\fB\fBfieldname=(attrname)\fR\fR
.ad
.RS 26n
Equivalent to \fBfieldname=attrname\fR
.RE

.sp
.LP
```

If a multi-valued \fBRHS\fR is assigned to a single-valued \fBLHS\fR, the \fBLHS\fR value will be the first of the \fBRHS\fR values. If the \fBRHS\fR is an attribute list, the first attribute is the first one returned by the LDAP server when queried. Otherwise, the definition of "first"is implementation dependent.

```
.sp
.LP
```

Finally, the \fBLHS\fR can be an explicit list of fields or attributes, such as:

```
.sp
.in +2
.nf
(name1,name2,name3)
```



```
.fi
.in -2

.sp
.LP
```

If the \fBRHS\fR is single-valued, this assigns the \fBRHS\fR value to all entities in the list. If the \fBRHS\fR is multi-valued, the first value is assigned to the first entity of the list, the second value to the second entity, and so on. Excess values or entities are silently ignored.

```
.SH EXAMPLES
.LP
```

\fBExample 1 \fRAssigning an Attribute Value to a Field

```
.sp
.LP
```

The following example illustrates how to assign the value of the \fBipHostNumber\fR attribute to the \fBaddr\fR field

```
.sp
.in +2
.nf
addr=ipHostNumber
.fi
.in -2

.LP
```

\fBExample 2 \fRCreating Multiple NIS Entries from Multi-Valued LDAP Attributes

```
.sp
.LP
```

An LDAP entry with:

```
.sp
.in +2
.nf
cn=name1
cn=name2
cn=name3
.fi
.in -2

.sp
.LP
```

and the following assignments:

```
.sp
.in +2
.nf
cname=cn
(name)=(cn)
.fi
.in -2

.sp
.LP
```

creates three NIS entries. Other attributes and fields are omitted for clarity.

```
.sp
.in +2
.nf
cname=name1, name=name1
cname=name1, name=name2
cname=name1, name=name3
.fi
.in -2

.LP
```

\fBExample 3 \fRAssigning String Constants

```
1444 .sp
1445 .LP
1446 The following expression sets the \fBpasswd\fR field to x:

1448 .sp
1449 .in +2
1450 .nf
1451 passwd=("x")
1452 .fi
1453 .in -2

1455 .LP
1456 \fBExample 4 \fRSplitting Field Values to Multi-Valued Attributes
1457 .sp
1458 .LP
1459 The \fBexpansion\fR field contains a comma-separated list of alias member
1460 names. In the following example, the expression assigns each member name to an
1461 instance of \fBmgrprfc822mailmember\fR:

1463 .sp
1464 .in +2
1465 .nf
1466 (mgrprfc822mailmember)=(expansion, ",")
1467 .fi
1468 .in -2

1470 .SH FILES
1471 .ne 2
1472 .na
1473 \fB\fB/var/yp/NISLDAPmapping\fR\fR
1474 .ad
1475 .RS 26n
1476 Mapping file used by the NIS server components
1477 .RE

1479 .SH ATTRIBUTES
1480 .LP
1481 See \fBattributes\fR(5) for descriptions of the following attributes:
1482 .sp

1484 .sp
1485 .TS
1486 box;
1487 c | c
1488 l | l .
1489 ATTRIBUTE TYPE  ATTRIBUTE VALUE
1490 _
1491 Interface Stability     Obsolete
1492 .TE

1494 .SH SEE ALSO
1495 .LP
1496 \fBinityp2l\fR(1M), \fBmakedbm\fR(1M), \fBypserv\fR(1M),
1497 \fBber_printf\fR(3LDAP), \fBsprintf\fR(3C), \fBsscanf\fR(3C),
1498 \fBypserv\fR(4), \fBattributes\fR(5)
1499 .sp
1500 .LP
1501 \fISystem Administration Guide: Naming and Directory Services (DNS, NIS, and
1502 LDAP)\fR
```

```
**********************************************************
   14335 Sun Sep 16 19:22:56 2018
new/usr/src/man/man4/contract.4
9842 man page typos and spelling
**********************************************************
```

```
   1 '\" te
   2 .\" Copyright (c) 2004, Sun Microsystems, Inc. All Rights Reserved.
   3 .\" The contents of this file are subject to the terms of the Common Development
   4 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
   5 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
   6 .TH CONTRACT 4 "Nov 26, 2017"
   7 .SH NAME
   8 contract \- the contract file system
   9 .SH SYNOPSIS
  10 .LP
  11 .nf
  12 \fB/system/contract\fR
  13 .fi

  15 .SH DESCRIPTION
  16 .LP
  17 The \fB/system/contract\fR file system acts as the primary interface to the
  18 contract subsystem. There is a subdirectory of \fB/system/contract\fR for each
  19 available contract type.
  20 .sp
  21 .LP
  22 \fB/system/contract\fR can be mounted on any mount point, in addition to the
  23 standard \fB/system/contract\fR mount point, and can be mounted several places
  24 at once. Such additional mounts are allowed in order to facilitate the
  25 confinement of processes to subtrees of the file system using \fBchroot\fR(1M)
  26 and yet allow such processes to continue to use contract commands and
  27 interfaces.
  28 .sp
  29 .LP
  30 A combination of standard system calls (for example, \fBopen\fR(2),
  31 \fBclose\fR(2), and \fBpoll\fR(2)) and calls to \fBlibcontract\fR(3LIB) access
  32 \fB/system/contract\fR files.
  33 .sp
  34 .LP
  35 Consumers of the contract file system must be large file aware. See
  36 \fBlargefile\fR(5) and \fBlfcompile64\fR(5).
  37 .SS "DIRECTORY STRUCTURE"
  38 .LP
  39 At the top level, the \fB/system/contract\fR directory contains subdirectories
  40 named with each available contract type, and one special directory, \fBall\fR.
  41 Each of these directories is world-readable and world-searchable.
  42 .SS "STRUCTURE OF \fB/system/contract/\fItype\fR\fR"
  43 .LP
  44 Each \fB/system/contract/\fItype\fR\fR directory contains a fixed number of
  45 files. It also contains a variable number of subdirectories corresponding to
  46 existing contracts of type \fItype\fR and named with the decimal representation
  47 of the contracts' IDs.
  48 .sp
  49 .LP
  50 The following files are in a \fB/system/contract/\fItype\fR\fR directory:
  51 .sp
  52 .ne 2
  53 .na
  54 \fB\fBtemplate\fR\fR
  55 .ad
  56 .RS 12n
  57 Opening this file returns a file descriptor for a new \fItype\fR contract
  58 template.
  59 .sp
  60 You can use the following \fBlibcontract\fR(3LIB) calls on a template file
  61 descriptor:
```



```
  62 .sp
  63 .in +2
  64 .nf
  65 >

  67  ct_tmpl_activate(3contract)
  68  ct_tmpl_clear(3contract)
  69  ct_tmpl_create(3contract)
  70 .fi
  71 .in -2
  72 .sp

  74 See TERMS for additional template functions.
  75 .RE

  77 .sp
  78 .ne 2
  79 .na
  80 \fBlatest\fR
  81 .ad
  82 .RS 12n
  83 Opening this file returns a file descriptor for the status file of the last
  84 \fItype\fR contract written by the opening LWP. See \fBSTRUCTURE OF
  85 /system/contract/\fItype\fR/\fIid\fR\fR. If the opening LWP has not created a
  86 \fItype\fR contract, opening latest fails with \fBESRCH\fR.
  87 .RE

  89 .sp
  90 .ne 2
  91 .na
  92 \fBbundle\fR
  93 .ad
  94 .RS 12n
  95 Opening this file returns a file descriptor for an event endpoint which
  96 receives events from all \fItype\fR contracts on the system. No privileges are
  97 required to open a type bundle event endpoint. Events sent by contracts owned
  98 and written by users other than the reader's effective user id are invisible,
  99 that is, they are silently skipped, unless the reader has
 100 \fB{PRIV_CONTRACT_OBSERVER}\fR in its effective set. See \fBEVENTS\fR.
 101 .RE

 103 .sp
 104 .ne 2
 105 .na
 106 \fBpbundle\fR
 107 .ad
 108 .RS 12n
 109 Opening this file returns a file descriptor for an event endpoint which
 110 receives events from all \fItype\fR contracts held by the opening process. See
 111 \fBEVENTS\fR.
 112 .RE

 114 .SS "STRUCTURE OF /system/contract/all"
 115 .LP
 116 The \fB/system/contract/all\fR directory contains a numerically named file for
 117 each contract in the system. Each file is a symbolic link to the type-specific
 118 directory for that contract, that is \fB/system/contract/all/\fIid\fR\fR points
 119 to \fB/system/contract/\fItype\fR/\fIid\fR\fR.
 120 .SS "STRUCTURE OF /system/contract/\fItype\fR/\fIid\fR"
 121 .LP
 122 Each \fB/system/contract/\fItype\fR/\fIid\fR\fR directory contains the
 123 following files:
 124 .sp
 125 .ne 2
 126 .na
 127 \fBctl\fR
```

```
 128 .ad
 129 .RS 10n
 130 Opening this file returns a file descriptor for contract \fIid\fR's control
 131 file. The open fails if the opening process does not hold contract \fIid\fR and
 132 the contract has not been inherited by the process contract of which the
 133 opening process is a member. See \fBprocess\fR(4).
 134 .sp
 135 The following \fBlibcontract\fR(3LIB) calls can be made on a \fBctl\fR file
 136 descriptor if the contract is owned by the caller:
 137 .sp
 138 .in +2
 139 .nf
 140 ct_ctl_abandon(3contract)
 141 ct_ctl_newct(3contract)
 142 ct_ctl_ack(3contract)
 143 ct_ctl_qack(3contract)
 144 .fi
 145 .in -2
 146 .sp

 148 The following \fBlibcontract\fR(3LIB) call can be made on a ctl file descriptor
 149 if the contract doesn't have an owner:
 150 .sp
 151 .in +2
 152 .nf
 153 ct_ctl_adopt(3contract)
 154 .fi
 155 .in -2
 156 .sp

 158 .RE

 160 .sp
 161 .ne 2
 162 .na
 163 \fBstatus\fR
 164 .ad
 165 .RS 10n
 166 Opening this file returns a file descriptor for contract \fIid\fR's status
 167 file. The following \fBlibcontract\fR(3LIB) calls can be made on a status file
 168 descriptor:
 169 .LP
 170 .nf
 171 ct_status_read(3contract)
 172 .fi

 174  See STATUS.
 175 .RE

 177 .sp
 178 .ne 2
 179 .na
 180 \fBevents\fR
 181 .ad
 182 .RS 10n
 183 Opening this file returns a file descriptor for an event endpoint which
 184 receives events from contract \fIid\fR. See \fBEVENTS\fR.
 185 .sp
 186 Only a process which has the same effective user ID as the process owning the
 187 contract, the same effective user ID as the contract's author, or has
 188 \fB{PRIV_CONTRACT_OBSERVER}\fR in its effective set can open the event endpoint
 189 for a contract.
 190 .RE

 192 .SS "TERMS"
 193 .LP
```



```
 194 The following terms are defined for all contracts:
 195 .sp
 196 .ne 2
 197 .na
 198 \fBcookie\fR
 199 .ad
 200 .RS 25n
 201 Specifies a 64-bit quantity that the contract author can use to identify the
 202 contract. Use \fBct_tmpl_set_cookie\fR(3CONTRACT) to set this term.
 203 .RE

 205 .sp
 206 .ne 2
 207 .na
 208 \fBinformative event set\fR
 209 .ad
 210 .RS 25n
 211 Selects which events are delivered as informative events. Use
 212 \fBct_tmpl_set_informative\fR(3CONTRACT) to set this term.
 213 .RE

 215 .sp
 216 .ne 2
 217 .na
 218 \fBcritical event set\fR
 219 .ad
 220 .RS 25n
 221 Selects which events are delivered as critical events. Use
 222 \fBct_tmpl_set_critical\fR(3CONTRACT) to set this term.
 223 .RE

 225 .SS "STATUS"
 226 .LP
 227 A status object returned by \fBct_status_read\fR(3CONTRACT) contains the
 228 following pieces of information:
 229 .sp
 230 .ne 2
 231 .na
 232 \fBcontract ID\fR
 233 .ad
 234 .sp .6
 235 .RS 4n
 236 The numeric ID of the contract. Use \fBct_status_get_id\fR(3CONTRACT) to obtain
 237 this information.
 238 .RE

 240 .sp
 241 .ne 2
 242 .na
 243 \fBcontract type\fR
 244 .ad
 245 .sp .6
 246 .RS 4n
 247 The type of the contract, specified as a string. Obtained using
 248 \fBct_status_get_type\fR(3CONTRACT). The contract type is the same as its
 249 subdirectory name under \fB/system/contract\fR.
 250 .RE

 252 .sp
 253 .ne 2
 254 .na
 255 \fBcreator's zone ID\fR
 256 .ad
 257 .sp .6
 258 .RS 4n
 259 The zone ID of the process which created the contract. Obtained using
```

```
260 \fBct_status_get_zoneid\fR(3CONTRACT).
261 .RE

263 .sp
264 .ne 2
265 .na
266 \fBownership state\fR
267 .ad
268 .sp .6
269 .RS 4n
270 The state of the contract, specified as \fBCTS_OWNED\fR, \fBCTS_INHERITED\fR,
271 \fBCTS_ORPHAN\fR, or \fBCTS_DEAD\fR. Use \fBct_status_get_state\fR(3CONTRACT)
272 to obtain this information.
273 .RE

275 .sp
276 .ne 2
277 .na
278 \fBcontract holder\fR
279 .ad
280 .sp .6
281 .RS 4n
282 If the contract's state is \fBCTS_OWNED\fR, the ID of the process which owns
283 the contract. If the contract's state is \fBCTS_INHERITED\fR, the ID of the
284 contract which is acting as regent. If the contract's state is \fBCTS_ORPHAN\fR
285 or \fBCTS_DEAD\fR, this is undefined. Use \fBct_status_get_holder\fR(3CONTRACT)
286 to obtain this information.
287 .RE

289 .sp
290 .ne 2
291 .na
292 \fBnumber of critical events\fR
293 .ad
294 .sp .6
295 .RS 4n
296 The number of unacknowledged critical events pending on the contract's event
297 queue. Use \fBct_status_get_nevents\fR(3CONTRACT) to obtain this information.
298 .RE

300 .sp
301 .ne 2
302 .na
303 \fBnegotiation time\fR
304 .ad
305 .sp .6
306 .RS 4n
307 The time remaining before the current synchronous negotiation times out. Use
308 \fBct_status_get_ntime\fR(3CONTRACT) to obtain this information.
309 .RE

311 .sp
312 .ne 2
313 .na
314 \fBnegotiation quantum time\fR
315 .ad
316 .sp .6
317 .RS 4n
318 The time remaining before the current negotiation quantum runs out. Use
319 \fBct_status_get_qtime\fR(3CONTRACT) to obtain this information.
320 .RE

322 .sp
323 .ne 2
324 .na
325 \fBnegotiation event ID\fR
```



```
326 .ad
327 .sp .6
328 .RS 4n
329 The ID of the event which initiated the negotiation timeout. Use
330 \fBct_status_get_nevid\fR(3CONTRACT) to obtain this information.
331 .RE

333 .sp
334 .ne 2
335 .na
336 \fBcookie (term)\fR
337 .ad
338 .sp .6
339 .RS 4n
340 The contract's cookie term. Use \fBct_status_get_cookie\fR(3CONTRACT) to obtain
341 this information.
342 .RE

344 .sp
345 .ne 2
346 .na
347 \fBInformative event set (term)\fR
348 .ad
349 .sp .6
350 .RS 4n
351 The contract's informative event set. Use
352 \fBct_status_get_informative\fR(3CONTRACT) to obtain this information.
353 .RE

355 .sp
356 .ne 2
357 .na
358 \fBCritical event set (term)\fR
359 .ad
360 .sp .6
361 .RS 4n
362 The contract's critical event set. Use \fBct_status_get_critical\fR(3CONTRACT)
363 to obtain this information.
364 .RE

366 .SS "EVENTS"
367 .LP
368 All three event endpoints, \fB/system/contract/\fItype\fR/bundle\fR,
369 \fB/system/contract/\fItype\fR/pbundle\fR, and
370 \fB/system/contract/\fItype\fR/\fIid\fR/events\fR, are accessed in the same
371 manner.
372 .sp
373 .LP
374 The following \fBlibcontract\fR(3LIB) interfaces are used with an event
375 endpoint file descriptor:
376 .sp
377 .in +2
378 .nf
379 ct_event_read(3contract)
380 ct_event_read_critical(3contract)
381 ct_event_reset(3contract)
382 .fi
383 .in -2
384 .sp

386 .sp
387 .LP
388 To facilitate processes watching multiple event endpoints, it is possible to
389 poll(2) on event endpoints. When it is possible to receive on an endpoint file
390 descriptor, POLLIN is set for that descriptor.
391 .sp
```

```
.LP
An event object returned by \fBct_event_read\fR(3CONTRACT) contains the
following information:
.sp
.ne 2
.na
\fBcontract ID\fR
.ad
.RS 28n
The ID of the contract that generated the event. Use
\fBct_event_read\fR(3CONTRACT) to obtain this information.
.RE

.sp
.ne 2
.na
\fBevent ID\fR
.ad
.RS 28n
The ID of the contract event.Use \fBct_event_get_evid\fR(3CONTRACT).
.RE

.sp
.ne 2
.na
\fBflags\fR
.ad
.RS 28n
A bit vector possibly including \fBCT_ACK\fR and \fBCTE_INFO\fR. Use
\fBct_event_get_flags\fR(3CONTRACT) to obtain this information.
.RE

.sp
.ne 2
.na
\fBevent type\fR
.ad
.RS 28n
The type of event, equal to one of the constants specified in the contract
type's manual page or \fBCT_EV_NEGEND\fR. Use
\fBct_event_get_type\fR(3CONTRACT) to obtain this information.
.RE

.SS "EVENT TYPES"
.LP
The following event types are defined:
.sp
.ne 2
.na
\fB\fBCT_EV_NEGEND\fR\fR
.ad
.RS 16n
Some time after an exit negotiation is initiated, the \fBCT_EV_NEGEND\fR event
is sent. This indicates that the negotiation ended. This might be because the
operation was cancelled, or because the operation was successful. If
successful, and the owner requested that a new contract be written, this
contains the ID of that contract.
.sp
\fBCT_EV_NEGEND\fR cannot be included in a contract's informative or critical
event set. It is always delivered and always critical. If \fBCT_EV_NEGEND\fR
indicates that the operation was successful, no further events are sent. The
contract's owner should use \fBct_ctl_abandon\fR(3CONTRACT) to abandon the
contract.
.sp
A \fBCT_EV_NEGEND\fR event contains:
.sp
```



```
.ne 2
.na
\fBnegotiation ID\fR
.ad
.RS 19n
The ID of the negotiation which ended. Use \fBct_event_get_nevid\fR(3CONTRACT)
to obain this information.
.RE

.sp
.ne 2
.na
\fBnew contract ID\fR
.ad
.RS 19n
The ID of the newly created contract. This value is 0 if no contract was
created, or the ID of the existing contract if the operation was not completed.
Use \fBct_event_get_newct\fR(3CONTRACT) to obtain this information.
.RE

.RE

.SH FILES
.ne 2
.na
\fB\fB/system/contract\fR\fR
.ad
.sp .6
.RS 4n
List of all contract types
.RE

.sp
.ne 2
.na
\fB\fB/system/contract/all\fR\fR
.ad
.sp .6
.RS 4n
Directory of all contract IDs
.RE

.sp
.ne 2
.na
\fB\fB/system/contract/all/\fIid\fR\fR\fR
.ad
.sp .6
.RS 4n
Symbolic link to the type-specific directory of contract \fIid\fR
.RE

.sp
.ne 2
.na
\fB\fB/system/contract/\fItype\fR\fR\fR
.ad
.sp .6
.RS 4n
Specific type directory
.RE

.sp
.ne 2
.na
\fB\fB/system/contract/\fItype\fR/template\fR\fR
```

```
523 \fB\fB/system/contract/\fItype\fR/templete\fR\fR
524 .ad
525 .sp .6
526 .RS 4n
527 Template for the contract type
528 .RE

530 .sp
531 .ne 2
532 .na
533 \fB\fB/system/contract/\fItype\fR/bundle\fR\fR
534 .ad
535 .sp .6
536 .RS 4n
537 Listening point for all contracts of that type
538 .RE

540 .sp
541 .ne 2
542 .na
543 \fB\fB/system/contract/\fItype\fR/pbundle\fR\fR
544 .ad
545 .sp .6
546 .RS 4n
547 Listening point for all contracts of that type for the opening process
548 .RE

550 .sp
551 .ne 2
552 .na
553 \fB\fB/system/contract/\fItype\fR /latest\fR\fR
554 .ad
555 .sp .6
556 .RS 4n
557 Status of most recent \fItype\fR contract created by the opening LWP
558 .RE

560 .sp
561 .ne 2
562 .na
563 \fB\fB/system/contract/\fItype\fR/\fIID\fR\fR\fR
564 .ad
565 .sp .6
566 .RS 4n
567 Directory for contract id
568 .RE

570 .sp
571 .ne 2
572 .na
573 \fB\fB/system/contract/\fItype\fR/\fIID\fR/events\fR\fR
574 .ad
575 .sp .6
576 .RS 4n
577 Listening point for contract id's events
578 .RE

580 .sp
581 .ne 2
582 .na
583 \fB\fB/system/contract/\fItype\fR/\fIID\fR/ctl\fR\fR
584 .ad
585 .sp .6
586 .RS 4n
587 Control file for contract ID
588 .RE
```



```
590 .sp
591 .ne 2
592 .na
593 \fB\fB/system/contract/\fItype\fR/\fIID\fR/status\fR\fR
594 .ad
595 .sp .6
596 .RS 4n
597 Status info for contract ID
598 .RE

600 .SH SEE ALSO
601 .LP
602 \fBctrun\fR(1), \fBctstat\fR(1), \fBctwatch\fR(1), \fBchroot\fR(1M),
603 \fBclose\fR(2), \fBioctl\fR(2), \fBopen\fR(2), \fBpoll\fR(2),
604 \fBct_ctl_abandon\fR(3CONTRACT), \fBct_event_read\fR(3CONTRACT),
605 \fBct_event_get_evid\fR(3CONTRACT), \fBct_event_get_flags\fR(3CONTRACT),
606 \fBct_event_get_nevid\fR(3CONTRACT), \fBct_event_get_newct\fR(3CONTRACT),
607 \fBct_event_get_type\fR(3CONTRACT),
608 \fBct_status_read\fR(3CONTRACT), \fBct_status_get_cookie\fR(3CONTRACT),
609 \fBct_status_get_critical\fR(3CONTRACT), \fBct_status_get_holder\fR(3CONTRACT),
610 \fBct_status_get_id\fR(3CONTRACT), \fBct_status_get_informative\fR(3CONTRACT),
611 \fBct_status_get_nevid\fR(3CONTRACT), \fBct_status_get_nevents\fR(3CONTRACT),
612 \fBct_status_get_ntime\fR(3CONTRACT), \fBct_status_get_qtime\fR(3CONTRACT),
613 \fBct_status_get_state\fR(3CONTRACT), \fBct_status_get_type\fR(3CONTRACT),
614 \fBct_tmpl_set_cookie\fR(3CONTRACT), \fBct_tmpl_set_critical\fR(3CONTRACT),
615 \fBct_tmpl_set_informative\fR(3CONTRACT), \fBlibcontract\fR(3LIB),
616 \fBprocess\fR(4), \fBlargefile\fR(5), \fBlfcompile\fR(5), \fBprivileges\fR(5)
```

```
**********************************************************
    8608 Sun Sep 16 19:22:56 2018
new/usr/src/man/man4/driver.conf.4
9842 man page typos and spelling
**********************************************************
```

```
'\" te
.\"  Copyright (c) 2005, Sun Microsystems, Inc. All Rights Reserved
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH DRIVER.CONF 4 "Sep 16, 2018"
.TH DRIVER.CONF 4 "Jan 5, 2007"
.SH NAME
driver.conf \- driver configuration files
.SH SYNOPSIS
.LP
.nf
\fBdriver.conf\fR
.fi

.SH DESCRIPTION
.sp
.LP
Driver configuration files provide values for device properties. The values
override values provided by the devices themselves. Most modern devices provide
enough property values to make a driver configuration file unnecessary.
.sp
.LP
The system associates a driver with its configuration file by name. For
example, a driver in \fB/usr/kernel/drv\fR called \fBwombat\fR has the driver
configuration file \fBwombat.conf\fR, also stored in \fB/usr/kernel/drv\fR,
associated with it. On systems that support 64-bit drivers, the driver
associated with it. On systems capable of support 64-bit drivers, the driver
configuration file should be placed in the directory in which the 32-bit driver
is (or would be) located, even if only a 64-bit version is provided. For
example, a 64-bit driver stored in \fB/usr/kernel/drv/sparcv9\fR stores its
driver configuration file in \fB/usr/kernel/drv\fR.
.sp
.LP
The value of the \fBname\fR property is the node name. In a \fBdriver.conf\fR
file, where the generic node name and \fBcompatible\fR property associated with
a self-identifying devices are typically not used, the node name must be a
binding name. The binding name is the name chosen by the system to bind a
driver to the device. The binding name is either an alias associated with the
driver established by \fBadd_drv\fR(1M) or the driver name itself.
.sp
.LP
The syntax of a single entry in a driver configuration file takes one of three
forms:
.sp
.in +2
.nf
\fBname\fR="\fInode name\fR" \fBparent\fR="\fIparent name\fR" [\fIproperty-name=
.fi
.in -2

.sp
.LP
In this form, the parent name can be either the binding name of the parent
nexus driver or a specific full pathname, beginning with a slash (\fB/\fR)
character, identifying a specific instance of a parent bus. If a binding name
is used then all parent nodes bound to that driver match. A generic name (for
example, \fBpci\fR) is not a valid binding name even though it can appear in
the full pathname of all intended parents.
.sp
.LP
```



```
Alternatively, the parent can be specified by the type of interface it presents
to its children.
.sp
.in +2
.nf
\fBname\fR="\fInode name\fR" \fBclass\fR="\fIclass name\fR" [\fIproperty-name=va
.fi
.in -2

.sp
.LP
For example, the driver for the \fBSCSI\fR host adapter can have different
names on different platforms, but the target drivers can use class \fBscsi\fR
to insulate themselves from these differences.
.sp
.LP
Entries of either form above correspond to a device information (\fBdevinfo\fR)
node in the kernel device tree. Each node has a \fIname\fR which is usually the
name of the driver, and a \fIparent\fR name which is the name of the parent
\fBdevinfo\fR node to which it will be connected. Any number of name-value
pairs can be specified to create properties on the prototype \fBdevinfo\fR
node. These properties can be retrieved using the DDI property interfaces (for
example, \fBddi_prop_get_int\fR(9F) and \fBddi_prop_lookup\fR(9F)). The
prototype \fBdevinfo\fR node specification must be terminated with a semicolon
(\fB;\fR).
.sp
.LP
The third form of an entry is simply a list of properties.
.sp
.in +2
.nf
[\fIproperty-name=value\fR ...]\fB;\fR
.fi
.in -2
.sp

.sp
.LP
A property created in this way is treated as global to the driver. It can be
overridden by a property with the same name on a particular \fBdevinfo\fR node,
either by creating one explicitly on the prototype node in the driver.conf file
or by the driver.
.sp
.LP
Items are separated by any number of newlines, \fBSPACE\fR or \fBTAB\fR
characters.
.sp
.LP
The configuration file can contain several entries to specify different device
configurations and parent nodes. The system can call the driver for each
possible prototype \fBdevinfo\fR node, and it is generally the responsibility
of the drivers \fBprobe\fR(9E) routine to determine if the hardware described
by the prototype \fBdevinfo\fR node is really present.
.sp
.LP
Property names must not violate the naming conventions for Open Boot PROM
properties or for IEEE 1275 names. In particular, property names should contain
only printable characters, and should not contain at-sign (\fB@\fR), slash
(\fB/\fR), backslash (\fB\e\fR), colon (\fB:\fR), or square brackets
(\fB[]\fR). Property values can be decimal integers or strings delimited by
double quotes (\fB"\fR). Hexadecimal integers can be constructed by prefixing
the digits with \fB0x\fR.
.sp
.LP
A comma separated list of integers can be used to construct properties whose
value is an integer array. The value of such properties can be retrieved inside
```

```
125 the driver using \fBddi_prop_lookup_int_array\fR(9F).
126 .sp
127 .LP
128 Comments are specified by placing a \fB#\fR character at the beginning of the
129 comment string, the comment string extends for the rest of the line.
130 .SH EXAMPLES
131 .LP
132 \fBExample 1 \fRConfiguration File for a PCI Bus Frame Buffer
133 .sp
134 .LP
135 The following is an example of a configuration file called
136 \fBACME,simple.conf\fR for a \fBPCI\fR bus frame buffer called
137 \fBACME,simple\fR.

139 .sp
140 .in +2
141 .nf
142 #
143 # Copyright (c) 1993, by ACME Fictitious Devices, Inc.
144 #
145 #ident  "@(#)ACME,simple.conf   1.3     1999/09/09"

147 name="ACME,simple" class="pci" unit-address="3,1"
148         debug-mode=12;
149 .fi
150 .in -2

152 .sp
153 .LP
154 This example creates a prototype \fBdevinfo\fR node called \fBACME,simple\fR
155 under all parent nodes of class \fBpci\fR. The node has device and function
156 numbers of 3 and 1, respectively; the property \fBdebug-mode\fR is provided for
157 all instances of the driver.

159 .LP
160 \fBExample 2 \fRConfiguration File for a Pseudo Device Driver
161 .sp
162 .LP
163 The following is an example of a configuration file called
164 \fBACME,example.conf\fR for a pseudo device driver called \fBACME,example\fR.

166 .sp
167 .in +2
168 .nf
169 #
170 # Copyright (c) 1993, ACME Fictitious Devices, Inc.
171 #
172 #ident  "@(#)ACME,example.conf  1.2   93/09/09"
173 name="ACME,example" parent="pseudo" instance=0
174     debug-level=1;

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

178 whizzy-mode="on";
179 debug-level=3;
180 .fi
181 .in -2

183 .sp
184 .LP
185 This creates two \fBdevinfo\fR nodes called \fBACME,example\fR which attaches
186 below the \fBpseudo\fR node in the kernel device tree. The \fBinstance\fR
187 property is only interpreted by the \fBpseudo\fR node, see \fBpseudo\fR(4) for
188 further details. A property called \fBdebug-level\fR is created on the first
189 \fBdevinfo\fR node which has the value 1. The \fBexample\fR driver is able to
190 fetch the value of this property using \fBddi_prop_get_int\fR(9F).
```



```
192 .sp
193 .LP
194 Two global driver properties are created, \fBwhizzy-mode\fR (which has the
195 string value "on") and \fBdebug-level\fR (which has the value 3). If the driver
196 looks up the property \fBwhizzy-mode\fR on either node, it retrieves the value
197 of the global \fBwhizzy-mode\fR property ("on"). If the driver looks up the
198 \fBdebug-level\fR property on the first node, it retrieves the value of the
199 \fBdebug-level\fR property on that node (1). Looking up the same property on
200 the second node retrieves the value of the global \fBdebug-level\fR property
201 (3).

203 .SH SEE ALSO
205 .sp
204 .LP
205 \fBadd_drv\fR(1M), \fBpci\fR(4), \fBpseudo\fR(4), \fBsbus\fR(4), \fBscsi\fR(4),
206 \fBprobe\fR(9E), \fBddi_getlongprop\fR(9F), \fBddi_getprop\fR(9F),
207 \fBddi_getproplen\fR(9F), \fBddi_prop_get_int\fR(9F),
208 \fBddi_prop_lookup\fR(9F), \fBddi_prop_op\fR(9F)
209 .sp
210 .LP
211 \fIWriting Device Drivers\fR
212 .SH WARNINGS
215 .sp
213 .LP
214 To avoid namespace collisions between multiple driver vendors, it is strongly
215 recommended that the \fIname\fR property of the driver should begin with a
216 vendor-unique string. A reasonably compact and unique choice is the vendor
217 over-the-counter stock symbol.
218 .SH NOTES
222 .sp
219 .LP
220 The \fBupdate_drv\fR(1M) command should be used to prompt the kernel to reread
221 \fBdriver.conf\fR files.
225 \fIdriver.conf\fR files. Using \fBmodunload\fR(1M) to update \fIdriver.conf\fR
226 continues to work in release 9 of the Solaris operating environment, but the
227 behavior will change in a future release.
```

```
**********************************************************
   26071 Sun Sep 16 19:22:56 2018
new/usr/src/man/man4/ike.config.4
9842 man page typos and spelling
**********************************************************
   1 '\" te
   2 .\" Copyright (c) 2009, Sun Microsystems, Inc. All Rights Reserved.
   3 .\" Copyright (c) 2015, Circonus, Inc. All Rights Reserved.
   4 .\" The contents of this file are subject to the terms of the Common Development
   5 .\"  See the License for the specific language governing permissions and limitat
   6 .\" fields enclosed by brackets "[]" replaced with your own identifying informat
   7 .TH IKE.CONFIG 4 "Apr 27, 2009"
   8 .SH NAME
   9 ike.config \- configuration file for IKE policy
  10 .SH SYNOPSIS
  11 .LP
  12 .nf
  13 \fB/etc/inet/ike/config\fR
  14 .fi

  16 .SH DESCRIPTION
  17 .LP
  18 The \fB/etc/inet/ike/config\fR file contains rules for matching inbound IKE
  19 requests. It also contains rules for preparing outbound \fBIKE\fR requests.
  20 .sp
  21 .LP
  22 You can test the syntactic correctness of an \fB/etc/inet/ike/config\fR file by
  23 using the \fB-c\fR or \fB-f\fR options of \fBin.iked\fR(1M). You must use the
  24 \fB-c\fR option to test a \fBconfig\fR file. You might need to use the \fB-f\fR
  25 option if it is not in \fB/etc/inet/ike/config\fR.
  26 .SS "Lexical Components"
  27 .LP
  28 On any line, an unquoted \fB#\fR character introduces a comment. The remainder
  29 of that line is ignored. Additionally, on any line, an unquoted \fB//\fR
  30 sequence introduces a comment. The remainder of that line is ignored.
  31 .sp
  32 .LP
  33 There are several types of lexical tokens in the \fBike.config\fR file:
  34 .sp
  35 .ne 2
  36 .na
  37 \fB\fInum\fR\fR
  38 .ad
  39 .sp .6
  40 .RS 4n
  41 A decimal, hex, or octal number representation is as in 'C'.
  42 .RE

  44 .sp
  45 .ne 2
  46 .na
  47 \fB\fIIPaddr\fR/\fIprefix\fR/\fIrange\fR\fR
  48 .ad
  49 .sp .6
  50 .RS 4n
  51 An IPv4 or IPv6 address with an optional /\fINNN\fR suffix, (where \fINNN\fR is
  52 a \fInum\fR) that indicates an address (\fBCIDR\fR) prefix (for example,
  53 \fB10.1.2.0/24\fR). An optional /\fIADDR\fR suffix (where \fIADDR\fR is a
  54 second IP address) indicates an address/mask pair (for example,
  55 \fB10.1.2.0/255.255.255.0\fR). An optional -\fIADDR\fR suffix (where \fIADDR\fR
  56 is a second IPv4 address) indicates an inclusive range of addresses (for
  57 example, \fB10.1.2.0-10.1.2.255\fR). The \fB/\fR or \fB-\fR can be surrounded
  58 by an arbitrary amount of white space.
  59 .RE

  61 .sp
```



```
  62 .ne 2
  63 .na
  64 \fB\fBXXX\fR | \fBYYY\fR | \fBZZZ\fR\fR
  65 .ad
  66 .sp .6
  67 .RS 4n
  68 Either the words \fBXX\fRX, \fBYYY\fR, or \fBZZZ\fR, for example, {yes,no}.
  69 .RE

  71 .sp
  72 .ne 2
  73 .na
  74 \fBp1-id-type\fR
  75 .ad
  76 .sp .6
  77 .RS 4n
  78 An IKE phase 1 identity type. IKE phase 1 identity types include:
  79 .br
  80 .in +2
  81 \fBdn, DN\fR
  82 .in -2
  83 .br
  84 .in +2
  85 \fBdns, DNS\fR
  86 .in -2
  87 .br
  88 .in +2
  89 \fBfqdn, FQDN\fR
  90 .in -2
  91 .br
  92 .in +2
  93 \fBgn, GN\fR
  94 .in -2
  95 .br
  96 .in +2
  97 \fBip, IP\fR
  98 .in -2
  99 .br
 100 .in +2
 101 \fBipv4\fR
 102 .in -2
 103 .br
 104 .in +2
 105 \fBipv4_prefix\fR
 106 .in -2
 107 .br
 108 .in +2
 109 \fBipv4_range\fR
 110 .in -2
 111 .br
 112 .in +2
 113 \fBipv6\fR
 114 .in -2
 115 .br
 116 .in +2
 117 \fBipv6_prefix\fR
 118 .in -2
 119 .br
 120 .in +2
 121 \fBipv6_range\fR
 122 .in -2
 123 .br
 124 .in +2
 125 \fBmbox, MBOX\fR
 126 .in -2
 127 .br
```

```
.in +2
\fBuser_fqdn\fR
.in -2
.RE

.sp
.ne 2
.na
\fB\fB"\fR\fIstring\fR\fB"\fR\fR
.ad
.sp .6
.RS 4n
A quoted string.
.sp
Examples include:\fB"Label foo"\fR, or \fB"C=US, OU=Sun Microsystems\e, Inc.,
N=olemcd@eng.example.com"\fR
.sp
A backslash (\fB\e\fR) is an escape character. If the string needs an actual
backslash, two must be specified.
.RE

.sp
.ne 2
.na
\fB\fIcert-sel\fR\fR
.ad
.sp .6
.RS 4n
A certificate selector, a \fIstring\fR which specifies the identities of zero
or more certificates. The specifiers can conform to \fBX.509\fR naming
conventions.
.sp
A \fIcert-sel\fR can also use various shortcuts to match either subject
alternative names, the filename or \fBslot\fR of a certificate in
\fB/etc/inet/ike/publickeys\fR, or even the \fBISSUER\fR. For example:
.sp
.in +2
.nf
"SLOT=0"
"EMAIL=postmaster@domain.org"
"webmaster@domain.org" # Some just work w/o TYPE=
"IP=10.0.0.1"
"10.21.11.11"          # Some just work w/o TYPE=
"DNS=www.domain.org"
"mailhost.domain.org"  # Some just work w/o TYPE=
"ISSUER=C=US, O=Sun Microsystems\\, Inc., CN=Sun CA"
.fi
.in -2
.sp

Any \fIcert-sel\fR preceded by the character \fB!\fR indicates a negative
match, that is, not matching this specifier. These are the same kind of strings
used in \fBikecert\fR(1M).
.RE

.sp
.ne 2
.na
\fB\fIldap-list\fR\fR
.ad
.sp .6
.RS 4n
A quoted, comma-separated list of LDAP servers and ports.
.sp
For example, \fB"ldap1.example.com"\fR, \fB"ldap1.example.com:389"\fR,
\fB"ldap1.example.com:389,ldap2.example.com"\fR.
```



```
.sp
The default port for LDAP is \fB389\fR.
.RE

.sp
.ne 2
.na
\fB\fIparameter-list\fR\fR
.ad
.sp .6
.RS 4n
A list of parameters.
.RE

.SS "File Body Entries"
.LP
There are four main types of entries:
.RS +4
.TP
.ie t \(bu
.el o
global parameters
.RE
.RS +4
.TP
.ie t \(bu
.el o
IKE phase 1 transform defaults
.RE
.RS +4
.TP
.ie t \(bu
.el o
IKE rule defaults
.RE
.RS +4
.TP
.ie t \(bu
.el o
IKE rules
.RE
.sp
.LP
The global parameter entries are as follows:
.sp
.ne 2
.na
\fBcert_root \fIcert-sel\fR\fR
.ad
.sp .6
.RS 4n
The X.509 distinguished name of a certificate that is a trusted root CA
certificate.It must be encoded in a file in the \fB/etc/inet/ike/publickeys\fR
directory. It must have a CRL in \fB/etc/inet/ike/crl\fRs. Multiple
\fBcert_root\fR parameters aggregate.
.RE

.sp
.ne 2
.na
\fBcert_trust \fIcert-sel\fR\fR
.ad
.sp .6
.RS 4n
Specifies an X.509 distinguished name of a certificate that is self-signed, or
has otherwise been verified as trustworthy for signing IKE exchanges. It must
```

```
260 be encoded in a file in \fB/etc/inet/ike/publickeys\fR. Multiple
261 \fBcert_trust\fR parameters aggregate.
262 .RE

264 .sp
265 .ne 2
266 .na
267 \fBexpire_timer \fIinteger\fR\fR
268 .ad
269 .sp .6
270 .RS 4n
271 The number of seconds to let a not-yet-complete IKE Phase I (Main Mode)
272 negotiation linger before deleting it. Default value: 300 seconds.
273 .RE

275 .sp
276 .ne 2
277 .na
278 \fBignore_crls\fR
279 .ad
280 .sp .6
281 .RS 4n
282 If this keyword is present in the file, \fBin.iked\fR(1M) ignores Certificate
283 Revocation Lists (\fBCRL\fRs) for root \fBCA\fRs (as given in \fBcert_root\fR)
284 .RE

286 .sp
287 .ne 2
288 .na
289 \fBldap_server \fIldap-list\fR\fR
290 .ad
291 .sp .6
292 .RS 4n
293 A list of LDAP servers to query for certificates. The list can be additive.
294 .RE

296 .sp
297 .ne 2
298 .na
299 \fBpkcs11_path \fIstring\fR\fR
300 .ad
301 .sp .6
302 .RS 4n
303 The string that follows is a name of a shared object (\fB\&.so\fR) that
304 implements the PKCS#11 standard. The name is passed directly into
305 \fBdlopen\fR(3C) for linking, with all of the semantics of that library call.
306 By default, \fBin.iked\fR(1M) runs the same ISA as the running kernel, so a
307 library specified using \fBpkcs11_path\fR and an absolute pathname \fBmust\fR
308 match the same ISA as the kernel. One can use the start/exec SMF property (see
309 \fBsvccfg\fR(1M)) to change \fBin.iked\fR's ISA, but it is not recommended.
310 .sp
311 If this setting is not present, the default value is set to \fBlibpkcs11.so\fR.
312 Most cryptographic providers go through the default library, and this parameter
313 should only be used if a specialized provider of IKE-useful cryptographic
314 services cannot interface with the Solaris Cryptographic Framework. See
315 \fBcryptoadm\fR(1M).
316 .sp
317 This option is now deprecated, and may be removed in a future release.
318 .RE

320 .sp
321 .ne 2
322 .na
323 \fBretry_limit \fIinteger\fR\fR
324 .ad
325 .sp .6
```



```
326 .RS 4n
327 The number of retransmits before any IKE negotiation is aborted. Default value:
328 5 times.
329 .RE

331 .sp
332 .ne 2
333 .na
334 \fBretry_timer_init \fIinteger\fR or \fIfloat\fR\fR
335 .ad
336 .sp .6
337 .RS 4n
338 The initial interval (in seconds) between retransmits. This interval is doubled
339 until the \fBretry_timer_max\fR value (see below) is reached. Default value:
340 0.5 seconds.
341 .RE

343 .sp
344 .ne 2
345 .na
346 \fBretry_timer_max \fIinteger\fR or \fIfloat\fR\fR
347 .ad
348 .sp .6
349 .RS 4n
350 The maximum interval (in seconds) between retransmits. The doubling retransmit
351 interval stops growing at this limit. Default value: 30 seconds.
352 .LP
353 Note -
354 .sp
355 .RS 2
356 This value is never reached with the default configuration. The longest
357 interval is 8 (0.5 * 2 ^ (5 - 1)) seconds.
358 .RE
359 .RE

361 .sp
362 .ne 2
363 .na
364 \fBproxy \fIstring\fR\fR
365 .ad
366 .sp .6
367 .RS 4n
368 The string following this keyword must be a URL for an HTTP proxy, for example,
369 \fBhttp://proxy:8080\fR.
370 .RE

372 .sp
373 .ne 2
374 .na
375 \fBsocks \fIstring\fR\fR
376 .ad
377 .sp .6
378 .RS 4n
379 The string following this keyword must be a URL for a SOCKS proxy, for example,
380 \fBsocks://socks-proxy\fR.
381 .RE

383 .sp
384 .ne 2
385 .na
386 \fBuse_http\fR
387 .ad
388 .sp .6
389 .RS 4n
390 If this keyword is present in the file, \fBin.iked\fR(1M) uses HTTP to retrieve
391 Certificate Revocation Lists (\fBCRL\fRs).
```

```
392 .RE

394 .sp
395 .LP
396 The following IKE phase 1 transform parameters can be prefigured using
397 file-level defaults. Values specified within any given transform override these
398 defaults.
399 .sp
400 .LP
401 The IKE phase 1 transform defaults are as follows:
402 .sp
403 .ne 2
404 .na
405 \fBp1_lifetime_secs \fInum\fR\fR
406 .ad
407 .sp .6
408 .RS 4n
409 The proposed default lifetime, in seconds, of an IKE phase 1 security
410 association (\fBSA\fR).
411 .RE

413 .sp
414 .ne 2
415 .na
416 \fBp1_nonce_len \fInum\fR\fR
417 .ad
418 .sp .6
419 .RS 4n
420 The length in bytes of the phase 1 (quick mode) nonce data. This cannot be
421 specified on a per-rule basis.
422 .RE

424 .sp
425 .LP
426 The following IKE rule parameters can be prefigured using file-level defaults.
427 Values specified within any given rule override these defaults, unless a rule
428 cannot.
429 .sp
430 .ne 2
431 .na
432 \fBp2_lifetime_secs \fInum\fR\fR
433 .ad
434 .sp .6
435 .RS 4n
436 The proposed default lifetime, in seconds, of an IKE phase 2 security
437 association (SA). This value is optional. If omitted, a default value is used.
438 .RE

440 .sp
441 .ne 2
442 .na
443 \fBp2_softlife_secs \fInum\fR\fR
444 .ad
445 .sp .6
446 .RS 4n
447 The soft lifetime of a phase 2 SA, in seconds. If this value is specified, the
448 SA soft expires after the number of seconds specified by
449 \fBp2_softlife_secs\fR. This causes \fBin.iked\fR to renegotiate a new phase 2
450 SA before the original SA expires.
451 .sp
452 This value is optional, if omitted soft expiry occurs after 90% of the lifetime
453 specified by \fBp2_lifetime_secs\fR. The value specified by
454 \fBp2_softlife_secs\fR is ignored if \fBp2_lifetime_secs\fR is not specified.
455 .sp
456 Setting \fBp2_softlife_secs\fR to the same value as \fBp2_lifetime_secs\fR
457 disables soft expires.
```



```
458 .RE

460 .sp
461 .ne 2
462 .na
463 \fBp2_idletime_secs \fInum\fR\fR
464 .ad
465 .sp .6
466 .RS 4n
467 The idle lifetime of a phase 2 SA, in seconds. If the value is specified, the
468 value specifies the lifetime of the SA, if the security association is not used
469 before the SA is revalidated.
470 .RE

472 .sp
473 .ne 2
474 .na
475 \fBp2_lifetime_kb \fInum\fR\fR
476 .ad
477 .sp .6
478 .RS 4n
479 The lifetime of an SA can optionally be specified in kilobytes. This parameter
480 specifies the default value. If lifetimes are specified in both seconds and
481 kilobytes, the SA expires when either the seconds or kilobyte thresholds are
481 kilobytes, the SA expires when either the seconds or kilobyte threshholds are
482 passed.
483 .RE

485 .sp
486 .ne 2
487 .na
488 \fBp2_softlife_kb \fInum\fR\fR
489 .ad
490 .sp .6
491 .RS 4n
492 This value is the number of kilobytes that can be protected by an SA before a
493 soft expire occurs (see \fBp2_softlife_secs\fR, above).
494 .sp
495 This value is optional. If omitted, soft expiry occurs after 90% of the
496 lifetime specified by \fBp2_lifetime_kb\fR. The value specified by
497 \fBp2_softlife_kb\fR is ignored if \fBp2_lifetime_kb\fR is not specified.
498 .RE

500 .sp
501 .ne 2
502 .na
503 \fBp2_nonce_len \fInum\fR\fR
504 .ad
505 .sp .6
506 .RS 4n
507 The length in bytes of the phase 2 (quick mode) nonce data. This cannot be
508 specified on a per-rule basis.
509 .RE

511 .sp
512 .ne 2
513 .na
514 \fBlocal_id_type \fIp1-id-type\fR\fR
515 .ad
516 .sp .6
517 .RS 4n
518 The local identity for IKE requires a type. This identity type is reflected in
519 the IKE exchange. The type can be one of the following:
520 .RS +4
521 .TP
522 .ie t \(bu
```

```
.el o
an IP address (for example, \fB10.1.1.2\fR)
.RE
.RS +4
.TP
.ie t \(bu
.el o
DNS name (for example, \fBtest.domain.com\fR)
.RE
.RS +4
.TP
.ie t \(bu
.el o
MBOX RFC 822 name (for example, \fBroot@domain.com\fR)
.RE
.RS +4
.TP
.ie t \(bu
.el o
DNX.509 distinguished name (for example, \fBC=US, O=Sun Microsystems\, Inc.,
CN=Sun Test cert\fR)
.RE
.RE

.sp
.ne 2
.na
\fBp1_xform ’{’ parameter-list ’}\fR
.ad
.sp .6
.RS 4n
A phase 1 transform specifies a method for protecting an IKE phase 1 exchange.
An initiator offers up lists of phase 1 transforms, and a receiver is expected
to only accept such an entry if it matches one in a phase 1 rule. There can be
several of these, and they are additive. There must be either at least one
phase 1 transform in a rule or a global default phase 1 transform list. In a
configuration file without a global default phase 1 transform list \fBand\fR a
rule without a phase, transform list is an invalid file. Unless specified as
optional, elements in the parameter-list must occur exactly once within a given
transform’s parameter-list:
.sp
.ne 2
.na
\fBoakley_group \fInumber\fR\fR
.ad
.sp .6
.RS 4n
The Oakley Diffie-Hellman group used for IKE SA key derivation. The group
numbers are defined in RFC 2409, Appendix A, RFC 3526, and RFC 5114, section
3.2. Acceptable values are currently:
.br
.in +2
1 (MODP 768-bit)
.in -2
.br
.in +2
2 (MODP 1024-bit)
.in -2
.br
.in +2
3 (EC2N 155-bit)
.in -2
.br
.in +2
4 (EC2N 185-bit)
.in -2
```



```
.br
.in +2
5 (MODP 1536-bit)
.in -2
.br
.in +2
14 (MODP 2048-bit)
.in -2
.br
.in +2
15 (MODP 3072-bit)
.in -2
.br
.in +2
16 (MODP 4096-bit)
.in -2
.br
.in +2
17 (MODP 6144-bit)
.in -2
.br
.in +2
18 (MODP 8192-bit)
.in -2
.br
.in +2
19 (ECP 256-bit)
.in -2
.br
.in +2
20 (ECP 384-bit)
.in -2
.br
.in +2
21 (ECP 521-bit)
.in -2
.br
.in +2
22 (MODP 1024-bit, with 160-bit Prime Order Subgroup)
.in -2
.br
.in +2
23 (MODP 2048-bit, with 224-bit Prime Order Subgroup)
.in -2
.br
.in +2
24 (MODP 2048-bit, with 256-bit Prime Order Subgroup)
.in -2
.br
.in +2
25 (ECP 192-bit)
.in -2
.br
.in +2
26 (ECP 224-bit)
.in -2
.RE

.sp
.ne 2
.na
\fBencr_alg {3des, 3des-cbc, blowfish, blowfish-cdc, des, des-cbc, aes,
aes-cbc}\fR
.ad
.sp .6
.RS 4n
```

```
655 An encryption algorithm, as in \fBipsecconf\fR(1M). However, of the ciphers
656 listed above, only \fBaes\fR and \fBaes-cbc\fR allow optional key-size setting,
657 using the "low value-to-high value" syntax. To specify a single AES key size,
658 the low value must equal the high value. If no range is specified, all three
659 AES key sizes are allowed.
660 .RE

662 .sp
663 .ne 2
664 .na
665 \fBauth_alg {md5, sha, sha1, sha256, sha384, sha512}\fR
666 .ad
667 .sp .6
668 .RS 4n
669 An authentication algorithm.
670 .sp
671 Use \fBipsecalgs\fR(1M) with the \fB-l\fR option to list the IPsec protocols
672 and algorithms currently defined on a system. The \fBcryptoadm list\fR command
673 displays a list of installed providers and their mechanisms. See
673 diplays a list of installed providers and their mechanisms. See
674 \fBcryptoadm\fR(1M).
675 .RE

677 .sp
678 .ne 2
679 .na
680 \fBauth_method {preshared, rsa_sig, rsa_encrypt, dss_sig}\fR
681 .ad
682 .sp .6
683 .RS 4n
684 The authentication method used for IKE phase 1.
685 .RE

687 .sp
688 .ne 2
689 .na
690 \fBp1_lifetime_secs \fInum\fR\fR
691 .ad
692 .sp .6
693 .RS 4n
694 Optional. The lifetime for a phase 1 SA.
695 .RE

697 .RE

699 .sp
700 .ne 2
701 .na
702 \fBp2_lifetime_secs \fInum\fR\fR
703 .ad
704 .sp .6
705 .RS 4n
706 If configuring the kernel defaults is not sufficient for different tasks, this
707 parameter can be used on a per-rule basis to set the IPsec \fBSA\fR lifetimes
708 in seconds.
709 .RE

711 .sp
712 .ne 2
713 .na
714 \fBp2_pfs \fInum\fR\fR
715 .ad
716 .sp .6
717 .RS 4n
718 Use perfect forward secrecy for phase 2 (quick mode). If selected, the oakley
719 group specified is used for phase 2 PFS. Acceptable values are:
```



```
720 .br
721 .in +2
722 0 (do not use Perfect Forward Secrecy for IPsec SAs)
723 .in -2
724 .br
725 .in +2
726 1 (768-bit)
727 .in -2
728 .br
729 .in +2
730 2 (1024-bit)
731 .in -2
732 .br
733 .in +2
734 5 (1536-bit)
735 .in -2
736 .br
737 .in +2
738 14 (2048-bit)
739 .in -2
740 .br
741 .in +2
742 15 (3072-bit)
743 .in -2
744 .br
745 .in +2
746 16 (4096-bit)
747 .in -2
748 .RE

750 .sp
751 .LP
752 An IKE rule starts with a right-curly-brace (\fB{\fR), ends with a
753 left-curly-brace (\fB}\fR), and has the following parameters in between:
754 .sp
755 .ne 2
756 .na
757 \fBlabel \fIstring\fR\fR
758 .ad
759 .sp .6
760 .RS 4n
761 Required parameter. The administrative interface to \fBin.iked\fR looks up
762 phase 1 policy rules with the label as the search string. The administrative
763 interface also converts the label into an index, suitable for an extended
764 ACQUIRE message from PF_KEY - effectively tying IPsec policy to IKE policy in
765 the case of a node initiating traffic. Only one \fBlabel\fR parameter is
766 allowed per rule.
767 .RE

769 .sp
770 .ne 2
771 .na
772 \fBlocal_addr <\fIIPaddr\fR/\fIprefix\fR/\fIrange\fR>\fR
773 .ad
774 .sp .6
775 .RS 4n
776 Required parameter. The local address, address prefix, or address range for
777 this phase 1 rule. Multiple \fBlocal_addr\fR parameters accumulate within a
778 given rule.
779 .RE

781 .sp
782 .ne 2
783 .na
784 \fBremote_addr <\fIIPaddr\fR/\fIprefix\fR/\fIrang\fRe>\fR
785 .ad
```

```
786 .sp .6
787 .RS 4n
788 Required parameter. The remote address, address prefix, or address range for
789 this phase 1 rule. Multiple \fBremote_addr\fR parameters accumulate within a
790 given rule.
791 .RE

793 .sp
794 .ne 2
795 .na
796 \fBlocal_id_type \fIp1-id-type\fR\fR
797 .ad
798 .sp .6
799 .RS 4n
800 Which phase 1 identity type I uses. This is needed because a single certificate
801 can contain multiple values for use in IKE phase 1. Within a given rule, all
802 phase 1 transforms must either use preshared or non-preshared authentication
803 (they cannot be mixed). For rules with preshared authentication, the
804 \fBlocal_id_type\fR parameter is optional, and defaults to \fBIP\fR. For rules
805 which use non-preshared authentication, the ’local_id_type’ parameter is
806 required. Multiple ’local_id_type’ parameters within a rule are not allowed.
807 .RE

809 .sp
810 .ne 2
811 .na
812 \fBlocal_id \fIcert-sel\fR\fR
813 .ad
814 .sp .6
815 .RS 4n
816 Disallowed for preshared authentication method; required parameter for
817 non-preshared authentication method. The local identity string or certificate
818 selector. Only one local identity per rule is used, the first one stated.
819 .RE

821 .sp
822 .ne 2
823 .na
824 \fBremote_id \fIcert-sel\fR\fR
825 .ad
826 .sp .6
827 .RS 4n
828 Disallowed for preshared authentication method; required parameter for
829 non-preshared authentication method. Selector for which remote phase 1
830 identities are allowed by this rule. Multiple \fBremote_id\fR parameters
831 accumulate within a given rule. If a single empty string (\fB""\fR) is given,
832 then this accepts any remote \fBID\fR for phase 1. It is recommended that
833 certificate trust chains or address enforcement be configured strictly to
834 prevent a breakdown in security if this value for \fBremote_id\fR is used.
835 .RE

837 .sp
838 .ne 2
839 .na
840 \fBp2_lifetime_secs \fInum\fR\fR
841 .ad
842 .sp .6
843 .RS 4n
844 If configuring the kernel defaults is not sufficient for different tasks, this
845 parameter can be used on a per-rule basis to set the IPsec \fBSA\fR lifetimes
846 in seconds.
847 .RE

849 .sp
850 .ne 2
851 .na
```



```
852 \fBp2_pfs \fInum\fR\fR
853 .ad
854 .sp .6
855 .RS 4n
856 Use perfect forward secrecy for phase 2 (quick mode). If selected, the oakley
857 group specified is used for phase 2 PFS. Acceptable values are:
858 .br
859 .in +2
860 0 (do not use Perfect Forward Secrecy for IPsec SAs)
861 .in -2
862 .br
863 .in +2
864 1 (768-bit)
865 .in -2
866 .br
867 .in +2
868 2 (1024-bit)
869 .in -2
870 .br
871 .in +2
872 5 (1536-bit)
873 .in -2
874 .br
875 .in +2
876 14 (2048-bit)
877 .in -2
878 .br
879 .in +2
880 15 (3072-bit)
881 .in -2
882 .br
883 .in +2
884 16 (4096-bit)
885 .in -2
886 .RE

888 .sp
889 .ne 2
890 .na
891 \fBp1_xform \fB{\fR \fIparameter-list\fR \fB}\fR\fR
892 .ad
893 .sp .6
894 .RS 4n
895 A phase 1 transform specifies a method for protecting an IKE phase 1 exchange.
896 An initiator offers up lists of phase 1 transforms, and a receiver is expected
897 to only accept such an entry if it matches one in a phase 1 rule. There can be
898 several of these, and they are additive. There must be either at least one
899 phase 1 transform in a rule or a global default phase 1 transform list. A
900 \fBike.config\fR file without a global default phase 1transform list \fBand\fR
901 a rule without a phase 1 transform list is an invalid file. Elements within the
902 parameter-list; unless specified as optional, must occur exactly once within a
903 given transform’s parameter-list:
904 .sp
905 .ne 2
906 .na
907 \fBoakley_group \fInumber\fR\fR
908 .ad
909 .sp .6
910 .RS 4n
911 The Oakley Diffie-Hellman group used for \fBIKE SA\fR key derivation.
912 Acceptable values are currently:
913 .br
914 .in +2
915 1 (768-bit)
916 .in -2
917 .br
```

```
 918 .in +2
 919 2 (1024-bit)
 920 .in -2
 921 .br
 922 .in +2
 923 5 (1536-bit)
 924 .in -2
 925 .br
 926 .in +2
 927 14 (2048-bit)
 928 .in -2
 929 .br
 930 .in +2
 931 15 (3072-bit)
 932 .in -2
 933 .br
 934 .in +2
 935 16 (4096-bit)
 936 .in -2
 937 .RE

 939 .sp
 940 .ne 2
 941 .na
 942 \fBencr_alg {3des, 3des-cbc, blowfish, blowfish-cdc, des, des-cbc, aes,
 943 aes-cbc}\fR
 944 .ad
 945 .sp .6
 946 .RS 4n
 947 An encryption algorithm, as in \fBipsecconf\fR(1M). However, of the ciphers
 948 listed above, only \fBaes\fR and \fBaes-cbc\fR allow optional key-size setting,
 949 using the "low value-to-high value" syntax. To specify a single AES key size,
 950 the low value must equal the high value. If no range is specified, all three
 951 AES key sizes are allowed.
 952 .RE

 954 .sp
 955 .ne 2
 956 .na
 957 \fBauth_alg {md5, sha, sha1}\fR
 958 .ad
 959 .sp .6
 960 .RS 4n
 961 An authentication algorithm, as specified in \fBipseckey\fR(1M).
 962 .RE

 964 .sp
 965 .ne 2
 966 .na
 967 \fBauth_method {preshared, rsa_sig, rsa_encrypt, dss_sig}\fR
 968 .ad
 969 .sp .6
 970 .RS 4n
 971 The authentication method used for IKE phase 1.
 972 .RE

 974 .sp
 975 .ne 2
 976 .na
 977 \fBp1_lifetime_secs \fInum\fR\fR
 978 .ad
 979 .sp .6
 980 .RS 4n
 981 Optional. The lifetime for a phase 1 SA.
 982 .RE
```



```
 984 .RE

 986 .SH EXAMPLES
 987 .LP
 988 \fBExample 1 \fRA Sample \fBike.config\fR File
 989 .sp
 990 .LP
 991 The following is an example of an \fBike.config\fR file:

 993 .sp
 994 .in +2
 995 .nf

 997 ### BEGINNING OF FILE

 999 ### First some global parameters...

1001 ### certificate parameters...

1003 # Root certificates. I SHOULD use a full Distinguished Name.
1004 # I must have this certificate in my local filesystem, see ikecert(1m).
1005 cert_root    "C=US, O=Sun Microsystems\\, Inc., CN=Sun CA"

1007 # Explicitly trusted certs that need no signatures, or perhaps
1008 # self-signed ones. Like root certificates, use full DNs for them
1009 # for now.
1010 cert_trust    "EMAIL=root@domain.org"

1012 # Where do I send LDAP requests?
1013 ldap_server        "ldap1.domain.org,ldap2.domain.org:389"

1015 ## phase 1 transform defaults...

1017 p1_lifetime_secs 14400
1018 p1_nonce_len 20

1020 ## Parameters that might also show up in rules.

1022 p1_xform { auth_method preshared oakley_group 5 auth_alg sha
1023          encr_alg 3des }
1024 p2_pfs 2



1028 ### Now some rules...

1030 {
1031    label "simple inheritor"
1032    local_id_type ip
1033    local_addr 10.1.1.1
1034    remote_addr 10.1.1.2
1035 }
______unchanged_portion_omitted_
```

```
**********************************************************
    8502 Sun Sep 16 19:22:56 2018
new/usr/src/man/man4/ldapsearchprefs.conf.4
9842 man page typos and spelling
**********************************************************
'\" te
.\" Copyright (C) 1990, Regents of the University of Michigan.  All Rights Reser
.\" Portions Copyright (C) 1997, Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH LDAPSEARCHPREFS.CONF 4 "Nov 26, 2017"
.SH NAME
ldapsearchprefs.conf \- configuration file for LDAP search preference routines
.SH SYNOPSIS
.LP
.nf
\fB/etc/opt/SUNWconn/ldap/current/ldapsearchprefs.conf\fR
.fi

.SH DESCRIPTION
.LP
The \fBldapsearchprefs.conf\fR file contains information used by LDAP when
searching the directory. Blank lines and lines that start with a hash ('#')
character are treated as comments and ignored. Non-comment lines contain one or
more tokens. Tokens are separated by white space, and double quotes can be used
to include white space inside a token.
.sp
.LP
Search preferences are typically used by LDAP-based client programs to specify
what a user may search for, which attributes are searched, and which options
are available to the user.
.sp
.LP
The first non-comment line specifies the version of the template information
The first non-commment line specifies the version of the template information
and must contain the token \fBVersion\fR followed by an integer version number.
For example:
.sp
.in +2
.nf
Version 1
.fi
.in -2
.sp

.sp
.LP
The current version is \fI1,\fR so the above example is always the correct
opening line.
.sp
.LP
The remainder of the file consists of one or more search preference
configurations. The first line of a search preference is a human-readable name
for the type of object being searched for, for example \fBPeople\fR or
\fBOrganizations\fR. This name is stored in the \fIso_objtypeprompt\fR member
of the \fBldap_searchobj\fR structure (see \fBldap_searchprefs\fR(3LDAP)). For
example:
.sp
.in +2
.nf
People
.fi
.in -2
.sp
```



```
.sp
.LP
specifies a label for a search preference designed to find X.500 entries for
people.
.sp
.LP
The next line specifies a list of options for this search object. The only
option currently allowed is "internal" which means that this search object
should not be presented directly to a user. Options are placed in the
\fIso_options\fR member of the \fIldap_searchobj\fR structure and can be tested
using the \fBLDAP_IS_SEARCHOBJ_OPTION_SET()\fR macro. Use "" if no special
options are required.
.sp
.LP
The next line specifies a label to use for "Fewer Choices" searches. "Fewer
Choices" searches are those where the user's input is fed to the ldap_filter
routines to determine an appropriate filter to use. This contrasts with
explicitly-constructed LDAP filters, or "More Choices" searches, where the user
can explicitly construct an LDAP filter.
.sp
.LP
For example:
.sp
.in +2
.nf
"Search For:"
.fi
.in -2
.sp

.sp
.LP
can be used by LDAP client programs to label the field into which the user can
type a "Fewer Choices" search.
.sp
.LP
The next line specifies an LDAP filter prefix to append to all "More Choices"
searched. This is typically used to limit the types of entries returned to
those containing a specific object class. For example:
.sp
.in +2
.nf
"(&(objectClass=person)"
.fi
.in -2
.sp

.sp
.LP
would cause only entries containing the object class \fIperson\fR to be
returned by a search. Note that parentheses may be unbalanced here, since this
is a filter prefix, not an entire filter.
.sp
.LP
The next line is an LDAP filter tag which specifies the set of LDAP filters to
be applied for "Fewer Choices" searching. The line
.sp
.in +2
.nf
\fB"x500-People"\fR
.fi
.in -2
.sp

.sp
.LP
```

would tell the client program to use the set of LDAP filters from the ldap filter configuration file tagged "x500-People".

.LP

The next line specifies an LDAP attribute to retrieve to help the user choose when several entries match the search terms specified. For example:

```
.sp
.in +2
.nf
"title"
.fi
.in -2
.sp

.sp
.LP
specifies that if more than one entry matches the search criteria, the client program should retrieve the \fBtitle\fR attribute that and present that to the user to allow them to select the appropriate entry. The next line specifies a label for the above attribute, for example,
.sp
.in +2
.nf
"Title:"
.fi
.in -2
.sp

.sp
.LP
Note that the values defined so far in the file are defaults, and are intended to be overridden by the specific search options that follow.
.sp
.LP
The next line specifies the scope of the LDAP search to be performed. Acceptable values are subtree, onelevel, and base.
.sp
.LP
The next section is a list of "More Choices" search options, terminated by a line containing only the string \fBEND\fR. For example:
.sp
.in +2
.nf
"Common Name"   cn      11111   ""      ""
"Surname"       sn      11111   ""      ""
"Business Phone"        "telephoneNumber"       11101   ""      ""
END
.fi
.in -2
.sp

.sp
.LP
 Each line represents one method of searching. In this example, there are three ways of searching - by Common Name, by Surname, and by Business Phone number. The first field is the text which should be displayed to user. The second field is the attribute which will be searched. The third field is a bitmap which specifies which of the match types are permitted for this search type. A "1" value in a given bit position indicates that a particular match type is valid, and a "0" indicates that is it not valid. The fourth and fifth fields are, respectively, the select attribute name and on-screen name for the selected attribute. These values are intended to override the defaults defined above. If no specific values are specified, the client software uses the default values above.
.sp
.LP
```



```
The next section is a list of search match options, terminated by a a line containing only the string \fBEND\fR. Example:
.sp
.in +2
.nf
"exactly matches"       "(%a=%v))"
"approximately matches" "(%a~=%v))"
"starts with"   "(%a=%v*))"
"ends with"     "(%a=*%v))"
"contains"      "(%a=*%v*))"
END
.fi
.in -2
.sp

.sp
.LP
In this example, there are five ways of refining the search. For each method, there is an LDAP filter suffix which is appended to the ldap filter.
.SH EXAMPLES
.LP
\fBExample 1 \fRA Sample Configuration Using Search Preference for "people"
.sp
.LP
The following example illustrates one possible configuration of search preferences for "people".

.sp
.in +2
.nf
# Version number
Version 1
# Name for this search object
People
# Label to place before text box user types in
"Search For:"
# Filter prefix to append to all "More Choices" searches
"(&(objectClass=person)"
# Tag to use for "Fewer Choices" searches - from ldapfilter.conf file
"x500-People"
# If a search results in > 1 match, retrieve this attribute to help
# user distinguish between the entries...
multilineDescription
# ...and label it with this string:
"Description"
# Search scope to use when searching
subtree
# Follows a list of "More Choices" search options. Format is:
# Label, attribute, select-bitmap, extra attr display name, extra attr ldap name
# If last two are null, "Fewer Choices" name/attributes used
"Common Name"                   cn                11111  ""  ""
"Surname"                       sn                11111  ""  ""
"Business Phone"                "telephoneNumber"  11101  ""  ""
"E-Mail Address"                "mail"            11111  ""  ""
"Uniqname"                      "uid"             11111  ""  ""
END
# Match types
"exactly matches"               "(%a=%v))"
"approximately matches"         "(%a~=%v))"
"starts with"                   "(%a=%v*))"
"ends with"                     "(%a=*%v))"
"contains"                      "(%a=*%v*))"
END
.fi
.in -2
```

```
 259 .sp
 260 .LP
 261 In this example, the user may search for People. For "fewer choices" searching,
 262 the tag for the \fBldapfilter.conf\fR(4) file is "x500-People".
 263 .SH ATTRIBUTES
 264 .LP
 265 See \fBattributes\fR(5) for a description of the following attributes:
 266 .sp

 268 .sp
 269 .TS
 270 box;
 271 c | c
 272 l | l .
 273 ATTRIBUTE TYPE  ATTRIBUTE VALUE
 274 Stability Level Evolving
 275 .TE

 277 .SH SEE ALSO
 278 .LP
 279 \fBldap_searchprefs\fR(3LDAP), \fBattributes\fR(5)
```

```
**********************************************************
    8770 Sun Sep 16 19:22:56 2018
new/usr/src/man/man4/ldaptemplates.conf.4
9842 man page typos and spelling
**********************************************************
```

```
'\" te
.\" Copyright (C) 1990, Regents of the University of Michigan.  All Rights Reser
.\" Portions Copyright (C) 1997, Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH LDAPTEMPLATES.CONF 4 "Jul 9, 2003"
.SH NAME
ldaptemplates.conf \- configuration file for LDAP display template routines
.SH SYNOPSIS
.LP
.nf
\fB/etc/opt/SUNWconn/ldap/current/ldaptemplates.conf\fR
.fi

.SH DESCRIPTION
.sp
.LP
The \fBldaptemplates.conf\fR file contains information used by the LDAP display
routines.
.sp
.LP
Blank lines and lines that start with a hash character ('#') are treated as
comments and ignored. Non-comment lines contain one or more tokens. Tokens are
separated by white space, and double quotes can be used to include white space
inside a token.
.sp
.LP
The first non-comment line specifies the version of the template information
The first non-commment line specifies the version of the template information
and must contain the token \fBVersion\fR followed by an integer version number.
For example,
.sp
.in +2
.nf
Version 1
.fi
.in -2
.sp

.sp
.LP
The current version is \fI1\fR, so the above example is always the correct
first line.
.sp
.LP
The remainder of the file consists of one or more display templates. The first
two lines of the display template each contain a single token that specifies
singular and plural names for the template in a user-friendly format. For
example,
.sp
.in +2
.nf
"Person"
"People"
.fi
.in -2
.sp

.sp
.LP
```



```
specifies appropriate names for a template designed to display person
information.
.sp
.LP
The next line specifies the name of the icon or similar element that is
associated with this template. For example,
.sp
.in +2
.nf
"person icon"
.fi
.in -2
.sp

.sp
.LP
The next line is a blank-separated list of template options. "" can be used if
no options are desired. Available options are: \fBaddable\fR (it is appropriate
to allow entries of this type to be added), \fBmodrdn\fR (it is appropriate to
offer the \fBmodify rdn\fR operation), \fBaltview\fR (this template is an
alternate view of another template). For example,
.sp
.in +2
.nf
"addable" "modrdn"
.fi
.in -2
.sp

.sp
.LP
The next portion of the template is a list of X.500 object classes that is used
to determine whether the template should be used to display a given entry. The
object class information consists of one or more lines, followed by a
terminating line that contains the single token \fBEND\fR. Each line contains
one or more object class names, all of which must be present in a directory
entry. Multiple lines can be used to associate more than one set of object
classes with a given template. For example,
.sp
.in +2
.nf
emailPerson
orgPerson
END
.fi
.in -2
.sp

.sp
.LP
means that the template is appropriate for display of \fBemailPerson\fR entries
or \fBorgPerson\fR entries.
.sp
.LP
The next line after the object class list is the name of the attribute to
authenticate as to make changes (use "" if it is appropriate to authenticate as
the entry itself). For example,
.sp
.in +2
.nf
"owner"
.fi
.in -2
.sp

.sp
```

```
126 .LP
127 The next line is the default attribute to use when naming a new entry, for
128 example,
129 .sp
130 .in +2
131 .nf
132 "cn"
133 .fi
134 .in -2
135 .sp

137 .sp
138 .LP
139 The next line is the distinguished name of the default location under which new
140 entries are created. For example,
141 .sp
142 .in +2
143 .nf
144 "o=XYZ, c=US"
145 .fi
146 .in -2
147 .sp

149 .sp
150 .LP
151 The next section is a list of rules used to assign default values to new
152 entries. The list should be terminated with a line that contains the single
153 token \fBEND\fR. Each line in this section should either begin with the token
154 \fBconstant\fR and be followed by the name of the attribute and a constant
155 value to assign, or the line should begin with \fBaddersdn\fR followed by the
156 name of an attribute whose value will be the DN of the person who has
157 authenticated to add the entry. For example,
158 .sp
159 .in +2
160 .nf
161 constant        associatedDomain        XYZ.us
162 addersdn        seeAlso
163 END
164 .fi
165 .in -2
166 .sp

168 .sp
169 .LP
170 The last portion of the template is a list of items to display. It consists of
171 one or more lines, followed by a terminating line that contains the single
172 token \fBEND\fR. Each line is must begin with the token \fBsamerow\fR or the
173 token \fBitem\fR
174 .sp
175 .LP
176 It is assumed that each item appears on a row by itself unless it was preceded
177 by a \fBsamerow\fR line (in which case it should be displayed on the same line
178 as the previous item, if possible). Lines that begin with \fBsamerow\fR should
179 not have any other tokens on them.
180 .sp
181 .LP
182 Lines that begin with \fBitem\fR must have at least three more tokens on them:
183 an item type, a label, and an attribute name. Any extra tokens are taken as
184 extra arguments.
185 .sp
186 .LP
187 The item type token must be one of the following strings:
188 .sp
189 .ne 2
190 .na
191 \fB\fBcis\fR \fR
```



```
192 .ad
193 .RS 14n
194 case-ignore string attributes
195 .RE

197 .sp
198 .ne 2
199 .na
200 \fB\fBmls\fR \fR
201 .ad
202 .RS 14n
203 multiline string attributes
204 .RE

206 .sp
207 .ne 2
208 .na
209 \fB\fBmail\fR \fR
210 .ad
211 .RS 14n
212 RFC-822 conformant mail address attributes
213 .RE

215 .sp
216 .ne 2
217 .na
218 \fB\fBdn\fR \fR
219 .ad
220 .RS 14n
221 distinguished name pointer attributes
222 .RE

224 .sp
225 .ne 2
226 .na
227 \fB\fBbool\fR \fR
228 .ad
229 .RS 14n
230 Boolean attributes
231 .RE

233 .sp
234 .ne 2
235 .na
236 \fB\fBjpeg\fR \fR
237 .ad
238 .RS 14n
239 JPEG photo attributes
240 .RE

242 .sp
243 .ne 2
244 .na
245 \fB\fBjpegbtn\fR \fR
246 .ad
247 .RS 14n
248 a button that will retrieve and show a JPEG photo attribute
249 .RE

251 .sp
252 .ne 2
253 .na
254 \fB\fBfax\fR \fR
255 .ad
256 .RS 14n
257 FAX T.4 format image attributes
```

```
.RE

.sp
.ne 2
.na
\fB\fBfaxbtn\fR \fR
.ad
.RS 14n
a button that will retrieve and show a FAX photo attribute
.RE

.sp
.ne 2
.na
\fB\fBaudiobtn\fR \fR
.ad
.RS 14n
audio attributes
.RE

.sp
.ne 2
.na
\fB\fBtime\fR \fR
.ad
.RS 14n
UTC time attributes
.RE

.sp
.ne 2
.na
\fB\fBdate\fR \fR
.ad
.RS 14n
UTC time attributes where only the date portion should be shown
.RE

.sp
.ne 2
.na
\fB\fBurl\fR \fR
.ad
.RS 14n
labeled Uniform Resource Locator attributes
.RE

.sp
.ne 2
.na
\fB\fBsearchact\fR \fR
.ad
.RS 14n
define an action that will do a directory search for other entries
.RE

.sp
.ne 2
.na
\fB\fBlinkact\fR \fR
.ad
.RS 14n
define an action which is a link to another display template
.RE

.sp
```



```
.ne 2
.na
\fB\fBprotected\fR \fR
.ad
.RS 14n
for an encrypted attribute, with values displayed as asterisks
.RE

.sp
.LP
An example of an item line for the drink attribute (displayed with label "Work
Phone"):
.sp
.in +2
.nf
item cis        "Work Phone"    telephoneNumber
.fi
.in -2
.sp

.SH EXAMPLES
.LP
\fBExample 1 \fRA Sample Configuration File Containing a Template that Displays
People Entries
.sp
.LP
The following template configuration file contains a templates for display of
people entries.

.sp
.in +2
.nf
    #
    # LDAP display templates
    #
    # Version must be 1 for now
    #
    Version 1
    #
    # Person template
    "Person"
    "People"

    # name of the icon that is associated with this template
    "person icon"

    # blank-separated list of template options ("" for none)
    "addable"

    #
    # objectclass list
    person
    END

    #
    # name of attribute to authenticate as ("" means auth as this entry)
    ""

    #
    # default attribute name to use when forming RDN of a new entry
    #
    "cn"

    #
    # default location when adding new entries (DN; "" means no default)
    "o=XYZ, c=US"
```

```
391     #
392     # rules used to define default values for new entries
393     END

395     #
396     # list of items for display
397     item jpegbtn        "View Photo"            jpegPhoto       "Next Photo"
398     item audiobtn       "Play Sound"            audio
399     item cis    "Also Known As"         cn
400     item cis    "Title"                 title
401     item mls    "Work Address"          postalAddress
402     item cis    "Work Phone"            telephoneNumber
403     item cis    "Fax Number"            facsimileTelephoneNumber
404     item mls    "Home Address"          homePostalAddress
405     item cis    "Home Phone"            homePhone
406     item cis    "User ID"               uid
407     item mail   "E-Mail Address"        mail
408     item cis    "Description"           description
409     item dn             "See Also"              seeAlso
410     END
411 .fi
412 .in -2
413 .sp

415 .SH ATTRIBUTES
417 .sp
416 .LP
417 See \fBattributes\fR(5) for a description of the following attributes:
418 .sp

420 .sp
421 .TS
422 box;
423 c | c
424 l | l .
425 ATTRIBUTE TYPE  ATTRIBUTE VALUE
426 Stability Level Evolving
427 .TE

429 .SH SEE ALSO
432 .sp
430 .LP
431 \fBldap_disptmpl\fR(3LDAP), \fBldap_entry2text\fR(3LDAP), \fBattributes\fR(5)
```

```
**********************************************************
    2774 Sun Sep 16 19:22:56 2018
new/usr/src/man/man4/mpapi.conf.4
9842 man page typos and spelling
**********************************************************
'\" te
.\" Copyright (c) 2004-2006 Storage Networking Industry Association. All Rights
.\" Portions Copyright (c) 2007, Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\"  See the License for the specific language governing permissions and limitat
.\" the fields enclosed by brackets "[]" replaced with your own identifying info
.TH MPAPI.CONF 4 "Sep 16, 2018"
.TH MPAPI.CONF 4 "Mar 15, 2006"
.SH NAME
mpapi.conf \- configuration file for libMPAPI
.SH SYNOPSIS
.LP
.nf
/etc/mpapi.conf
.fi

.SH DESCRIPTION
.LP
The \fB/etc/mpapi.conf\fR file is used to specify the vendor-provided plugin
library that is installed on the system. This file is used by the
\fBlibMPAPI\fR(3LIB) common library to load the individual plugin library when
its interface is called. If changes are made to the file while the library is
in use, the library should be unloaded and reloaded. Addition and removal of
the plugin library should be handled through \fBMP_RegisterPlugin\fR(3MPAPI)
and \fBMP_DeregisterPlugin\fR(3MPAPI).
.sp
.LP
Each plugin library entry is a single line of the form:
.sp
.in +2
.nf
"id"        "library file name"
.fi
.in -2
.sp

.sp
.LP
where
.sp
.ne 2
.na
\fB\fBid\fR\fR
.ad
.RS 21n
The identification of the library.  It is the reversed domain name of the vendor
The identification of library.  It is the resersed domain name of the vendor
followed by \fB\&.\fR followed by the vendor specific name of the plugin that
uniquely identifies the plugin library.
.RE

.sp
.ne 2
.na
\fB\fBlibrary file name\fR\fR
.ad
.RS 21n
The absolute path to the shared object library file.
The  shared  object library file in the absolute path format.
.RE
```



```
.SH EXAMPLES
.LP
\fBExample 1 \fRExample of an \fB/etc/mpapi.conf\fR file
.sp
.in +2
.nf
# This file contains names and references to MP API plugin libraries
#
#  Do NOT manually edit this file
#
# Format:
#
# <library ID>  <library file name>
#
com.sun.mpapi32         /lib/libmpscsi_vhci.so
com.sun.mpapi64         /lib/64/libmpscsi_vhci.so
.fi
.in -2

.SH ATTRIBUTES
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
Interface Stability     T{
Standard: ANSI INCITS 412 Multipath Management API
T}
.TE

.SH SEE ALSO
.LP
\fBlibMPAPI\fR(3LIB), \fBMP_DeregisterPlugin\fR(3MPAPI),
\fBMP_RegisterPlugin\fR(3MPAPI), \fBattributes\fR(5)
```

```
**********************************************************
    7200 Sun Sep 16 19:22:56 2018
new/usr/src/man/man4/sbus.4
9842 man page typos and spelling
**********************************************************
'\" te
.\" Copyright (c) 1999, Sun Microsystems, Inc.
.\" All Rights Reserved
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH SBUS 4 "Dec 31, 1996"
.SH NAME
sbus \- configuration files for SBus device drivers
.SH DESCRIPTION
.sp
.LP
The \fBSBus\fR is a geographically addressed peripheral bus present on many
\fBSPARC\fR hardware platforms. \fBSBus\fR devices are \fIself-identifying\fR
\(em that is to say the \fBSBus\fR card itself provides information to the
system so that it can identify the device driver that needs to be used. The
device usually provides additional information to the system in the form of
name-value pairs that can be retrieved using the \fBDDI\fR property interfaces.
See \fBddi_prop_op\fR(9F) for details.
.sp
.LP
The information is usually derived from a small Forth program stored in the
\fBFCode\fR \fBPROM\fR on the card, so driver configuration files should be
completely unnecessary for these devices. However, on some occasions, drivers
for \fBSBus\fR devices may need to use driver configuration files to augment
the information provided by the \fBSBus\fR card. See \fBdriver.conf\fR(4) for
further details.
.sp
.LP
When they are needed, configuration files for \fBSBus\fR device drivers should
identify the parent bus driver implicitly using the \fIclass\fR keyword.  This
removes the dependency on the particular bus driver involved since this may be
named differently on different platforms.
.sp
.LP
All bus drivers of class \fBsbus\fR recognise the following properties:
.sp
.ne 2
.na
\fB\fBreg\fR\fR
.ad
.RS 14n
An arbitrary length array where each element of the array consists of a 3-tuple
of integers.  Each array element describes a logically contiguous mappable
resource on the \fBSBus.\fR
.sp
The first integer of each tuple specifies the slot number the card is plugged
into. The second integer of each 3-tuple specifies the offset in the slot
address space identified by the first element. The third integer of each
3-tuple specifies the size in bytes of the mappable resource.
.sp
The driver can refer to the elements of this array by index, and construct
kernel mappings to these addresses using \fBddi_map_regs\fR(9F). The index into
the array is passed as the \fIrnumber\fR argument of \fBddi_map_regs()\fR.
.sp
You can use the \fBddi_get*\fR and \fBddi_put*\fR family of functions to access
register space from a high-level interrupt context.
.RE

.sp
.ne 2
```



```
.na
\fB\fBinterrupts\fR\fR
.ad
.RS 14n
An arbitrary length array where each element of the array consists of a single
integer. Each array element describes a possible \fBSBus\fR interrupt level
that the device might generate.
.sp
The driver can refer to the elements of this array by index, and register
interrupt handlers with the system using \fBddi_add_intr\fR(9F). The index into
the array is passed as the \fIinumber\fR argument of \fBddi_add_intr()\fR.
.RE

.sp
.ne 2
.na
\fB\fBregisters\fR\fR
.ad
.RS 14n
An arbitrary length array where each element of the array consists of a 3-tuple
of integers.  Each array element describes a logically contiguous mappable
resource on the \fBSBus.\fR
.sp
The first integer of each tuple should be set to \fB\(mi1\fR, specifying that
any SBus slot may be matched. The second integer of each 3-tuple specifies the
offset in the slot address space identified by the first element.  The third
integer of each 3-tuple specifies the size in bytes of the mappable resource.
integer of each 3-tuple specifies the size in bytes of the mappable resoure.
.sp
The \fBregisters\fR property can only be used to augment an incompletely
specified \fBreg\fR property with information from a driver configuration file.
It may only be specified in a driver configuration file.
.RE

.sp
.LP
All \fBSBus\fR devices must provide \fBreg\fR properties to the system. The
first two integer elements of the \fBreg\fR property are used to construct the
address part of the device name under \fB/devices\fR.
.sp
.LP
Only devices that generate interrupts need to provide \fBinterrupts\fR
properties.
.sp
.LP
Occasionally, it may be necessary to override or augment the configuration
information supplied by the \fBSBus\fR device. This can be achieved by writing
a driver configuration file that describes a prototype device information
(devinfo) node specification, containing the additional properties required.
.sp
.LP
For the system to merge the information, certain conditions must be met. First,
the \fBname\fR property must be the same. Second, either the first two integers
(slot number and offset) of the two \fBreg\fR properties must be the same, or
the second integer (offset) of the \fBreg\fR and \fBregisters\fR properties
must be the same.
.sp
.LP
In the event that the \fBSBus\fR card has no \fBreg\fR property at all, the
self-identifying information cannot be used, so all the details of the card
must be specified in a driver configuration file.
.SH EXAMPLES
.LP
\fBExample 1 \fRA sample configuration file.
.sp
.LP
```

```
126 Here is a configuration file for an \fBSBus\fR card called \fBSUNW,netboard\fR.
127 The card already has a simple \fBFCode\fR \fBPROM\fR that creates \fBname\fR
128 and \fBreg\fR properties, and will have a complete set of properties for normal
129 use once the driver and firmware is complete.

131 .sp
132 .LP
133 In this example, we want to augment the properties given to us by the firmware.
134 We use the same \fBname\fR property, and use the \fBregisters\fR property to
135 match the firmware \fBreg\fR property. That way we don't have to worry about
136 which slot the card is really plugged into.

138 .sp
139 .LP
140 We want to add an \fBinterrupts\fR property while we are developing the
141 firmware and driver so that we can start to experiment with interrupts. The
142 device can generate interrupts at \fBSBus\fR level 3. Additionally, we want to
143 set a \fBdebug-level\fR property to 4.

145 .sp
146 .in +2
147 .nf
148 #
149 # Copyright (c) 1992, by Sun Microsystems, Inc.
150 #ident  "@(#)SUNW,netboard.conf         1.4     92/03/10 SMI"
151 #
152 name="SUNW,netboard" class="sbus"
153         registers=-1,0x40000,64,-1,0x80000,1024
154         interrupts=3 debug-level=4;
155 .fi
156 .in -2
157 .sp

159 .SH ATTRIBUTES
161 .sp
160 .LP
161 See \fBattributes\fR(5) for descriptions of the following attributes:
162 .sp

164 .sp
165 .TS
166 box;
167 c | c
168 l | l .
169 ATTRIBUTE TYPE  ATTRIBUTE VALUE
170 _
171 Architecture    SPARC
172 .TE

174 .SH SEE ALSO
177 .sp
175 .LP
176 \fBdriver.conf\fR(4), \fBattributes\fR(5), \fBddi_add_intr\fR(9F),
177 \fBddi_map_regs\fR(9F), \fBddi_prop_op\fR(9F)
178 .sp
179 .LP
180 \fIWriting Device Drivers\fR
181 .SH WARNINGS
185 .sp
182 .LP
183 The wildcarding mechanism of the \fBregisters\fR property matches every
184 instance of the particular device attached to the system. This may not always
185 be what is wanted.
```

```
**********************************************************
    5370 Sun Sep 16 19:22:57 2018
new/usr/src/man/man4/warn.conf.4
9842 man page typos and spelling
**********************************************************
   1 '\" te
   2 .\" Copyright 1987, 1989 by the Student Information Processing Board of the Mass
   3 .\" Portions Copyright (c) 2004, Sun Microsystems, Inc.  All Rights Reserved
   4 .\" The contents of this file are subject to the terms of the Common Development
   5 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
   6 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
   7 .TH WARN.CONF 4 "Mar 30, 2005"
   8 .SH NAME
   9 warn.conf \- Kerberos warning configuration file
  10 .SH SYNOPSIS
  11 .LP
  12 .nf
  13 /etc/krb5/warn.conf
  14 .fi

  16 .SH DESCRIPTION
  17 .sp
  17 .LP
  18 The \fBwarn.conf\fR file contains configuration information specifying how
  19 users will be warned by the \fBktkt_warnd\fR daemon about ticket expiration. In
  20 addition, this file can be used to auto-renew the user's Ticket-Granting Ticket
  21 (TGT) instead of warning the user. Credential expiration warnings and
  22 auto-renew results are sent, by means of syslog, to \fBauth.notice\fR.
  23 .sp
  24 .LP
  25 Each Kerberos client host must have a \fBwarn.conf\fR file in order for users
  26 on that host to get Kerberos warnings from the client. Entries in the
  27 \fBwarn.conf\fR file must have the following format:
  28 .sp
  29 .in +2
  30 .nf
  31 \fIprincipal\fR [renew[:\fIopt1\fR,...\fIoptN\fR]] syslog|terminal \fItime\fR
  32 .fi
  33 .in -2

  35 .sp
  36 .LP
  37 or:
  38 .sp
  39 .in +2
  40 .nf
  41 \fIprincipal\fR [renew[:\fIopt1\fR,...\fIoptN\fR]] mail \fItime\fR [\fIemail add
  42 .fi
  43 .in -2

  45 .sp
  46 .ne 2
  47 .na
  48 \fB\fIprincipal\fR\fR
  49 .ad
  50 .RS 17n
  51 Specifies the principal name to be warned. The asterisk (\fB*\fR) wildcard can
  52 be used to specify groups of principals.
  53 .RE

  55 .sp
  56 .ne 2
  57 .na
  58 \fB\fBrenew\fR\fR
  59 .ad
  60 .RS 17n
```



```
  61 Automatically renew the credentials (TGT) until renewable lifetime expires.
  62 This is equivalent to the user running \fBkinit\fR \fB-R\fR.
  63 .sp
  64 The renew options include:
  65 .sp
  66 .ne 2
  67 .na
  68 \fB\fBlog-success\fR\fR
  69 .ad
  70 .RS 15n
  71 Log the result of the renew attempt on success using the specified method
  72 (\fBsyslog\fR|\fBterminal\fR|\fBmail\fR).
  73 .RE

  75 .sp
  76 .ne 2
  77 .na
  78 \fB\fBlog-failure\fR\fR
  79 .ad
  80 .RS 15n
  81 Log the result of the renew attempt on failure using the specified method
  82 (\fBsyslog\fR|\fBterminal\fR|\fBmail\fR). Some renew failure conditions are:
  83 TGT renewable lifetime has expired, the KDCs are unavailable, or the cred cache
  84 file has been removed.
  85 .RE

  87 .sp
  88 .ne 2
  89 .na
  90 \fB\fBlog\fR\fR
  91 .ad
  92 .RS 15n
  93 Same as specifying both \fBlog-success\fR and \fBlog-failure\fR.
  94 Same as specifing both \fBlog-success\fR and \fBlog-failure\fR.
  94 .RE

  96 .LP
  97 Note -
  98 .sp
  99 .RS 2
 100 If no log options are given, no logging is done.
 101 .RE
 102 .RE

 104 .sp
 105 .ne 2
 106 .na
 107 \fB\fBsyslog\fR\fR
 108 .ad
 109 .RS 17n
 110 Sends the warnings to the system's syslog. Depending on the
 111 \fB/etc/syslog.conf\fR file, syslog entries are written to the
 112 \fB/var/adm/messages\fR file and/or displayed on the terminal.
 113 .RE

 115 .sp
 116 .ne 2
 117 .na
 118 \fB\fBterminal\fR\fR
 119 .ad
 120 .RS 17n
 121 Sends the warnings to display on the terminal.
 122 .RE

 124 .sp
 125 .ne 2
```

```
126 .na
127 \fB\fBmail\fR\fR
128 .ad
129 .RS 17n
130 Sends the warnings as email to the address specified by \fIemail_address\fR.
131 .RE

133 .sp
134 .ne 2
135 .na
136 \fB\fItime\fR\fR
137 .ad
138 .RS 17n
139 Specifies how much time before the \fBTGT\fR expires when a warning should be
140 sent. The default time value is seconds, but you can specify \fBh\fR (hours)
141 and \fBm\fR (minutes) after the number to specify other time values.
142 .RE

144 .sp
145 .ne 2
146 .na
147 \fB\fIemail_address\fR\fR
148 .ad
149 .RS 17n
150 Specifies the email address at which to send the warnings. This field must be
151 specified only with the \fBmail\fR field.
152 .RE

154 .SH EXAMPLES
155 .LP
156 \fBExample 1 \fRSpecifying Warnings
157 .sp
158 .LP
159 The following \fBwarn.conf\fR entry

161 .sp
162 .in +2
163 .nf
164 \fB* syslog 5m\fR
165 .fi
166 .in -2
167 .sp

169 .sp
170 .LP
171 specifies that warnings will be sent to the syslog five minutes before the
172 expiration of the \fBTGT\fR for all principals. The form of the message is:

174 .sp
175 .in +2
176 .nf
177 jdb@ACME.COM: your kerberos credentials expire in 5 minutes
178 .fi
179 .in -2
180 .sp

182 .LP
183 \fBExample 2 \fRSpecifying Renewal
184 .sp
185 .LP
186 The following \fBwarn.conf\fR entry:

188 .sp
189 .in +2
190 .nf
191 * renew:log terminal 30m
```



```
192 .fi
193 .in -2

195 .sp
196 .LP
197 \&...specifies that renew results will be sent to the user's terminal 30
198 minutes before the expiration of the TGT for all principals. The form of the
199 message (on renew success) is:

201 .sp
202 .in +2
203 .nf
204 myname@ACME.COM: your kerberos credentials have been renewed
205 .fi
206 .in -2

208 .SH FILES
210 .sp
209 .ne 2
210 .na
211 \fB\fB/usr/lib/krb5/ktkt_warnd\fR\fR
212 .ad
213 .RS 28n
214 Kerberos warning daemon
215 .RE

217 .SH ATTRIBUTES
220 .sp
218 .LP
219 See \fBattributes\fR(5) for descriptions of the following attributes:
220 .sp

222 .sp
223 .TS
224 box;
225 c | c
226 l | l .
227 ATTRIBUTE TYPE  ATTRIBUTE VALUE
228 _
229 Interface Stability     Evolving
230 .TE

232 .SH SEE ALSO
236 .sp
233 .LP
234 \fBkinit\fR(1), \fBkdestroy\fR(1), \fBktkt_warnd\fR(1M), \fBsyslog.conf\fR(4),
235 \fButmpx\fR(4), \fBattributes\fR(5), \fBkerberos\fR(5), \fBpam_krb5\fR(5)
236 .SH NOTES
241 .sp
237 .LP
238 The auto-renew of the TGT is attempted only if the user is logged-in, as
239 determined by examining \fButmpx\fR(4).
```

```
**********************************************************
    4654 Sun Sep 16 19:22:57 2018
new/usr/src/man/man4/ypfiles.4
9842 man page typos and spelling
**********************************************************
```

```
   1 '\" te
   2 .\" Copyright (c) 2003, Sun Microsystems, Inc. All Rights Reserved.
   3 .\" Copyright 1989 AT&T
   4 .\" The contents of this file are subject to the terms of the Common Development
   5 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
   6 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
   7 .TH YPFILES 4 "Feb 25, 2017"
   8 .SH NAME
   9 ypfiles \- Network Information Service Version 2, formerly known as YP
  10 .SH DESCRIPTION
  11 .LP
  12 The NIS network information service uses a distributed, replicated database of
  13 \fBdbm\fR files , in ASCII form, that are contained in the \fB/var/yp\fR
  14 directory hierarchy on each NIS server.
  15 .sp
  16 .LP
  17 A \fBdbm\fR database served by the NIS server is called a NIS \fImap\fR. A NIS
  18 \fIdomain\fR is a subdirectory of \fB/var/yp\fR that contains a set of NIS maps
  19 on each NIS server.
  20 .sp
  21 .LP
  22 Standard nicknames are defined in the file \fB/var/yp/nicknames\fR. These names
  23 can be used in place of the full map name in the \fBypmatch\fR and \fBypcat\fR
  24 commands. Use the command \fBypwhich\fR \fB-x\fR to display the current set of
  25 nicknames. Use the command \fBypwhich\fR \fB-m\fR to display all the available
  26 maps. Each line of the nickname file contains two fields separated by white
  27 space. The first field is the nickname, and the second field is the name of the
  28 map that it expands to. The nickname cannot contain a ".".
  29 .SS "NIS to LDAP (N2L)"
  30 .LP
  31 If the \fB/var/yp/NISLDAPmapping\fR configuration file is present, the NIS
  32 server operates in NIS to LDAP (N2L) mode. In this mode, NIS maps are stored in
  33 a new set of DBM files, prepended by the \fBLDAP_\fR prefix, at
  34 \fB/var/yp/\fIdomainname\fR\fR. These files are used as a cache backed by
  34 \fB/var/yp/\fIdomainename\fR\fR. These files are used as a cache backed by
  35 information from an LDAP server. Additional DBM files are created in the same
  36 directory to hold the cache's TTL values.
  37 .sp
  38 .LP
  39 N2L mode enables NIS clients to be supported in an LDAP environment.
  40 .sp
  41 .LP
  42 In N2L mode, the old style DBM files, NIS source files, and the
  43 \fBypmake\fR(1M) utility have to role. They are retained to enable easy
  44 conversion back to the traditional mode, if required.
  45 .SS "Converting from N2L to Traditional NIS"
  46 .LP
  47 When NIS is operating in N2L mode, it uses a new set of NIS maps with an
  48 \fBLDAP_\fR prefix, based on the contents of the LDAP DIT. The NIS source files
  49 are unused and become out of date. If you wish to convert back to the
  50 traditional NIS mode, the N2L configuration file should be deleted. The system
  51 will then return to using the standard map files. Optionally, the N2L mode map
  52 files, \fB/var/yp/*/LDAP_*\fR can also be deleted.
  53 .sp
  54 .LP
  55 If you want to run the system in traditional mode with information based on the
  56 DIT, then the NIS source files must be regenerated based on the N2L maps. To
  57 regenerate the NIS source files based on the N2L maps, run \fBypmap2src\fR(1M).
  58 .SH FILES
  59 .ne 2
  60 .na
```



```
  61 \fB\fB/var/yp\fR\fR
  62 .ad
  63 .sp .6
  64 .RS 4n
  65 Directory containing NIS configuration files.
  66 .RE

  68 .sp
  69 .ne 2
  70 .na
  71 \fB\fB/var/yp/binding\fR\fR
  72 .ad
  73 .sp .6
  74 .RS 4n
  75 Stores the information required to bind the NIS client to the NIS server.
  76 .RE

  78 .sp
  79 .ne 2
  80 .na
  81 \fB\fB/var/yp/binding/\fIypdomain\fR/ypservers\fR\fR
  82 .ad
  83 .sp .6
  84 .RS 4n
  85 Contains the servers to which the NIS client is allowed to bind.
  86 .RE

  88 .sp
  89 .ne 2
  90 .na
  91 \fB\fB/var/yp/Makefile\fR\fR
  92 .ad
  93 .sp .6
  94 .RS 4n
  95 Builds the NIS \fBndbm\fR databases.
  96 .RE

  98 .sp
  99 .ne 2
 100 .na
 101 \fB\fB/var/yp/nicknames\fR\fR
 102 .ad
 103 .sp .6
 104 .RS 4n
 105 Nicknames file.
 106 .RE

 108 .sp
 109 .ne 2
 110 .na
 111 \fB\fB/var/yp/securenets\fR\fR
 112 .ad
 113 .sp .6
 114 .RS 4n
 115 Defines the hosts and networks that are granted access to information in the
 116 served domain. This file is read at startup time by \fBypserv\fR and
 117 \fBypxfrd\fR.
 118 .RE

 120 .sp
 121 .ne 2
 122 .na
 123 \fB\fB/var/yp/\fIypdomain\fR\fR\fR
 124 .ad
 125 .sp .6
 126 .RS 4n
```

```
 127 Directory containing \fBndbm\fR databases.
 128 .RE

 130 .sp
 131 .ne 2
 132 .na
 133 \fB\fB/var/yp/NISLDAPmapping\fR\fR
 134 .ad
 135 .sp .6
 136 .RS 4n
 137 NIS to LDAP configuration file
 138 .RE

 140 .sp
 141 .ne 2
 142 .na
 143 \fB\fB/var/yp/*/LDAP_*\fR\fR
 144 .ad
 145 .sp .6
 146 .RS 4n
 147 NIS to LDAP mode map files
 148 .RE

 150 .SH SEE ALSO
 151 .LP
 152 \fBldap\fR(1), \fBmakedbm\fR(1M), \fBypbind\fR(1M), \fBypinit\fR(1M),
 153 \fBypmake\fR(1M), \fBypmap2src\fR(1M), \fBypserv\fR(1M), \fBypxfrd\fR(1M),
 154 \fBndbm\fR(3C), \fBypclnt\fR(3NSL)
```

```
**********************************************************
   17879 Sun Sep 16 19:22:57 2018
new/usr/src/man/man5/acl.5
9842 man page typos and spelling
**********************************************************
'\" te
.\" Copyright 2014 Nexenta Systems, Inc.  All rights reserved.
.\" Copyright (c) 2008, Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH ACL 5 "Nov 24, 2014"
.SH NAME
acl \- Access Control Lists
.SH DESCRIPTION
.LP
Access control lists (ACLs) are discretionary access control mechanisms that
grant and deny access to files and directories. Two different ACL models are
supported in the Solaris release: POSIX-draft ACLs and NFSv4 ACLs.
.sp
.LP
The older, POSIX-draft model is supported by the UFS file system. This model is
based on a withdrawn ACL POSIX specification that was never standardized. It
was subsequently withdrawn by the POSIX committee.
.sp
.LP
The other model is based on the standards of the NFSv4 working group and is an
approved standard from the Internet Engineering Task Force (IETF). The ZFS file
system uses the NFSv4 model, and provides richer semantics and finer grained
permission capabilities than the POSIX-draft model.
.SS "\fBPOSIX\fR-draft \fBACL\fRs"
.LP
POSIX-draft ACLs provide an alternative security mechanism to basic UNIX file
permissions in the Solaris release. Their purpose is to further restrict access
to files and directories or to extend permissions to a particular user. ACLs
can be used to change the permissions for the standard owner, group and other
class bits of a file's mode. ACLs can give additional users and groups access
to the file. A directory can also have a special kind of ACL called a
\fBdefault\fR ACL, which defines ACL entries to be inherited by descendents of
the directory. POSIX-draft ACLs have an ACL entry called \fBmask\fR. The mask
defines the maximum permissions that can be granted to additional user and
group entries. Whenever a file is created or its mode is changed by
\fBchmod\fR(1) or \fBchmod\fR(2), the mask is recomputed. It is recomputed to
be the group permission defined in the mode passed to \fBchmod\fR(2).
.sp
.LP
The POSIX-draft ACL model uses the standard \fBrwx\fR model of traditional UNIX
permissions.
.sp
.LP
An ACL is represented as follows:
.sp
.in +2
.nf
\fIacl_entry\fR[,\fIacl_entry\fR]...
.fi
.in -2
.sp

.sp
.LP
Each \fIacl_entry\fR contains one ACL entry. An ACL entry is represented by two
or three colon-separated(\fB:\fR) fields.
.sp
.ne 2
.na
```



```
\fB\fIuser\fR:[\fIuid\fR]:\fIperms\fR\fR
.ad
.RS 21n
If \fIuid\fR blank, it represents the file owner.
.RE

.sp
.ne 2
.na
\fB\fIgroup\fR:[\fIgid\fR]:\fIperms\fR\fR
.ad
.RS 21n
If \fIgid\fR is blank, it represents the owning group.
.RE

.sp
.ne 2
.na
\fB\fIother\fR:\fIperms\fR\fR
.ad
.RS 21n
Represents the file other class.
.RE

.sp
.ne 2
.na
\fB\fImask\fR:\fIperms\fR\fR
.ad
.RS 21n
Defines the \fBMAX\fR permission to hand out.
.RE

.sp
.LP
For example to give user \fBjoe\fR read and write permissions, the ACL entry is
specified as:
.sp
.in +2
.nf
user:joe:rw-
.fi
.in -2
.sp

.SS "\fBNFS\fRv4 \fBACL\fRs"
.LP
NFSv4 ACL model is based loosely on the Windows NT ACL model. NFSv4 ACLs
provide a much richer ACL model than POSIX-draft ACLs.
.sp
.LP
The major differences between NFSv4 and POSIX-draft ACLs are as follows:
.RS +4
.TP
.ie t \(bu
.el o
NFSv4 ACLs provide finer grained permissions than the \fBrwx\fR model.
.RE
.RS +4
.TP
.ie t \(bu
.el o
NFSv4 ACLs allow for both \fBALLOW\fR and \fBDENY\fR entries.
.RE
.RS +4
.TP
```

```
.ie t \(bu
.el o
NFSv4 ACLs provide a rich set of inheritance semantics. POSIX ACLs also have
inheritance, but with the NFSv4 model you can control the following inheritance
features:
.RS +4
.TP
.ie t \(bu
.el o
Whether inheritance cascades to both files and directories or only to files or
directories.
.RE
.RS +4
.TP
.ie t \(bu
.el o
In the case of directories, you can indicate whether inheritance is applied to
the directory itself, to just one level of subdirectories, or cascades to all
subdirectories of the directory.
.RE
.RE
.RS +4
.TP
.ie t \(bu
.el o
NFSv4 ACLs provide a mechanism for hooking into a system’s audit trail.
Currently, Solaris does not support this mechanism.
.RE
.RS +4
.TP
.ie t \(bu
.el o
NFSv4 ACLs enable administrators to specify the order in which ACL entries are
checked. With POSIX-draft ACLs the file system reorders ACL entries into a well
defined, strict access, checking order.
.RE
.sp
.LP
POSIX-draft ACL semantics can be achieved with NFSv4 ACLs. However, only some
NFSv4 ACLs can be translated to equivalent POSIX-draft ACLs.
.sp
.LP
Permissions can be specified in three different \fBchmod\fR ACL formats:
verbose, compact, or positional. The verbose format uses words to indicate that
the permissions are separated with a forward slash (\fB/\fR) character. Compact
format uses the permission letters and positional format uses the permission
letters or the hyphen (\fB-\fR) to identify no permissions.
.sp
.LP
The permissions for verbose mode and their abbreviated form in parentheses for
compact and positional mode are described as follows:
.sp
.ne 2
.na
\fBread_data (\fBr\fR)\fR
.ad
.RS 24n
Permission to read the data of the file
.RE

.sp
.ne 2
.na
\fBlist_directory (\fBr\fR)\fR
```



```
.ad
.RS 24n
Permission to list the contents of a directory.
.RE

.sp
.ne 2
.na
\fBwrite_data (\fBw\fR)\fR
.ad
.RS 24n
Permission to modify a file’s data anywhere in the file’s offset range. This
includes the ability to grow the file or write to any arbitrary offset.
.RE

.sp
.ne 2
.na
\fBadd_file (\fBw\fR)\fR
.ad
.RS 24n
Permission to add a new file to a directory.
.RE

.sp
.ne 2
.na
\fBappend_data (\fBp\fR)\fR
.ad
.RS 24n
The ability to modify the file’s data, but only starting at EOF. Currently,
this permission is not supported.
.RE

.sp
.ne 2
.na
\fBadd_subdirectory (\fBp\fR)\fR
.ad
.RS 24n
Permission to create a subdirectory to a directory.
.RE

.sp
.ne 2
.na
\fBread_xattr (\fBR\fR)\fR
.ad
.RS 24n
The ability to read the extended attributes of a file or do a lookup in the
extended attributes directory.
.RE

.sp
.ne 2
.na
\fBwrite_xattr (\fBW\fR)\fR
.ad
.RS 24n
The ability to create extended attributes or write to the extended attributes
directory.
.RE

.sp
.ne 2
.na
```

```
258 \fBexecute (\fBx\fR)\fR
259 .ad
260 .RS 24n
261 Permission to execute a file.
262 .RE

264 .sp
265 .ne 2
266 .na
267 \fBread_attributes (\fBa\fR)\fR
268 .ad
269 .RS 24n
270 The ability to read basic attributes (non-ACLs) of a file. Basic attributes are
271 considered to be the stat level attributes. Allowing this access mask bit means
272 that the entity can execute \fBls\fR(1) and \fBstat\fR(2).
273 .RE

275 .sp
276 .ne 2
277 .na
278 \fBwrite_attributes (\fBA\fR)\fR
279 .ad
280 .RS 24n
281 Permission to change the times associated with a file or directory to an
282 arbitrary value.
283 .RE

285 .sp
286 .ne 2
287 .na
288 \fBdelete (\fBd\fR)\fR
289 .ad
290 .RS 24n
291 Permission to delete the file.
292 .RE

294 .sp
295 .ne 2
296 .na
297 \fBdelete_child (\fBD\fR)\fR
298 .ad
299 .RS 24n
300 Permission to delete a file within a directory.
301 .RE

303 .sp
304 .ne 2
305 .na
306 \fBread_acl (\fBc\fR)\fR
307 .ad
308 .RS 24n
309 Permission to read the ACL.
310 .RE

312 .sp
313 .ne 2
314 .na
315 \fBwrite_acl (\fBC\fR)\fR
316 .ad
317 .RS 24n
318 Permission to write the ACL or the ability to execute \fBchmod\fR(1) or
319 \fBsetfacl\fR(1).
320 .RE

322 .sp
323 .ne 2
```



```
324 .na
325 \fBwrite_owner (\fBo\fR)\fR
326 .ad
327 .RS 24n
328 Permission to change the owner or the ability to execute \fBchown\fR(1) or
329 \fBchgrp\fR(1).
330 .RE

332 .sp
333 .ne 2
334 .na
335 \fBsynchronize (\fBs\fR)\fR
336 .ad
337 .RS 24n
338 Permission to access a file locally at the server with synchronous reads and
339 writes. Currently, this permission is not supported.
340 .RE

342 .sp
343 .LP
344 The following inheritance flags are supported by NFSv4 ACLs:
345 .sp
346 .ne 2
347 .na
348 \fBfile_inherit (\fBf\fR)\fR
349 .ad
350 .RS 26n
351 Inherit to all newly created files in a directory.
352 .RE

354 .sp
355 .ne 2
356 .na
357 \fBdir_inherit (\fBd\fR)\fR
358 .ad
359 .RS 26n
360 Inherit to all newly created directories in a directory.
361 .RE

363 .sp
364 .ne 2
365 .na
366 \fBinherit_only (\fBi\fR)\fR
367 .ad
368 .RS 26n
369 Placed on a directory, but does not apply to the directory itself, only to
370 newly created files and directories. This flag requires file_inherit
371 and/or dir_inherit to indicate what to inherit.
372 .RE

374 .sp
375 .ne 2
376 .na
377 \fBno_propagate (\fBn\fR)\fR
378 .ad
379 .RS 26n
380 Placed on directories and indicates that ACL entries should only be inherited
381 one level of the tree. This flag requires file_inherit and/or dir_inherit to
382 indicate what to inherit.
383 .RE

385 .sp
386 .ne 2
387 .na
388 \fBsuccessful_access (\fBS\fR)\fR
389 .ad
```

```
.RS 26n
Indicates whether an alarm or audit record should be initiated upon successful
accesses. Used with audit/alarm ACE types.
.RE

.sp
.ne 2
.na
\fBfailed_access (\fBF\fR)\fR
.ad
.RS 26n
Indicates whether an alarm or audit record should be initiated when access
fails. Used with audit/alarm ACE types.
.RE

.sp
.ne 2
.na
\fBinherited (\fBI\fR)\fR
.ad
.RS 26n
ACE was inherited.
.RE

.sp
.ne 2
.na
\fB\fB-\fR\fR
.ad
.RS 26n
No permission granted.
.RE

.sp
.LP
An NFSv4 ACL is expressed using the following syntax:
.sp
.in +2
.nf
\fIacl_entry\fR[,\fIacl_entry\fR]...

    owner@:<perms>[:inheritance flags]:<allow|deny>
    group@:<perms>[:inheritance flags]:<allow|deny>
    everyone@:<perms>[:inheritance flags]:<allow|deny>
    user:<username>:<perms>[:inheritance flags]:<allow|deny>
    usersid:<sid string>:<perms>[:inheritance flags]:<allow|deny>
    group:<groupname>:<perms>[:inheritance flags]:<allow|deny>
    groupsid:<sid string>:<perms>[:inheritance flags]:<allow|deny>
    sid:<sid string>:<perms>[:inheritance flags]:<allow|deny>
.fi
.in -2

.sp
.ne 2
.na
\fBowner@\fR
.ad
.RS 10n
File owner
.RE

.sp
.ne 2
.na
\fBgroup@\fR
.ad
```



```
.RS 10n
Group owner
.RE

.sp
.ne 2
.na
\fBuser\fR
.ad
.RS 10n
Permissions for a specific user
.RE

.sp
.ne 2
.na
\fBgroup\fR
.ad
.RS 10n
Permissions for a specific group
.RE

.sp
.LP
Permission and inheritance flags are separated by a \fB/\fR character.
.sp
.LP
ACL specification examples:
.sp
.in +2
.nf
user:fred:read_data/write_data/read_attributes:file_inherit:allow
owner@:read_data:allow,group@:read_data:allow,user:tom:read_data:deny
.fi
.in -2
.sp

.sp
.LP
Using the compact ACL format, permissions are specified by using 14 unique
letters to indicate permissions.
.sp
.LP
Using the positional ACL format, permissions are specified as positional
arguments similar to the \fBls -V\fR format. The hyphen (\fB-\fR), which
indicates that no permission is granted at that position, can be omitted and
only the required letters have to be specified.
.sp
.LP
The letters above are listed in the order they would be specified in positional
notation.
.sp
.LP
With these letters you can specify permissions in the following equivalent
ways.
.sp
.in +2
.nf
user:fred:rw------R------:file_inherit:allow
.fi
.in -2
.sp

.sp
.LP
Or you can remove the \fB-\fR and scrunch it together.
```

```
.sp
.in +2
.nf
user:fred:rwR:file_inherit:allow
.fi
.in -2
.sp

.sp
.LP
The inheritance flags can also be specified in a more compact manner, as
follows:
.sp
.in +2
.nf
user:fred:rwR:f:allow
user:fred:rwR:f------:allow
.fi
.in -2
.sp

.SS "Shell-level Solaris \fBAPI\fR"
.LP
The Solaris command interface supports the manipulation of ACLs. The following
Solaris utilities accommodate both ACL models:
.sp
.ne 2
.na
\fB\fBchmod\fR\fR
.ad
.RS 12n
The \fBchmod\fR utility has been enhanced to allow for the setting and deleting
of ACLs. This is achieved by extending the symbolic-mode argument to support
ACL manipulation. See \fBchmod\fR(1) for details.
.RE

.sp
.ne 2
.na
\fB\fBcompress\fR\fR
.ad
.RS 12n
When a file is compressed any ACL associated with the original file is
preserved with the compressed file.
.RE

.sp
.ne 2
.na
\fB\fBcp\fR\fR
.ad
.RS 12n
By default, \fBcp\fR ignores ACLs, unless the \fB-p\fR option is specified.
When \fB-p\fR is specified the owner and group id, permission modes,
modification and access times, ACLs, and extended attributes if applicable are
preserved.
.RE

.sp
.ne 2
.na
\fB\fBcpio\fR\fR
.ad
.RS 12n
ACLs are preserved when the \fB-P\fR option is specified.
.RE
```



```
.sp
.ne 2
.na
\fB\fBfind\fR\fR
.ad
.RS 12n
Find locates files with ACLs when the \fB-acl\fR flag is specified.
.RE

.sp
.ne 2
.na
\fB\fBls\fR\fR
.ad
.RS 12n
By default \fBls\fR does not display ACL information. When the \fB-v\fR option
is specified, a file's ACL is displayed.
.RE

.sp
.ne 2
.na
\fB\fBmv\fR\fR
.ad
.RS 12n
When a file is moved, all attributes are carried along with the renamed file.
When a file is moved across a file system boundary, the ACLs are replicated. If
the ACL information cannot be replicated, the move fails and the source file is
not removed.
.RE

.sp
.ne 2
.na
\fB\fBpack\fR\fR
.ad
.RS 12n
When a file is packed, any ACL associated with the original file is preserved
with the packed file.
.RE

.sp
.ne 2
.na
\fB\fBrcp\fR\fR
.ad
.RS 12n
\fBrcp\fR has been enhanced to support copying. A file's ACL is only preserved
when the remote host supports ACLs.
.RE

.sp
.ne 2
.na
\fB\fBtar\fR\fR
.ad
.RS 12n
ACLs are preserved when the \fB-p\fR option is specified.
.RE

.sp
.ne 2
.na
\fB\fBunpack\fR\fR
.ad
```

```
654 .RS 12n
655 When a file with an ACL is unpacked, the unpacked file retains the ACL
656 information.
657 .RE

659 .SS "Application-level \fBAPI\fR"
660 .LP
661 The primary interfaces required to access file system ACLs at the programmatic
662 level are the \fBacl_get()\fR and \fBacl_set()\fR functions. These functions
663 support both POSIX draft ACLs and NFSv4 ACLs.
664 .SS "Retrieving a file's \fBACL\fR"
665 .in +2
666 .nf
667 int acl_get(const char *path, int flag, acl_t **aclp);
668 int facl_get(int fd, int flag, acl_t **aclp);
669 .fi
670 .in -2

672 .sp
673 .LP
674 The \fBacl_get\fR(3SEC) and \fBfacl_get\fR(3SEC) functions retrieves an ACL on
675 a file whose name is given by path or referenced by the open file descriptor
676 fd. The flag argument specifies whether a trivial ACL should be retrieved. When
677 the flag argument equals \fBACL_NO_TRIVIAL\fR then only ACLs that are not
678 trivial are retrieved. The ACL is returned in the \fBaclp\fR argument.
679 .SS "Freeing \fBACL\fR structure"
680 .in +2
681 .nf
682 void acl_free(acl_t *aclp)s;
683 .fi
684 .in -2

686 .sp
687 .LP
688 The \fBacl_free()\fR function frees up memory allocated for the argument
689 \fBaclp;\fR.
690 .SS "Setting an \fBACL\fR on a file"
691 .in +2
692 .nf
693 int acl_set(const char *path, acl_t *aclp);
694 int facl_set(int fd, acl_t *aclp);
695 .fi
696 .in -2

698 .sp
699 .LP
700 The \fBacl_set\fR(3SEC) and \fBfacl_get\fR(3SEC) functions are used for setting
701 an ACL on a file whose name is given by path or referenced by the open file
702 descriptor \fBfd\fR. The \fBaclp\fR argument specifies the ACL to set. The
703 \fBacl_set\fR(3SEC) translates an POSIX-draft ACL into a NFSv4 ACL when the
704 target file systems supports NFSv4 ACLs. No translation is performed when
705 trying to set an NFSv4 ACL on a POSIX-draft ACL supported file system.
706 .SS "Determining an \fBACL\fR's trivialness"
707 .in +2
708 .nf
709 int acl_trivial(const char *path);
710 .fi
711 .in -2

713 .sp
714 .LP
715 The \fBacl_trivial()\fR function is used to determine whether a file has a
716 trivial ACL.
717 .SS "Removing all \fBACL\fRs from a file"
718 .in +2
719 .nf
```



```
720 int acl_strip(const char *path, uid_t uid, gid_t gid, mode_t mode);
721 .fi
722 .in -2

724 .sp
725 .LP
726 The \fBacl_strip()\fR function removes all ACLs from a file and replaces them
727 with a trivial ACL based off of the passed in argument mode. After replacing
728 the ACL the owner and group of the file are set to the values specified in the
729 uid and gid parameters.
730 .SS "Converting \fBACL\fRs to/from external representation"
731 .in +2
732 .nf
733 int acl_fromtext(const char *path, acl_t **aclp);
734 char *acl_totext(acl_t *aclp, int flags);
735 .fi
736 .in -2

738 .sp
739 .LP
740 The \fBacl_totext()\fR function converts an internal ACL representation pointed
741 to by aclp into an external representation. See \fBDESCRIPTION\fR for details
742 about external representation.
743 .sp
744 .LP
745 The \fBacl_fromtext()\fR functions converts and external representation into an
746 internal representation. See \fBDESCRIPTION\fR for details about external
747 representation.
748 .SH EXAMPLES
749 .LP
750 The following examples demonstrate how the API can be used to perform basic
751 operations on ACLs.
752 .LP
753 \fBExample 1 \fRRetrieving and Setting an ACL
754 .sp
755 .LP
756 Use the following to retrieve an ACL and set it on another file:

758 .sp
759 .in +2
760 .nf
761 error = acl_get("file", ACL_NO_TRIVIAL, &aclp);

763 if (error == 0 && aclp != NULL) {
764 .in +8
765 error = acl_set("file2", aclp);
766 acl_free(aclp);
767 .in -8
768 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    9374 Sun Sep 16 19:22:57 2018
new/usr/src/man/man5/byteorder.5
9842 man page typos and spelling
**********************************************************
```

```
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\" Copyright 2016 Joyent, Inc.
.\"
.Dd August 2, 2018
.Dt BYTEORDER 5
.Os
.Sh NAME
.Nm byteorder ,
.Nm endian
.Nd byte order and endianness
.Sh DESCRIPTION
Integer values which occupy more than 1 byte in memory can be laid out
in different ways on different platforms.
In particular, there is a major split between those which place the least
significant byte of an integer at the lowest address, and those which place the
most significant byte there instead.
As this difference relates to which end of the integer is found in memory first,
the term
.Em endian
is used to refer to a particular byte order.
.Pp
A platform is referred to as using a
.Em big-endian
byte order when it places the most significant byte at the lowest
address, and
.Em little-endian
when it places the least significant byte first.
Some platforms may also switch between big- and little-endian mode and run code
compiled for either.
.Pp
Historically, there have also been some systems that utilized
.Em middle-endian
byte orders for integers larger than 2 bytes.
Such orderings are not in common use today.
.Pp
Endianness is also of particular importance when dealing with values
that are being read into memory from an external source.
For example, network protocols such as IP conventionally define the fields in a
packet as being always stored in big-endian byte order.
This means that a little-endian machine will have to perform transformations on
these fields in order to process them.
.Ss Examples
To illustrate endianness in memory, let us consider the decimal integer
2864434397.
This number fits in 32 bits of storage (4 bytes).
.Pp
On a big-endian system, this integer would be written into memory as
the bytes 0xAA, 0xBB, 0xCC, 0xDD, in order from lowest memory address to
highest.
.Pp
On a little-endian system, it would be written instead as the bytes
```



```
0xDD, 0xCC, 0xBB, 0xAA, in that order.
.Pp
If both the big- and little-endian systems were asked to store this
integer at address 0x100, we would see the following in each of their
memory:
.Bd -literal

                    Big-Endian

        ++------++------++------++------++
        || 0xAA || 0xBB || 0xCC || 0xDD ||
        ++------++------++------++------++
            ^^      ^^      ^^      ^^
          0x100   0x101   0x102   0x103
            vv      vv      vv      vv
        ++------++------++------++------++
        || 0xDD || 0xCC || 0xBB || 0xAA ||
        ++------++------++------++------++

                  Little-Endian
.Ed
.Pp
It is particularly important to note that even though the byte order is
different between these two machines, the bit ordering within each byte,
by convention, is still the same.
.Pp
For example, take the decimal integer 4660, which occupies in 16 bits (2
bytes).
.Pp
On a big-endian system, this would be written into memory as 0x12, then
0x34.
.Pp
On a little-endian system, it would be written as 0x34, then 0x12.
Note that this is not at all the same as seeing 0x43 then 0x21 in memory --
only the bytes are re-ordered, not any bits (or nybbles) within them.
.Pp
As before, storing this at address 0x100:
.Bd -literal
                    Big-Endian

                ++------++------++
                || 0x12 || 0x34 ||
                ++------++------++
                    ^^      ^^
                  0x100   0x101
                    vv      vv
                ++------++------++
                || 0x34 || 0x12 ||
                ++------++------++

                   Little-Endian
.Ed
.Pp
This example shows how an eight byte number, 0xBADCAFEDEADBEEF is stored
in both big and little-endian:
.Bd -literal
                        Big-Endian

    +------+------+------+------+------+------+------+------+
    | 0xBA | 0xDC | 0xAF | 0xFE | 0xDE | 0xAD | 0xBE | 0xEF |
    +------+------+------+------+------+------+------+------+
       ^^     ^^     ^^     ^^     ^^     ^^     ^^     ^^
     0x100  0x101  0x102  0x103  0x104  0x105  0x106  0x107
       vv     vv     vv     vv     vv     vv     vv     vv
    +------+------+------+------+------+------+------+------+
    | 0xEF | 0xBE | 0xAD | 0xDE | 0xFE | 0xAF | 0xDC | 0xBA |
```

```
128     +------+------+------+------+------+------+------+------+

130                         Little-Endian

132 .Ed
133 .Pp
134 The treatment of different endian values would not be complete without
135 discussing
136 .Em PDP-endian ,
137 which is also known as
138 .Em middle-endian .
139 While the PDP-11 was a 16-bit little-endian system, it laid out 32-bit
140 values in a different way from current little-endian systems.
141 First, it would divide a 32-bit number into two 16-bit numbers.
142 Each 16-bit number would be stored in little-endian; however, the two 16-bit
143 words would be stored with the larger 16-bit word appearing first in memory,
144 followed by the latter.
145 .Pp
146 The following image illustrates PDP-endian and compares it against
147 little-endian values.
148 Here, we'll start with the value 0xAABBCCDD and show how the four bytes for it
149 will be laid out, starting at 0x100.
150 .Bd -literal
151                       PDP-Endian

153         ++------++------++------++------++
154         || 0xBB || 0xAA || 0xDD || 0xCC ||
155         ++------++------++------++------++
156             ^^      ^^      ^^      ^^
157           0x100   0x101   0x102   0x103
158             vv      vv      vv      vv
159         ++------++------++------++------++
160         || 0xDD || 0xCC || 0xBB || 0xAA ||
161         ++------++------++------++------++

163                   Little-Endian

165 .Ed
166 .Ss Network Byte Order
167 The term 'network byte order' refers to big-endian ordering, and
168 originates from the IEEE.
169 Early disagreements over which byte ordering to use for network traffic prompted
170 RFC1700 to define that all IETF-specified network protocols use big-endian
171 ordering unless noted explicitly otherwise.
172 The Internet protocol family (IP, and thus TCP and UDP etc) particularly adhere
173 to this convention.
174 .Ss Determining the System's Byte Order
175 The operating system supports both big-endian and little-endian CPUs.
176 To make it easier for programs to determine the endianness of the platform they
177 are being compiled for, functions and macro constants are provided in the system
178 header files.
179 .Pp
180 The endianness of the system can be obtained by including the header
181 .In sys/types.h
182 and using the pre-processor macros
183 .Sy _LITTLE_ENDIAN
184 and
185 .Sy _BIG_ENDIAN .
186 See
187 .Xr types.h 3HEAD
188 for more information.
189 .Pp
190 Additionally, the header
191 .In endian.h
192 defines an alternative means for determining the endianness of the
193 current system.
```



```
194 See
195 .Xr endian.h 3HEAD
196 for more information.
197 .Pp
198 illumos runs on both big- and little-endian systems.
199 When writing software for which the endianness is important, one must always
200 check the byte order and convert it appropriately.
201 .Ss Converting Between Byte Orders
202 The system provides two different sets of functions to convert values
203 between big-endian and little-endian.
204 They are defined in
205 .Xr byteorder 3C
206 and
207 .Xr endian 3C .
208 .Pp
209 The
210 .Xr byteorder 3C
211 family of functions convert data between the host's native byte order
212 and big- or little-endian.
213 The functions operate on either 16-bit, 32-bit, or 64-bit values.
214 Functions that convert from network byte order to the host's byte order
215 start with the string
216 .Sy ntoh ,
217 while functions which convert from the host's byte order to network byte
218 order, begin with
219 .Sy hton .
220 For example, to convert a 32-bit value, a long, from network byte order
221 to the host's, one would use the function
222 .Xr ntohl 3C .
223 .Pp
224 These functions have been standardized by POSIX.
225 However, the 64-bit variants,
226 .Xr ntohll 3C
227 and
228 .Xr htonll 3C
229 are not standardized and may not be found on other systems.
230 For more information on these functions, see
231 .Xr byteorder 3C .
232 .Pp
233 The second family of functions,
234 .Xr endian 3C ,
235 provide a means to convert between the host's byte order
236 and big-endian and little-endian specifically.
237 While these functions are similar to those in
238 .Xr byteorder 3C ,
239 they more explicitly cover different data conversions.
240 Like them, these functions operate on either 16-bit, 32-bit, or 64-bit values.
241 When converting from big-endian, to the host's endianness, the functions
242 begin with
243 .Sy betoh .
244 If instead, one is converting data from the host's native endianness to
245 another, then it starts with
246 .Sy htobe .
247 When working with little-endian data, the prefixes
248 .Sy letoh
249 and
250 .Sy htole
251 convert little-endian data to the host's endianness and from the host's
252 to little-endian respectively.
253 .Pp
254 These functions are not standardized and the header they appear in varies
255 between the BSDs and GNU/Linux.
256 .Sy Applications that wish to be portable, should instead use the
256 Applications that wish to be portable, shoulda instead use the
257 .Xr byteorder 3C
258 functions.
```

```
259 .Pp
260 All of these functions in both families simply return their input when
261 the host’s native byte order is the same as the desired order.
262 For example, when calling
263 .Xr htonl 3C
264 on a big-endian system the original data is returned with no conversion
265 or modification.
266 .Sh SEE ALSO
267 .Xr byteorder 3C ,
268 .Xr endian 3C ,
269 .Xr endian.h 3HEAD ,
270 .Xr inet 3HEAD
```

```
**********************************************************
   16790 Sun Sep 16 19:22:57 2018
new/usr/src/man/man5/cancellation.5
9842 man page typos and spelling
**********************************************************
.\"
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for
.\" permission to reproduce portions of its copyrighted documentation.
.\" Original documentation from The Open Group can be obtained online at
.\" http://www.opengroup.org/bookstore/.
.\"
.\" The Institute of Electrical and Electronics Engineers and The Open
.\" Group, have given us permission to reprint portions of their
.\" documentation.
.\"
.\" In the following statement, the phrase ‘‘this text’’ refers to portions
.\" of the system documentation.
.\"
.\" Portions of this text are reprinted and reproduced in electronic form
.\" in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition,
.\" Standard for Information Technology -- Portable Operating System
.\" Interface (POSIX), The Open Group Base Specifications Issue 6,
.\" Copyright (C) 2001-2004 by the Institute of Electrical and Electronics
.\" Engineers, Inc and The Open Group.  In the event of any discrepancy
.\" between these versions and the original IEEE and The Open Group
.\" Standard, the original IEEE and The Open Group Standard is the referee
.\" document.  The original Standard can be obtained online at
.\" http://www.opengroup.org/unix/online.html.
.\"
.\" This notice shall appear on any product containing this material.
.\"
.\" The contents of this file are subject to the terms of the
.\" Common Development and Distribution License (the "License").
.\" You may not use this file except in compliance with the License.
.\"
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
.\" or http://www.opensolaris.org/os/licensing.
.\" See the License for the specific language governing permissions
.\" and limitations under the License.
.\"
.\" When distributing Covered Code, include this CDDL HEADER in each
.\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
.\" If applicable, add the following below this CDDL HEADER, with the
.\" fields enclosed by brackets "[]" replaced with your own identifying
.\" information: Portions Copyright [yyyy] [name of copyright owner]
.\"
.\"
.\" Portions Copyright (c) 1995 IEEE.  All Rights Reserved.
.\" Copyright (c) 2001, The IEEE and The Open Group.  All Rights Reserved.
.\" Copyright (c) 2008, Sun Microsystems, Inc.  All Rights Reserved.
.\"
.TH CANCELLATION 5 "Oct 4, 2005"
.SH NAME
cancellation \- overview of concepts related to POSIX thread cancellation
.SH DESCRIPTION
.sp


.sp
.TS
box;
c | c
l | l .
FUNCTION        ACTION
_
\fBpthread_cancel()\fR  Cancels thread execution.
\fBpthread_setcancelstate()\fR  Sets the cancellation \fIstate\fR of a thread.
```



```
\fBpthread_setcanceltype()\fR   Sets the cancellation \fItype\fR of a thread.
\fBpthread_testcancel()\fR      T{
Creates a cancellation point in the calling thread.
T}
\fBpthread_cleanup_push()\fR    Pushes a cleanup handler routine.
\fBpthread_cleanup_pop()\fR     Pops a cleanup handler routine.
.TE

.SS "Cancellation"
.sp
.LP
Thread cancellation allows a thread to terminate the execution of  any
application thread in the process. Cancellation is useful when further
operations of one or more threads are undesirable or unnecessary.
.sp
.LP
An example of a situation that could benefit from using cancellation is an
asynchronously-generated cancel condition such as a user requesting to close or
exit some running operation. Another example is the completion of a task
undertaken by a number of threads, such as solving a maze. While many threads
search for the solution, one of the threads might solve the puzzle while the
others continue to operate. Since they are serving no purpose at that point,
they should all be canceled.
.SS "Planning Steps"
.sp
.LP
Planning and programming for most cancellations follow this pattern:
.RS +4
.TP
1.
Identify which threads you want to cancel, and insert
\fBpthread_cancel\fR(3C) statements.
.RE
.RS +4
.TP
2.
Identify system-defined cancellation points where a thread that might be
canceled could have changed system or program state that should be restored.
See the \fBCancellation Points\fR for a list.
.RE
.RS +4
.TP
3.
When a thread changes the system or program state just before a cancellation
point, and should restore that state before the thread is canceled, place a
cleanup handler before the cancellation point with
\fBpthread_cleanup_push\fR(3C). Wherever a thread restores the changed state,
pop the cleanup handler from the cleanup stack with
\fBpthread_cleanup_pop\fR(3C).
.RE
.RS +4
.TP
4.
Know whether the threads you are canceling call into cancel-unsafe
libraries, and disable cancellation with \fBpthread_setcancelstate\fR(3C)
before the call into the library.  See \fBCancellation State\fR and
\fBCancel-Safe\fR.
.RE
.RS +4
.TP
5.
To cancel a thread in a procedure that contains no cancellation points,
insert your own cancellation points with \fBpthread_testcancel\fR(3C). This
function creates cancellation points by testing for pending cancellations and
performing those cancellations if they are found. Push and pop cleanup handlers
around the cancellation point, if necessary (see Step 3, above).
```

.RE
.SS "Cancellation Points"
.sp
.LP
The system defines certain points at which cancellation can occur (cancellation points), and you can create additional cancellation points in your application with \fBpthread_testcancel()\fR.
.sp
.LP
The following cancellation points are defined by the system (system-defined cancellation points): \fBcreat\fR(2), \fBaio_suspend\fR(3C), \fBclose\fR(2), \fBcreat\fR(2), \fBgetmsg\fR(2), \fBgetpmsg\fR(2), \fBlockf\fR(3C), \fBmq_receive\fR(3C), \fBmq_send\fR(3C), \fBmsgrcv\fR(2), \fBmsgsnd\fR(2), \fBmsync\fR(3C), \fBnanosleep\fR(3C), \fBopen\fR(2), \fBpause\fR(2), \fBpoll\fR(2), \fBpread\fR(2), \fBpthread_cond_timedwait\fR(3C), \fBpthread_cond_wait\fR(3C), \fBpthread_join\fR(3C), \fBpthread_testcancel\fR(3C), \fBputmsg\fR(2), \fBputpmsg\fR(2), \fBpwrite\fR(2), \fBread\fR(2), \fBreadv\fR(2), \fBselect\fR(3C), \fBsem_wait\fR(3C), \fBsigpause\fR(3C), \fBsigwaitinfo\fR(3C), \fBsigsuspend\fR(2), \fBsigtimedwait\fR(3C), \fBsigwait\fR(2), \fBsleep\fR(3C), \fBsync\fR(2), \fBsystem\fR(3C), \fBtcdrain\fR(3C), \fBusleep\fR(3C), \fBwait\fR(3C), \fBwaitid\fR(2), \fBwait3\fR(3C), \fBwaitpid\fR(3C), \fBwrite\fR(2), \fBwritev\fR(2), and \fBfcntl\fR(2), when specifying \fBF_SETLKW\fR as the command.
.sp
.LP
When cancellation is asynchronous, cancellation can occur at any time (before, during, or after the execution of the function defined as the cancellation point). When cancellation is deferred (the default case), cancellation occurs only within the scope of a function defined as a cancellation point (after the function is called and before the function returns). See \fBCancellation Type\fR for more information about deferred and asynchronous cancellation.
.sp
.LP
Choosing where to place cancellation points and understanding how cancellation affects your program depend upon your understanding of both your application and of cancellation mechanics.
.sp
.LP
Typically, any call that might require a long wait should be a cancellation point. Operations need to check for pending cancellation requests when the operation is about to block indefinitely. This includes threads waiting in \fBpthread_cond_wait()\fR and \fBpthread_cond_timedwait()\fR, threads waiting for the termination of another thread in \fBpthread_join()\fR, and threads blocked on \fBsigwait()\fR.
.sp
.LP
A mutex is explicitly not a cancellation point and should be held for only the minimal essential time.
.sp
.LP
Most of the dangers in performing cancellations deal with properly restoring invariants and freeing shared resources. For example, a carelessly canceled thread might leave a mutex in a locked state, leading to a deadlock. Or it might leave a region of memory allocated with no way to identify it and therefore no way to free it.
.SS "Cleanup Handlers"
.sp
.LP
When a thread is canceled, it should release resources and clean up the state that is shared with other threads. So, whenever a thread that might be canceled changes the state of the system or of the program, be sure to push a cleanup handler with \fBpthread_cleanup_push\fR(3C) before the cancellation point.
.sp
.LP
When a thread is canceled, all the currently-stacked cleanup handlers are

executed in last-in-first-out (LIFO) order. Each handler is run in the scope in which it was pushed. When the last cleanup handler returns, the thread-specific data destructor functions are called. Thread execution terminates when the last destructor function returns.
.sp
.LP
When, in the normal course of the program, an uncanceled thread restores state that it had previously changed, be sure to pop the cleanup handler (that you had set up where the change took place) using \fBpthread_cleanup_pop\fR(3C). That way, if the thread is canceled later, only currently-changed state will be restored by the handlers that are left in the stack.
.sp
.LP
The \fBpthread_cleanup_push()\fR and \fBpthread_cleanup_pop()\fR functions can be implemented as macros. The application must ensure that they appear as statements, and in pairs within the same lexical scope (that is, the \fBpthread_cleanup_push()\fR macro can be thought to expand to a token list whose first token is '{' with \fBpthread_cleanup_pop()\fR expanding to a token list whose last token is the corresponding '}').
.sp
.LP
The effect of the use of \fBreturn\fR, \fBbreak\fR, \fBcontinue\fR, and \fBgoto\fR to prematurely leave a code block described by a pair of \fBpthread_cleanup_push()\fR and \fBpthread_cleanup_pop()\fR function calls is undefined.
.SS "Cancellation State"
.sp
.LP
Most programmers will use only the default cancellation state of \fBPTHREAD_CANCEL_ENABLE\fR, but can choose to change the state by using \fBpthread_setcancelstate\fR(3C), which determines whether a thread is cancelable at all. With the default \fIstate\fR of \fBPTHREAD_CANCEL_ENABLE\fR, cancellation is enabled and the thread is cancelable at points determined by its cancellation \fItype\fR. See \fBCancellation Type\fR.
.sp
.LP
If the \fIstate\fR is \fBPTHREAD_CANCEL_DISABLE\fR, cancellation is disabled, the thread is not cancelable at any point, and all cancellation requests to it are held pending.
.sp
.LP
You might want to disable cancellation before a call to a cancel-unsafe library, restoring the old cancel state when the call returns from the library. See \fBCancel-Safe\fR for explanations of cancel safety.
.SS "Cancellation Type"
.sp
.LP
A thread's cancellation \fBtype\fR is set with \fBpthread_setcanceltype\fR(3C), and determines whether the thread can be canceled anywhere in its execution or only at cancellation points.
.sp
.LP
With the default \fItype\fR of \fBPTHREAD_CANCEL_DEFERRED\fR, the thread is cancelable only at cancellation points, and then only when cancellation is enabled.
.sp
.LP
If the \fItype\fR is \fBPTHREAD_CANCEL_ASYNCHRONOUS\fR, the thread is cancelable at any point in its execution (assuming, of course, that cancellation is enabled). Try to limit regions of asynchronous cancellation to sequences with no external dependencies that could result in dangling resources or unresolved state conditions. Using asynchronous cancellation is discouraged because of the danger involved in trying to guarantee correct cleanup handling at absolutely every point in the program.
.sp

```
.sp
.TS
box;
c | c | c
l | l | l .
Cancellation Type/State Table
Type    State
        Enabled (Default)       Disabled
_
Deferred (Default)      T{
Cancellation occurs when the target thread reaches a cancellation point and a ca
T}      T{
All cancellation requests to the target thread are held pending.
T}
Asynchronous    T{
Receipt of a \fBpthread_cancel()\fR call causes immediate cancellation.
T}      T{
All cancellation requests to the target thread are held pending; as
soon as cancellation is re-enabled, pending cancellations are executed
immediately.
All cancellation requests to the target thread are held pending; as soon as canc
T}
.TE

.SS "Cancel-Safe"
.sp
.LP
With the arrival of POSIX cancellation, the Cancel-Safe level has been added to
the list of MT-Safety levels. See \fBattributes\fR(5). An application or
library is Cancel-Safe whenever it has arranged for cleanup handlers to restore
system or program state wherever cancellation can occur. The application or
library is specifically Deferred-Cancel-Safe when it is Cancel-Safe for threads
whose cancellation type is \fBPTHREAD_CANCEL_DEFERRED\fR. See \fBCancellation
State\fR. It is specifically Asynchronous-Cancel-Safe when it is Cancel-Safe
for threads whose cancellation type is \fBPTHREAD_CANCEL_ASYNCHRONOUS\fR.
.sp
.LP
It is easier to arrange for deferred cancel safety, as this requires system and
program state protection only around cancellation points. In general, expect
that most applications and libraries are not Asynchronous-Cancel-Safe.
.SS "POSIX Threads Only"
.sp
.LP
The cancellation functions described in this manual page are available for
POSIX threads, only (the Solaris threads interfaces do not provide cancellation
functions).
.SH EXAMPLES
.LP
\fBExample 1 \fRCancellation example
.sp
.LP
The following short C++ example shows the pushing/popping of cancellation
handlers, the disabling/enabling of cancellation, the use of
\fBpthread_testcancel()\fR, and so on. The \fBfree_res()\fR cancellation
handler in this example is a dummy function that simply prints a message, but
that would free resources in a real application. The function \fBf2()\fR is
called from the main thread, and goes deep into its call stack by calling
itself recursively.

.sp
.LP
Before \fBf2()\fR starts running, the newly created thread has probably posted
a cancellation on the main thread since the main thread calls \fBthr_yield()\fR
right after creating thread2.  Because cancellation was initially disabled in
the main thread, through a call to \fBpthread_setcancelstate()\fR, the call to
```



```
\fBf2()\fR from  \fBmain()\fR continues and constructs X at each recursive
call,  even though the main thread has a pending cancellation.

.sp
.LP
When \fBf2()\fR is called for the fifty-first time (when \fB"i == 50"\fR),
\fBf2()\fR enables cancellation by calling \fBpthread_setcancelstate()\fR. It
then establishes a cancellation point for itself by calling
\fBpthread_testcancel()\fR. (Because a cancellation is pending, a call to a
cancellation point such as  \fBread\fR(2) or  \fBwrite\fR(2) would also cancel
the caller here.)

.sp
.LP
After the \fBmain()\fR thread is canceled at the fifty-first iteration, all the
cleanup handlers that were pushed are called in sequence; this is indicated by
the calls to  \fBfree_res()\fR and the calls to the destructor for \fIX\fR. At
each level, the C++ runtime calls the destructor for \fIX\fR and then the
cancellation handler,  \fBfree_res()\fR. The print messages from
\fBfree_res()\fR and \fIX\fR's destructor show the sequence of calls.

.sp
.LP
At the end, the main thread is joined by thread2. Because the main thread was
canceled, its return status from  \fBpthread_join()\fR is
\fBPTHREAD_CANCELED\fR. After the status is printed, thread2 returns, killing
the  process (since it is the last thread in the process).

.sp
.in +2
.nf
#include <pthread.h>
#include <sched.h>
extern "C" void thr_yield(void);

extern "C" void printf(...);

struct X {
        int x;
        X(int i){x = i; printf("X(%d) constructed.\en", i);}
        ~X(){ printf("X(%d) destroyed.\en", x);}
};
_____unchanged_portion_omitted_
.fi
.in -2

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
MT-Level        MT-Safe
.TE

.SH SEE ALSO
.sp
.LP
\fBread\fR(2), \fBsigwait\fR(2), \fBwrite\fR(2), \fBIntro\fR(3),
```

```
425 \fBcondition\fR(5), \fBpthread_cleanup_pop\fR(3C),
426 \fBpthread_cleanup_push\fR(3C), \fBpthread_exit\fR(3C), \fBpthread_join\fR(3C),
427 \fBpthread_setcancelstate\fR(3C), \fBpthread_setcanceltype\fR(3C),
428 \fBpthread_testcancel\fR(3C), \fBsetjmp\fR(3C), \fBattributes\fR(5),
429 \fBstandards\fR(5)
```

```
***********************************************************
    5396 Sun Sep 16 19:22:57 2018
new/usr/src/man/man5/loader.4th.5
9842 man page typos and spelling
***********************************************************
```

```
.\" Copyright (c) 1999 Daniel C. Sobral
.\" All rights reserved.
.\"
.\" Redistribution and use in source and binary forms, with or without
.\" modification, are permitted provided that the following conditions
.\" are met:
.\" 1. Redistributions of source code must retain the above copyright
.\"    notice, this list of conditions and the following disclaimer.
.\" 2. Redistributions in binary form must reproduce the above copyright
.\"    notice, this list of conditions and the following disclaimer in the
.\"    documentation and/or other materials provided with the distribution.
.\"
.\" THIS SOFTWARE IS PROVIDED BY THE AUTHOR AND CONTRIBUTORS ''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 THE AUTHOR OR CONTRIBUTORS 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 DAMAGE.
.\"
.Dd Apr 22, 2017
.Dt LOADER.4TH 5
.Os
.Sh NAME
.Nm loader.4th
.Nd loader.conf processing tools
.Sh DESCRIPTION
The file that goes by the name of
.Nm
is a set of commands designed to manipulate
.Xr loader.conf 4
files.
The default
.Pa /boot/loader.rc
includes
.Nm
and uses one of its commands to automatically read and process
the standard
.Xr loader.conf 4
files.
Other commands exists to help the user specify alternate
configurations.
.Pp
The commands of
.Nm
by themselves are not enough for most uses.
Please refer to the
examples below for the most common situations, and to
.Xr loader 5
for additional commands.
.Pp
Before using any of the commands provided in
.Nm ,
it must be included
through the command:
.Pp
.Dl include loader.4th
```



```
.Pp
This line is present in the default
.Pa /boot/loader.rc
file, so it is not needed (and should not be re-issued) in a normal setup.
.Pp
The commands provided by it are:
.Bl -tag -width disable-module_module -compact -offset indent
.It Ic boot
.It Ic boot Ar kernelname Op Cm ...
.It Ic boot Ar directory Op Cm ...
.It Ic boot Fl flag Cm ...
Boot as specified by the
.Xr loader.conf 4
files read.
.Pp
Depending on the arguments passed, it can override boot flags and
either the kernel name or the search path for kernel and modules.
.Pp
.It Ic boot-conf
.It Ic boot-conf Ar kernelname Op Cm ...
.It Ic boot-conf Ar directory Op Cm ...
.It Ic boot-conf Fl flag Cm ...
Works like
.Ic boot
described above, but instead of booting immediately, uses
.Ic autoboot ,
so it can be stopped.
.Pp
.It Ic start
Reads
.Pa /boot/defaults/loader.conf ,
all other
.Xr loader.conf 4
files specified in it, then loads the desired kernel and modules
.Pq if not already loaded .
After which you can use the
.Ic boot
or
.Ic autoboot
commands or simply exit (provided
commmands or simply exit (provided
.Va autoboot_delay
is not set to NO) to boot the system.
.Ic start
is the command used in the default
.Pa /boot/loader.rc
file
.Pq see Xr loader 5 .
.Pp
.It Ic initialize
Initialize the support library so commands can be used without executing
.Ic start
first.
Like
.Ic start ,
it reads
.Pa /boot/defaults/loader.conf
and all other
.Xr loader.conf 4
files specified in it
.Pq but does not load kernel or modules .
Returns a flag on the stack to indicate
if any configuration files were successfully loaded.
.Pp
.It Ic read-conf Ar filename
Reads and processes a
```

```
127 .Xr loader.conf 4
128 file.
129 Does not proceed to boot.
130 .Pp
131 .It Ic enable-module Ar module
132 Enables the loading of
133 .Ar module .
134 .Pp
135 .It Ic disable-module Ar module
136 Disables the loading of
137 .Ar module .
138 .Pp
139 .It Ic toggle-module Ar module
140 Toggles the loading of
141 .Ar module
142 on and off.
143 .Pp
144 .It Ic show-module Ar module
145 Shows the information gathered in the
146 .Xr loader.conf 4
147 files about the module
148 .Ar module .
149 .Pp
150 .It Ic show-module-options
151 Shows all modules defined in current
152 .Xr loader.conf 4
153 configuration.
154 .Pp
155 .It Ic retry
156 Used inside
157 .Xr loader.conf 4
158 files to specify the action after a module loading fails.
159 .Pp
160 .It Ic ignore
161 Used inside
162 .Xr loader.conf 4
163 files to specify the action after a module loading fails.
164 .It Ic try-include Ar file Op Ar
165 Process script files if they exist.
166 Each file, in turn, is completely read into memory,
167 and then each of its lines is passed to the command line interpreter.
168 If any error is returned by the interpreter, the try-include
169 command aborts immediately, without reading any other files, and
170 silently returns without error.
171 .El
172 .Sh FILES
173 .Bl -tag -width /boot/forth/loader.4th -compact
174 .It Pa /boot/loader
175 The
176 .Xr loader 5 .
177 .It Pa /boot/forth/loader.4th
178 .Nm
179 itself.
180 .It Pa /boot/loader.rc
181 .Xr loader 5
182 bootstrapping script.
183 .It Pa /boot/defaults/loader.conf
184 File loaded by the
185 .Ic start
186 command.
187 .El
188 .Sh EXAMPLES
189 Standard
190 .Pa /boot/loader.rc :
191 .Pp
192 .Bd -literal -offset indent -compact
```



```
193 include /boot/forth/loader.4th
194 start
195 .Ed
196 .Pp
197 Read an additional configuration file and then proceed to boot:
198 .Pp
199 .Bd -literal -offset indent -compact
200 unload
201 read-conf /boot/special.conf
202 boot-conf
203 .Ed
204 .Sh SEE ALSO
205 .Xr loader.conf 4 ,
206 .Xr loader 5
```

```
**********************************************************
   53478 Sun Sep 16 19:22:57 2018
new/usr/src/man/man5/tecla.5
9842 man page typos and spelling
**********************************************************
'\" te
.\" Copyright (c) 2000, 2001, 2002, 2003, 2004 by Martin C. Shepherd. All Rights
.\" Permission is hereby granted, free of charge, to any person obtaining a copy
.\" "Software"), to deal in the Software without restriction, including
.\" without limitation the rights to use, copy, modify, merge, publish,
.\" distribute, and/or sell copies of the Software, and to permit persons
.\" to whom the Software is furnished to do so, provided that the above
.\" copyright notice(s) and this permission notice appear in all copies of
.\" the Software and that both the above copyright notice(s) and this
.\" permission notice appear in supporting documentation.
.\"
.\" THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS
.\" OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
.\" MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT
.\" OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
.\" HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR ANY SPECIAL
.\" INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING
.\" FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT,
.\" NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION
.\" WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
.\"
.\" Except as contained in this notice, the name of a copyright holder
.\" shall not be used in advertising or otherwise to promote the sale, use
.\" or other dealings in this Software without prior written authorization
.\" of the copyright holder.
.\" Portions Copyright (c) 2005, Sun Microsystems, Inc. All Rights Reserved.
.TH TECLA 5 "April 9, 2016"
.SH NAME
tecla, teclarc \- User interface provided by the tecla library.
.SH DESCRIPTION
.LP
This man page describes the command-line editing features that are available to
users of programs that read keyboard input via the tecla library. Users of the
\fBtcsh shell\fR will find the default key bindings very familiar. Users of the
\fBbash\fR shell will also find it quite familiar, but with a few minor
differences, most notably in how forward and backward searches through the list
of historical commands are performed. There are two major editing modes, one
with \fBemacs\fR-like key bindings and another with \fBvi\fR-like key bindings.
By default \fBemacs\fR mode is enabled, but \fBvi\fR(1) mode can alternatively
be selected via the user's configuration file. This file can also be used to
change the bindings of individual keys to suit the user's preferences. By
default, tab completion is provided. If the application hasn't reconfigured
this to complete other types of symbols, then tab completion completes file
names.
.SS "Key Sequence Notation"
.LP
In the rest of this man page, and also in all tecla configuration files, key
sequences are expressed as follows.
.sp
.ne 2
.na
\fB\fB^A\fR or \fBC-a\fR\fR
.ad
.RS 13n
This is a 'CONTROL-A', entered by pressing the CONTROL key at the same time as
the 'A' key.
.RE

.sp
.ne 2
.na
```



```
\fB\eE\fR or \fBM-\fR
.ad
.RS 13n
In key sequences, both of these notations can be entered either by pressing the
ESCAPE key, then the following key, or by pressing the META key at the same
time as the following key. Thus the key sequence \fBM-p\fR can be typed in two
ways, by pressing the ESCAPE key, followed by pressing 'P', or by pressing the
META key at the same time as 'P'.
.RE

.sp
.ne 2
.na
\fBup\fR
.ad
.RS 13n
This refers to the up-arrow key.
.RE

.sp
.ne 2
.na
\fBdown\fR
.ad
.RS 13n
This refers to the down-arrow key.
.RE

.sp
.ne 2
.na
\fBleft\fR
.ad
.RS 13n
This refers to the left-arrow key.
.RE

.sp
.ne 2
.na
\fBright\fR
.ad
.RS 13n
This refers to the right-arrow key.
.RE

.sp
.ne 2
.na
\fBa\fR
.ad
.RS 13n
This is just a normal 'A' key.
.RE

.SS "The Tecla Configuration File"
.LP
By default, tecla looks for a file called \fB\&.teclarc\fR in your home
directory (ie. \fB~/.teclarc\fR). If it finds this file, it reads it,
interpreting each line as defining a new key binding or an editing
configuration option. Since the \fBemacs\fR key-bindings are installed by
default, if you want to use the non-default \fBvi\fR editing mode, the most
important item to go in this file is the following line:
.sp
.in +2
.nf
```

```
edit-mode vi
.fi
.in -2

.sp
.LP
This will re-configure the default bindings for \fBvi\fR-mode. The complete set
of arguments that this command accepts are:
.sp
.ne 2
.na
\fBvi\fR
.ad
.RS 9n
Install key bindings like those of the \fBvi\fR editor.
.RE

.sp
.ne 2
.na
\fBemacs\fR
.ad
.RS 9n
Install key bindings like those of the \fBemacs\fR editor. This is the default.
.RE

.sp
.ne 2
.na
\fBnone\fR
.ad
.RS 9n
Use just the native line editing facilities provided by the terminal driver.
.RE

.sp
.LP
To prevent the terminal bell from being rung, such as when an unrecognized
control-sequence is typed, place the following line in the configuration file:
.sp
.in +2
.nf
nobeep
.fi
.in -2

.sp
.LP
An example of a key binding line in the configuration file is the following.
.sp
.in +2
.nf
bind M-[2~ insert-mode
.fi
.in -2

.sp
.LP
On many keyboards, the above key sequence is generated when one presses the
insert key, so with this key binding, one can toggle between the
\fBemacs\fR-mode insert and overwrite modes by hitting one key. One could also
do it by typing out the above sequence of characters one by one. As explained
above, the \fBM-\fR part of this sequence can be typed either by pressing the
ESCAPE key before the following key, or by pressing the META key at the same
time as the following key. Thus if you had set the above key binding, and the
insert key on your keyboard didn't generate the above key sequence, you could
```



```
still type it in either of the following 2 ways.
.RS +4
.TP
1.
Hit the ESCAPE key momentarily, then press '[', then '2', then finally '~'.
.RE
.RS +4
.TP
2.
Press the META key at the same time as pressing the '[' key, then press '2',
then '~'.
.RE
.sp
.LP
If you set a key binding for a key sequence that is already bound to a
function, the new binding overrides the old one. If in the new binding you omit
the name of the new function to bind to the key sequence, the original binding
becomes undefined.
.sp
.LP
Starting with versions of \fBlibtecla\fR later than 1.3.3 it is now possible to
bind key sequences that begin with a printable character. Previously key
sequences were required to start with a CONTROL or META character.
.sp
.LP
Note that the special keywords "up", "down", "left", and "right" refer to the
arrow keys, and are thus not treated as key sequences. So, for example, to
rebind the up and down arrow keys to use the history search mechanism instead
of the simple history recall method, you could place the following in your
configuration file:
.sp
.in +2
.nf
bind up history-search-backwards
bind down history-search-backwards
.fi
.in -2

.sp
.LP
To unbind an existing binding, you can do this with the bind command by
omitting to name any action to rebind the key sequence to. For example, by not
specifying an action function, the following command unbinds the default
beginning-of-line action from the \fB^A\fR key sequence:
.sp
.in +2
.nf
bind ^A
.fi
.in -2

.sp
.LP
If you create a \fB~/.teclarc\fR configuration file, but it appears to have no
effect on the program, check the documentation of the program to see if the
author chose a different name for this file.
.SS "Filename and Tilde Completion"
.LP
With the default key bindings, pressing the TAB key (aka. \fB^I\fR) results in
tecla attempting to complete the incomplete file name that precedes the cursor.
Tecla searches backwards from the cursor, looking for the start of the file
name, stopping when it hits either a space or the start of the line. If more
than one file has the specified prefix, then tecla completes the file name up
to the point at which the ambiguous matches start to differ, then lists the
possible matches.
.sp
```

```
260 .LP
261 In addition to literally written file names, tecla can complete files that
262 start with \fB~/\fR and \fB~user/\fR expressions and that contain \fB$envvar\fR
263 expressions. In particular, if you hit TAB within an incomplete \fB~user\fR,
264 expression, tecla will attempt to complete the username, listing any ambiguous
265 matches.
266 .sp
267 .LP
268 The completion binding is implemented using the \fBcpl_complete_word()\fR
269 function, which is also available separately to users of this library. See the
270 \fBcpl_complete_word\fR(3TECLA) man page for more details.
271 .SS "Filename Expansion"
272 .LP
273 With the default key bindings, pressing \fB^X*\fR causes tecla to expand the
274 file name that precedes the cursor, replacing \fB~/\fR and \fB~user/\fR
275 expressions with the corresponding home directories, and replacing
276 \fB$envvar\fR expressions with the value of the specified environment variable,
277 then if there are any wildcards, replacing the so far expanded file name with a
278 space-separated list of the files which match the wild cards.
279 .sp
280 .LP
281 The expansion binding is implemented using the \fBef_expand_file()\fR function.
282 See the \fBef_expand_file\fR(3TECLA) man page for more details.
283 .SS "Recalling Previously Typed Lines"
284 .LP
285 Every time that a new line is entered by the user, it is appended to a list of
286 historical input lines maintained within the \fBGetLine\fR resource object. You
287 can traverse up and down this list using the up and down arrow keys.
288 Alternatively, you can do the same with the \fB^P\fR, and \fB^N\fR keys, and in
289 \fBvi\fR command mode you can alternatively use the k and j characters. Thus
290 pressing up-arrow once, replaces the current input line with the previously
291 entered line. Pressing up-arrow again, replaces this with the line that was
292 entered before it, etc.. Having gone back one or more lines into the history
293 list, one can return to newer lines by pressing down-arrow one or more times.
294 If you do this sufficient times, you will return to the original line that you
295 were entering when you first hit up-arrow.
296 .sp
297 .LP
298 Note that in \fBvi\fR mode, all of the history recall functions switch the
299 library into command mode.
300 .sp
301 .LP
302 In \fBemacs\fR mode the \fBM-p\fR and \fBM-n\fR keys work just like the
303 \fB^P\fR and \fB^N\fR keys, except that they skip all but those historical
304 lines which share the prefix that precedes the cursor. In \fBvi\fR command mode
305 the upper case 'K' and 'J' characters do the same thing, except that the string
306 that they search for includes the character under the cursor as well as what
307 precedes it.
308 .sp
309 .LP
310 Thus for example, suppose that you were in \fBemacs\fR mode, and you had just
311 entered the following list of commands in the order shown:
312 .sp
313 .in +2
314 .nf
315 ls ~/tecla/
316 cd ~/tecla
317 ls -l getline.c
318 \fBemacs\fR ~/tecla/getline.c
319 .fi
320 .in -2

322 .sp
323 .LP
324 If you next typed:
325 .sp
```



```
326 .in +2
327 .nf
328 ls
329 .fi
330 .in -2

332 .sp
333 .LP
334 and then hit \fBM-p\fR, then rather than returning the previously typed
335 \fBemacs\fR line, which doesn't start with "ls", tecla would recall the "ls -l
336 getline.c" line. Pressing \fBM-p\fR again would recall the "ls ~/tecla/" line.
337 .sp
338 .LP
339 Note that if the string that you are searching for, contains any of the special
340 characters, *, ?, or '[', then it is interpreted as a pattern to be matched.
341 Thus, continuing with the above example, after typing in the list of commands
340 characters, *, ?, or '[', then it is interpretted as a pattern to be matched.
341 Thus, cotinuing with the above example, after typing in the list of commands
342 shown, if you then typed:
343 .sp
344 .in +2
345 .nf
346 *tecla*
347 .fi
348 .in -2

350 .sp
351 .LP
352 and hit \fBM-p\fR, then the "\fBemacs\fR ~/tecla/getline.c" line would be
353 recalled first, since it contains the word tecla somewhere in the line,
354 Similarly, hitting \fBM-p\fR again, would recall the "ls ~/tecla/" line, and
355 hitting it once more would recall the "ls ~/tecla/" line. The pattern syntax is
356 the same as that described for file name expansion, in the
357 \fBef_expand_file\fR(3TECLA).
358 .SS "History Files"
359 .LP
360 Authors of programs that use the tecla library have the option of saving
361 historical command-lines in a file before exiting, and subsequently reading
362 them back in from this file when the program is next started. There is no
363 standard name for this file, since it makes sense for each application to use
364 its own history file, so that commands from different applications don't get
365 mixed up.
366 .SS "International Character Sets"
367 .LP
368 Since \fBlibtecla\fR version 1.4.0, tecla has been 8-bit clean. This means that
369 all 8-bit characters that are printable in the user's current locale are now
370 displayed verbatim and included in the returned input line. Assuming that the
371 calling program correctly contains a call like the following,
372 .sp
373 .in +2
374 .nf
375 setlocale(LC_CTYPE, "");
376 .fi
377 .in -2

379 .sp
380 .LP
381 then the current locale is determined by the first of the environment variables
382 \fBLC_CTYPE\fR, \fBLC_ALL\fR, and \fBLANG\fR, that is found to contain a valid
383 locale name. If none of these variables are defined, or the program neglects to
384 call \fBsetlocale\fR, then the default C locale is used, which is US 7-bit
385 ASCII. On most unix-like platforms, you can get a list of valid locales by
386 typing the command:
387 .sp
388 .in +2
389 .nf
```

```
locale -a
.fi
.in -2

.sp
.LP
at the shell prompt.
.SS "Meta Keys and Locales"
.LP
Beware that in most locales other than the default C locale, META characters
become printable, and they are then no longer considered to match \fBM-c\fR
style key bindings. This allows international characters to be entered with the
compose key without unexpectedly triggering META key bindings. You can still
invoke META bindings, since there are actually two ways to do this. For example
the binding \fBM-c\fR can also be invoked by pressing the ESCAPE key
momentarily, then pressing the c key, and this will work regardless of locale.
Moreover, many modern terminal emulators, such as gnome's gnome-terminal's and
KDE's konsole terminals, already generate escape pairs like this when you use
the META key, rather than a real meta character, and other emulators usually
have a way to request this behavior, so you can continue to use the META key on
most systems.
.sp
.LP
For example, although xterm terminal emulators generate real 8-bit meta
characters by default when you use the META key, they can be configured to
output the equivalent escape pair by setting their \fBEightBitInput\fR X
resource to False. You can either do this by placing a line like the following
in your \fB~/.Xdefaults\fR file,
.sp
.in +2
.nf
XTerm*EightBitInput: False
.fi
.in -2

.sp
.LP
or by starting an \fBxterm\fR with an \fB-xrm\fR \&'*EightBitInput: False'
command-line argument. In recent versions of xterm you can toggle this feature
on and off with the 'Meta Sends Escape' option in the menu that is displayed
when you press the left mouse button and the CONTROL key within an xterm
window. In CDE, dtterms can be similarly coerced to generate escape pairs in
place of meta characters, by setting the \fBDtterm*KshMode\fR resource to True.
.SS "Entering International Characters"
.LP
If you don't have a keyboard that generates all of the international characters
that you need, there is usually a compose key that will allow you to enter
special characters, or a way to create one. For example, under X windows on
unix-like systems, if your keyboard doesn't have a compose key, you can
designate a redundant key to serve this purpose with the xmodmap command. For
example, on many PC keyboards there is a microsoft-windows key, which is
otherwise useless under Linux. On a laptop, for example, the \fBxev\fR program
might report that pressing this key generates keycode 115. To turn this key
into a COMPOSE  key, do the following:
.sp
.in +2
.nf
xmodmap -e 'keycode 115 = Multi_key'
.fi
.in -2

.sp
.LP
Type this key followed by a " character to enter an 'I' with a umlaut over it.
.SS "The Available Key Binding Functions"
.LP
```



```
The following is a list of the editing functions provided by the tecla library.
The names in the leftmost column of the list can be used in configuration files
to specify which function a given key or combination of keys should invoke.
They are also used in the next two sections to list the default key bindings in
\fBemacs\fR and \fBvi\fR modes.
.sp
.ne 2
.na
\fBuser-interrupt\fR
.ad
.RS 30n
Send a SIGINT signal to the parent process.
.RE

.sp
.ne 2
.na
\fBsuspend\fR
.ad
.RS 30n
Suspend the parent process.
.RE

.sp
.ne 2
.na
\fBstop-output\fR
.ad
.RS 30n
Pause terminal output.
.RE

.sp
.ne 2
.na
\fBstart-output\fR
.ad
.RS 30n
Resume paused terminal output.
.RE

.sp
.ne 2
.na
\fBliteral-next\fR
.ad
.RS 30n
Arrange for the next character to be treated as a normal character. This allows
control characters to be entered.
.RE

.sp
.ne 2
.na
\fBcursor-right\fR
.ad
.RS 30n
Move the cursor one character right.
.RE

.sp
.ne 2
.na
\fBcursor-left\fR
.ad
.RS 30n
```

```
522 Move the cursor one character left.
523 .RE

525 .sp
526 .ne 2
527 .na
528 \fBinsert-mode\fR
529 .ad
530 .RS 30n
531 Toggle between insert mode and overwrite mode.
532 .RE

534 .sp
535 .ne 2
536 .na
537 \fBbeginning-of-line\fR
538 .ad
539 .RS 30n
540 Move the cursor to the beginning of the line.
541 .RE

543 .sp
544 .ne 2
545 .na
546 \fBend-of-line\fR
547 .ad
548 .RS 30n
549 Move the cursor to the end of the line.
550 .RE

552 .sp
553 .ne 2
554 .na
555 \fBdelete-line\fR
556 .ad
557 .RS 30n
558 Delete the contents of the current line.
559 .RE

561 .sp
562 .ne 2
563 .na
564 \fBkill-line\fR
565 .ad
566 .RS 30n
567 Delete everything that follows the cursor.
568 .RE

570 .sp
571 .ne 2
572 .na
573 \fBbackward-kill-line\fR
574 .ad
575 .RS 30n
576 Delete all characters between the cursor and the start of the line.
577 .RE

579 .sp
580 .ne 2
581 .na
582 \fBforward-word\fR
583 .ad
584 .RS 30n
585 Move to the end of the word which follows the cursor.
586 .RE
```



```
588 .sp
589 .ne 2
590 .na
591 \fBforward-to-word\fR
592 .ad
593 .RS 30n
594 Move the cursor to the start of the word that follows the cursor.
595 .RE

597 .sp
598 .ne 2
599 .na
600 \fBbackward-word\fR
601 .ad
602 .RS 30n
603 Move to the start of the word which precedes the cursor.
604 .RE

606 .sp
607 .ne 2
608 .na
609 \fBgoto-column\fR
610 .ad
611 .RS 30n
612 Move the cursor to the 1-relative column in the line specified by any preceding
613 digit-argument sequences (see Entering Repeat Counts below).
614 .RE

616 .sp
617 .ne 2
618 .na
619 \fBfind-parenthesis\fR
620 .ad
621 .RS 30n
622 If the cursor is currently over a parenthesis character, move it to the
623 matching parenthesis character. If not over a parenthesis character move right
624 to the next close parenthesis.
625 .RE

627 .sp
628 .ne 2
629 .na
630 \fBforward-delete-char\fR
631 .ad
632 .RS 30n
633 Delete the character under the cursor.
634 .RE

636 .sp
637 .ne 2
638 .na
639 \fBbackward-delete-char\fR
640 .ad
641 .RS 30n
642 Delete the character which precedes the cursor.
643 .RE

645 .sp
646 .ne 2
647 .na
648 \fBlist-or-eof\fR
649 .ad
650 .RS 30n
651 This is intended for binding to \fB^D\fR. When invoked when the cursor is
652 within the line it displays all possible completions then redisplays the line
653 unchanged. When invoked on an empty line, it signals end-of-input (EOF) to the
```

```
654 caller of \fBgl_get_line()\fR.
655 .RE

657 .sp
658 .ne 2
659 .na
660 \fBdel-char-or-list-or-eof\fR
661 .ad
662 .RS 30n
663 This is intended for binding to \fB^D\fR. When invoked when the cursor is
664 within the line it invokes forward-delete-char. When invoked at the end of the
665 line it displays all possible completions then redisplays the line unchanged.
666 When invoked on an empty line, it signals end-of-input (EOF) to the caller of
667 \fBgl_get_line()\fR.
668 .RE

670 .sp
671 .ne 2
672 .na
673 \fBforward-delete-word\fR
674 .ad
675 .RS 30n
676 Delete the word which follows the cursor.
677 .RE

679 .sp
680 .ne 2
681 .na
682 \fBbackward-delete-word\fR
683 .ad
684 .RS 30n
685 Delete the word which precedes the cursor.
686 .RE

688 .sp
689 .ne 2
690 .na
691 \fBupcase-word\fR
692 .ad
693 .RS 30n
694 Convert all of the characters of the word which follows the cursor, to upper
695 case.
696 .RE

698 .sp
699 .ne 2
700 .na
701 \fBdowncase-word\fR
702 .ad
703 .RS 30n
704 Convert all of the characters of the word which follows the cursor, to lower
705 case.
706 .RE

708 .sp
709 .ne 2
710 .na
711 \fBcapitalize-word\fR
712 .ad
713 .RS 30n
714 Capitalize the word which follows the cursor.
715 .RE

717 .sp
718 .ne 2
719 .na
```



```
720 \fBchange-case\fR
721 .ad
722 .RS 30n
723 If the next character is upper case, toggle it to lower case and vice versa.
724 .RE

726 .sp
727 .ne 2
728 .na
729 \fBredisplay\fR
730 .ad
731 .RS 30n
732 Redisplay the line.
733 .RE

735 .sp
736 .ne 2
737 .na
738 \fBclear-screen\fR
739 .ad
740 .RS 30n
741 Clear the terminal, then redisplay the current line.
742 .RE

744 .sp
745 .ne 2
746 .na
747 \fBtranspose-chars\fR
748 .ad
749 .RS 30n
750 Swap the character under the cursor with the character just before the cursor.
751 .RE

753 .sp
754 .ne 2
755 .na
756 \fBset-mark\fR
757 .ad
758 .RS 30n
759 Set a mark at the position of the cursor.
760 .RE

762 .sp
763 .ne 2
764 .na
765 \fBexchange-point-and-mark\fR
766 .ad
767 .RS 30n
768 Move the cursor to the last mark that was set, and move the mark to where the
769 cursor used to be.
770 .RE

772 .sp
773 .ne 2
774 .na
775 \fBkill-region\fR
776 .ad
777 .RS 30n
778 Delete the characters that lie between the last mark that was set, and the
779 cursor.
780 .RE

782 .sp
783 .ne 2
784 .na
785 \fBcopy-region-as-kill\fR
```

```
.ad
.RS 30n
Copy the text between the mark and the cursor to the cut buffer, without
deleting the original text.
.RE

.sp
.ne 2
.na
\fByank\fR
.ad
.RS 30n
Insert the text that was last deleted, just before the current position of the
cursor.
.RE

.sp
.ne 2
.na
\fBappend-yank\fR
.ad
.RS 30n
Paste the current contents of the cut buffer, after the cursor.
.RE

.sp
.ne 2
.na
\fBup-history\fR
.ad
.RS 30n
Recall the next oldest line that was entered. Note that in \fBvi\fR mode you
are left in command mode.
.RE

.sp
.ne 2
.na
\fBdown-history\fR
.ad
.RS 30n
Recall the next most recent line that was entered. If no history recall session
is currently active, the next line from a previous recall session is recalled.
Note that in vi mode you are left in command mode.
.RE

.sp
.ne 2
.na
\fBhistory-search-backward\fR
.ad
.RS 30n
Recall the next oldest line who’s prefix matches the string which currently
precedes the cursor (in \fBvi\fR command-mode the character under the cursor is
also included in the search string). Note that in \fBvi\fR mode you are left in
command mode.
.RE

.sp
.ne 2
.na
\fBhistory-search-forward\fR
.ad
.RS 30n
Recall the next newest line who’s prefix matches the string which currently
precedes the cursor (in \fBvi\fR command-mode the character under the cursor is
```



```
also included in the search string). Note that in \fBvi\fR mode you are left in
command mode.
.RE

.sp
.ne 2
.na
\fBhistory-re-search-backward\fR
.ad
.RS 30n
Recall the next oldest line who’s prefix matches that established by the last
invocation of either history-search-forward or history-search-backward.
.RE

.sp
.ne 2
.na
\fBhistory-re-search-forward\fR
.ad
.RS 30n
Recall the next newest line who’s prefix matches that established by the last
invocation of either history-search-forward or history-search-backward.
.RE

.sp
.ne 2
.na
\fBcomplete-word\fR
.ad
.RS 30n
Attempt to complete the incomplete word which precedes the cursor. Unless the
host program has customized word completion, file name completion is attempted.
In \fBvi\fR command mode the character under the cursor is also included in
In \fBvi\fR commmand mode the character under the cursor is also included in
the word being completed, and you are left in \fBvi\fR insert mode.
.RE

.sp
.ne 2
.na
\fBexpand-filename\fR
.ad
.RS 30n
Within the command line, expand wild cards, tilde expressions and dollar
expressions in the file name which immediately precedes the cursor. In \fBvi\fR
command mode the character under the cursor is also included in the file name
commmand mode the character under the cursor is also included in the file name
being expanded, and you are left in \fBvi\fR insert mode.
.RE

.sp
.ne 2
.na
\fBlist-glob\fR
.ad
.RS 30n
List any file names which match the wild-card, tilde and dollar expressions in
the file name which immediately precedes the cursor, then redraw the input line
unchanged.
.RE

.sp
.ne 2
.na
\fBlist-history\fR
.ad
```

```
 916 .RS 30n
 917 Display the contents of the history list for the current history group. If a
 918 repeat count of \fB> 1\fR is specified, only that many of the most recent lines
 919 are displayed. See the Entering Repeat Counts section.
 920 .RE

 922 .sp
 923 .ne 2
 924 .na
 925 \fBread-from-file\fR
 926 .ad
 927 .RS 30n
 928 Temporarily switch to reading input from the file who’s name precedes the
 929 cursor.
 930 .RE

 932 .sp
 933 .ne 2
 934 .na
 935 \fBread-init-files\fR
 936 .ad
 937 .RS 30n
 938 Re-read \fBteclarc\fR configuration files.
 939 .RE

 941 .sp
 942 .ne 2
 943 .na
 944 \fBbeginning-of-history\fR
 945 .ad
 946 .RS 30n
 947 Move to the oldest line in the history list. Note that in \fBvi\fR mode you are
 948 left in command mode.
 949 .RE

 951 .sp
 952 .ne 2
 953 .na
 954 \fBend-of-history\fR
 955 .ad
 956 .RS 30n
 957 Move to the newest line in the history list (ie. the current line). Note that
 958 in \fBvi\fR mode this leaves you in command mode.
 959 .RE

 961 .sp
 962 .ne 2
 963 .na
 964 \fBdigit-argument\fR
 965 .ad
 966 .RS 30n
 967 Enter a repeat count for the next key binding function. For details, see the
 968 Entering Repeat Counts section.
 969 .RE

 971 .sp
 972 .ne 2
 973 .na
 974 \fBnewline\fR
 975 .ad
 976 .RS 30n
 977 Terminate and return the current contents of the line, after appending a
 978 newline character. The newline character is normally ’\en’, but will be the
 979 first character of the key sequence that invoked the newline action, if this
 980 happens to be a printable character. If the action was invoked by the ’\en’
 981 newline character or the ’\er’ carriage return character, the line is appended
```



```
 982 to the history buffer.
 983 .RE

 985 .sp
 986 .ne 2
 987 .na
 988 \fBrepeat-history\fR
 989 .ad
 990 .RS 30n
 991 Return the line that is being edited, then arrange for the next most recent
 992 entry in the history buffer to be recalled when tecla is next called.
 993 Repeatedly invoking this action causes successive historical input lines to be
 994 re-executed. Note that this action is equivalent to the ’Operate’ action in
 995 ksh.
 996 .RE

 998 .sp
 999 .ne 2
1000 .na
1001 \fBring-bell\fR
1002 .ad
1003 .RS 30n
1004 Ring the terminal bell, unless the bell has been silenced via the nobeep
1005 configuration option (see The Tecla Configuration File section).
1006 .RE

1008 .sp
1009 .ne 2
1010 .na
1011 \fBforward-copy-char\fR
1012 .ad
1013 .RS 30n
1014 Copy the next character into the cut buffer (NB. use repeat counts to copy more
1015 than one).
1016 .RE

1018 .sp
1019 .ne 2
1020 .na
1021 \fBbackward-copy-char\fR
1022 .ad
1023 .RS 30n
1024 Copy the previous character into the cut buffer.
1025 .RE

1027 .sp
1028 .ne 2
1029 .na
1030 \fBforward-copy-word\fR
1031 .ad
1032 .RS 30n
1033 Copy the next word into the cut buffer.
1034 .RE

1036 .sp
1037 .ne 2
1038 .na
1039 \fBbackward-copy-word\fR
1040 .ad
1041 .RS 30n
1042 Copy the previous word into the cut buffer.
1043 .RE

1045 .sp
1046 .ne 2
1047 .na
```

```
\fBforward-find-char\fR
.ad
.RS 30n
Move the cursor to the next occurrence of the next character that you type.
.RE

.sp
.ne 2
.na
\fBbackward-find-char\fR
.ad
.RS 30n
Move the cursor to the last occurrence of the next character that you type.
.RE

.sp
.ne 2
.na
\fBforward-to-char\fR
.ad
.RS 30n
Move the cursor to the character just before the next occurrence of the next
character that the user types.
.RE

.sp
.ne 2
.na
\fBbackward-to-char\fR
.ad
.RS 30n
Move the cursor to the character just after the last occurrence before the
cursor of the next character that the user types.
.RE

.sp
.ne 2
.na
\fBrepeat-find-char\fR
.ad
.RS 30n
Repeat the last backward-find-char, forward-find-char, backward-to-char or
forward-to-char.
.RE

.sp
.ne 2
.na
\fBinvert-refind-char\fR
.ad
.RS 30n
Repeat the last backward-find-char, forward-find-char, backward-to-char, or
forward-to-char in the opposite direction.
.RE

.sp
.ne 2
.na
\fBdelete-to-column\fR
.ad
.RS 30n
Delete the characters from the cursor up to the column that is specified by the
repeat count.
.RE

.sp
```



```
.ne 2
.na
\fBdelete-to-parenthesis\fR
.ad
.RS 30n
Delete the characters from the cursor up to and including the matching
parenthesis, or next close parenthesis.
.RE

.sp
.ne 2
.na
\fBforward-delete-find\fR
.ad
.RS 30n
Delete the characters from the cursor up to and including the following
occurrence of the next character typed.
.RE

.sp
.ne 2
.na
\fBbackward-delete-find\fR
.ad
.RS 30n
Delete the characters from the cursor up to and including the preceding
occurrence of the next character typed.
.RE

.sp
.ne 2
.na
\fBforward-delete-to\fR
.ad
.RS 30n
Delete the characters from the cursor up to, but not including, the following
occurrence of the next character typed.
.RE

.sp
.ne 2
.na
\fBbackward-delete-to\fR
.ad
.RS 30n
Delete the characters from the cursor up to, but not including, the preceding
occurrence of the next character typed.
.RE

.sp
.ne 2
.na
\fBdelete-refind\fR
.ad
.RS 30n
Repeat the last *-delete-find or *-delete-to action.
.RE

.sp
.ne 2
.na
\fBdelete-invert-refind\fR
.ad
.RS 30n
Repeat the last *-delete-find or *-delete-to action, in the opposite direction.
.RE
```

```
.sp
.ne 2
.na
\fBcopy-to-column\fR
.ad
.RS 30n
Copy the characters from the cursor up to the column that is specified by the
repeat count, into the cut buffer.
.RE

.sp
.ne 2
.na
\fBcopy-to-parenthesis\fR
.ad
.RS 30n
Copy the characters from the cursor up to and including the matching
parenthesis, or next close parenthesis, into the cut buffer.
.RE

.sp
.ne 2
.na
\fBforward-copy-find\fR
.ad
.RS 30n
Copy the characters from the cursor up to and including the following occurrence
of the next character typed, into the cut buffer.
.RE

.sp
.ne 2
.na
\fBbackward-copy-find\fR
.ad
.RS 30n
Copy the characters from the cursor up to and including the preceding occurrence
of the next character typed, into the cut buffer.
.RE

.sp
.ne 2
.na
\fBforward-copy-to\fR
.ad
.RS 30n
Copy the characters from the cursor up to, but not including, the following
occurrence of the next character typed, into the cut buffer.
.RE

.sp
.ne 2
.na
\fBbackward-copy-to\fR
.ad
.RS 30n
Copy the characters from the cursor up to, but not including, the preceding
occurrence of the next character typed, into the cut buffer.
.RE

.sp
.ne 2
.na
\fBcopy-refind\fR
.ad
```



```
.RS 30n
Repeat the last *-copy-find or *-copy-to action.
.RE

.sp
.ne 2
.na
\fBcopy-invert-refind\fR
.ad
.RS 30n
Repeat the last *-copy-find or *-copy-to action, in the opposite direction.
.RE

.sp
.ne 2
.na
\fBvi-mode\fR
.ad
.RS 30n
Switch to \fBvi\fR mode from emacs mode.
.RE

.sp
.ne 2
.na
\fBemacs-mode\fR
.ad
.RS 30n
Switch to \fBemacs\fR mode from \fBvi\fR mode.
.RE

.sp
.ne 2
.na
\fBvi-insert\fR
.ad
.RS 30n
From \fBvi\fR command mode, switch to insert mode.
.RE

.sp
.ne 2
.na
\fBvi-overwrite\fR
.ad
.RS 30n
From \fBvi\fR command mode, switch to overwrite mode.
.RE

.sp
.ne 2
.na
\fBvi-insert-at-bol\fR
.ad
.RS 30n
From \fBvi\fR command mode, move the cursor to the start of the line and switch
to insert mode.
.RE

.sp
.ne 2
.na
\fBvi-append-at-eol\fR
.ad
.RS 30n
From \fBvi\fR command mode, move the cursor to the end of the line and switch
```

```
1312 to append mode.
1313 .RE

1315 .sp
1316 .ne 2
1317 .na
1318 \fBvi-append\fR
1319 .ad
1320 .RS 30n
1321 From \fBvi\fR command mode, move the cursor one position right, and switch to
1322 insert mode.
1323 .RE

1325 .sp
1326 .ne 2
1327 .na
1328 \fBvi-replace-char\fR
1329 .ad
1330 .RS 30n
1331 From \fBvi\fR command mode, replace the character under the cursor with the
1332 next character entered.
1333 .RE

1335 .sp
1336 .ne 2
1337 .na
1338 \fBvi-forward-change-char\fR
1339 .ad
1340 .RS 30n
1341 From \fBvi\fR command mode, delete the next character then enter insert mode.
1342 .RE

1344 .sp
1345 .ne 2
1346 .na
1347 \fBvi-backward-change-char\fR
1348 .ad
1349 .RS 30n
1350 From vi command mode, delete the preceding character then enter insert mode.
1351 .RE

1353 .sp
1354 .ne 2
1355 .na
1356 \fBvi-forward-change-word\fR
1357 .ad
1358 .RS 30n
1359 From \fBvi\fR command mode, delete the next word then enter insert mode.
1360 .RE

1362 .sp
1363 .ne 2
1364 .na
1365 \fBvi-backward-change-word\fR
1366 .ad
1367 .RS 30n
1368 From vi command mode, delete the preceding word then enter insert mode.
1369 .RE

1371 .sp
1372 .ne 2
1373 .na
1374 \fBvi-change-rest-of-line\fR
1375 .ad
1376 .RS 30n
1377 From \fBvi\fR command mode, delete from the cursor to the end of the line, then
```



```
1378 enter insert mode.
1379 .RE

1381 .sp
1382 .ne 2
1383 .na
1384 \fBvi-change-line\fR
1385 .ad
1386 .RS 30n
1387 From \fBvi\fR command mode, delete the current line, then enter insert mode.
1388 .RE

1390 .sp
1391 .ne 2
1392 .na
1393 \fBvi-change-to-bol\fR
1394 .ad
1395 .RS 30n
1396 From \fBvi\fR command mode, delete all characters between the cursor and the
1397 beginning of the line, then enter insert mode.
1398 .RE

1400 .sp
1401 .ne 2
1402 .na
1403 \fBvi-change-to-column\fR
1404 .ad
1405 .RS 30n
1406 From \fBvi\fR command mode, delete the characters from the cursor up to the
1407 column that is specified by the repeat count, then enter insert mode.
1408 .RE

1410 .sp
1411 .ne 2
1412 .na
1413 \fBvi-change-to-parenthesis\fR
1414 .ad
1415 .RS 30n
1416 Delete the characters from the cursor up to and including the matching
1417 parenthesis, or next close parenthesis, then enter \fBvi\fR insert mode.
1418 .RE

1420 .sp
1421 .ne 2
1422 .na
1423 \fBvi-forward-change-find\fR
1424 .ad
1425 .RS 30n
1426 From \fBvi\fR command mode, delete the characters from the cursor up to and
1427 including the following occurrence of the next character typed, then enter
1428 insert mode.
1429 .RE

1431 .sp
1432 .ne 2
1433 .na
1434 \fBvi-backward-change-find\fR
1435 .ad
1436 .RS 30n
1437 From vi command mode, delete the characters from the cursor up to and including
1438 the preceding occurrence of the next character typed, then enter insert mode.
1439 .RE

1441 .sp
1442 .ne 2
1443 .na
```

```
\fBvi-forward-change-to\fR
.ad
.RS 30n
From \fBvi\fR command mode, delete the characters from the cursor up to, but
not including, the following occurrence of the next character typed, then enter
insert mode.
.RE

.sp
.ne 2
.na
\fBvi-backward-change-to\fR
.ad
.RS 30n
From \fBvi\fR command mode, delete the characters from the cursor up to, but
not including, the preceding occurrence of the next character typed, then enter
insert mode.
.RE

.sp
.ne 2
.na
\fBvi-change-refind\fR
.ad
.RS 30n
Repeat the last vi-*-change-find or vi-*-change-to action.
.RE

.sp
.ne 2
.na
\fBvi-change-invert-refind\fR
.ad
.RS 30n
Repeat the last vi-*-change-find or vi-*-change-to action, in the opposite
direction.
.RE

.sp
.ne 2
.na
\fBvi-undo\fR
.ad
.RS 30n
In \fBvi\fR mode, undo the last editing operation.
.RE

.sp
.ne 2
.na
\fBvi-repeat-change\fR
.ad
.RS 30n
In \fBvi\fR command mode, repeat the last command that modified the line.
.RE

.SS "Default Key Bindings In \fBemacs\fR Mode"
.LP
The following default key bindings, which can be overriden by the tecla
configuration file, are designed to mimic most of the bindings of the unix
\fBtcsh shell\fR shell, when it is in \fBemacs\fR editing mode.
.sp
.LP
This is the default editing mode of the tecla library.
.sp
.LP
```



```
Under UNIX the terminal driver sets a number of special keys for certain
functions. The tecla library attempts to use the same key bindings to maintain
consistency. The key sequences shown for the following 6 bindings are thus just
examples of what they will probably be set to. If you have used the stty
command to change these keys, then the default bindings should match.
.sp
.ne 2
.na
\fB\fB^C\fR\fR
.ad
.RS 6n
user-interrupt
.RE

.sp
.ne 2
.na
\fB^\e\fR
.ad
.RS 6n
abort
.RE

.sp
.ne 2
.na
\fB\fB^Z\fR\fR
.ad
.RS 6n
suspend
.RE

.sp
.ne 2
.na
\fB\fB^Q\fR\fR
.ad
.RS 6n
start-output
.RE

.sp
.ne 2
.na
\fB\fB^S\fR\fR
.ad
.RS 6n
stop-output
.RE

.sp
.ne 2
.na
\fB\fB^V\fR\fR
.ad
.RS 6n
literal-next
.RE

.sp
.LP
The cursor keys are referred to by name, as follows. This is necessary because
different types of terminals generate different key sequences when their cursor
keys are pressed.
.sp
.ne 2
```

```
1576 .na
1577 \fBright\fR
1578 .ad
1579 .RS 9n
1580 cursor-right
1581 .RE

1583 .sp
1584 .ne 2
1585 .na
1586 \fBleft\fR
1587 .ad
1588 .RS 9n
1589 cursor-left
1590 .RE

1592 .sp
1593 .ne 2
1594 .na
1595 \fBup\fR
1596 .ad
1597 .RS 9n
1598 up-history
1599 .RE

1601 .sp
1602 .ne 2
1603 .na
1604 \fBdown\fR
1605 .ad
1606 .RS 9n
1607 down-history
1608 .RE

1610 .sp
1611 .LP
1612 The remaining bindings don't depend on the terminal settings.
1612 The remaining bindings don't depend on the terminal setttings.
1613 .sp
1614 .ne 2
1615 .na
1616 \fB\fB^F\fR\fR
1617 .ad
1618 .RS 21n
1619 cursor-right
1620 .RE

1622 .sp
1623 .ne 2
1624 .na
1625 \fB\fB^B\fR\fR
1626 .ad
1627 .RS 21n
1628 cursor-left
1629 .RE

1631 .sp
1632 .ne 2
1633 .na
1634 \fB\fBM-i\fR\fR
1635 .ad
1636 .RS 21n
1637 insert-mode
1638 .RE

1640 .sp
```



```
1641 .ne 2
1642 .na
1643 \fB\fB^A\fR\fR
1644 .ad
1645 .RS 21n
1646 beginning-of-line
1647 .RE

1649 .sp
1650 .ne 2
1651 .na
1652 \fB\fB^E\fR\fR
1653 .ad
1654 .RS 21n
1655 end-of-line
1656 .RE

1658 .sp
1659 .ne 2
1660 .na
1661 \fB\fB^U\fR\fR
1662 .ad
1663 .RS 21n
1664 delete-line
1665 .RE

1667 .sp
1668 .ne 2
1669 .na
1670 \fB\fB^K\fR\fR
1671 .ad
1672 .RS 21n
1673 kill-line
1674 .RE

1676 .sp
1677 .ne 2
1678 .na
1679 \fB\fBM-f\fR\fR
1680 .ad
1681 .RS 21n
1682 forward-word
1683 .RE

1685 .sp
1686 .ne 2
1687 .na
1688 \fB\fBM-b\fR\fR
1689 .ad
1690 .RS 21n
1691 backward-word
1692 .RE

1694 .sp
1695 .ne 2
1696 .na
1697 \fB\fB^D\fR\fR
1698 .ad
1699 .RS 21n
1700 del-char-or-list-or-eof
1701 .RE

1703 .sp
1704 .ne 2
1705 .na
1706 \fB\fB^H\fR\fR
```

```
1707 .ad
1708 .RS 21n
1709 backward-delete-char
1710 .RE

1712 .sp
1713 .ne 2
1714 .na
1715 \fB\fB^?\fR\fR
1716 .ad
1717 .RS 21n
1718 backward-delete-char
1719 .RE

1721 .sp
1722 .ne 2
1723 .na
1724 \fB\fBM-d\fR\fR
1725 .ad
1726 .RS 21n
1727 forward-delete-word
1728 .RE

1730 .sp
1731 .ne 2
1732 .na
1733 \fB\fBM-^H\fR\fR
1734 .ad
1735 .RS 21n
1736 backward-delete-word
1737 .RE

1739 .sp
1740 .ne 2
1741 .na
1742 \fB\fBM-^?\fR\fR
1743 .ad
1744 .RS 21n
1745 backward-delete-word
1746 .RE

1748 .sp
1749 .ne 2
1750 .na
1751 \fB\fBM-u\fR\fR
1752 .ad
1753 .RS 21n
1754 upcase-word
1755 .RE

1757 .sp
1758 .ne 2
1759 .na
1760 \fB\fBM-l\fR\fR
1761 .ad
1762 .RS 21n
1763 downcase-word
1764 .RE

1766 .sp
1767 .ne 2
1768 .na
1769 \fB\fBM-c\fR\fR
1770 .ad
1771 .RS 21n
1772 capitalize-word
```



```
1773 .RE

1775 .sp
1776 .ne 2
1777 .na
1778 \fB\fB^R\fR\fR
1779 .ad
1780 .RS 21n
1781 redisplay
1782 .RE

1784 .sp
1785 .ne 2
1786 .na
1787 \fB\fB^L\fR\fR
1788 .ad
1789 .RS 21n
1790 clear-screen
1791 .RE

1793 .sp
1794 .ne 2
1795 .na
1796 \fB\fB^T\fR\fR
1797 .ad
1798 .RS 21n
1799 transpose-chars
1800 .RE

1802 .sp
1803 .ne 2
1804 .na
1805 \fB\fB^@\fR\fR
1806 .ad
1807 .RS 21n
1808 set-mark
1809 .RE

1811 .sp
1812 .ne 2
1813 .na
1814 \fB\fB^X^X\fR\fR
1815 .ad
1816 .RS 21n
1817 exchange-point-and-mark
1818 .RE

1820 .sp
1821 .ne 2
1822 .na
1823 \fB\fB^W\fR\fR
1824 .ad
1825 .RS 21n
1826 kill-region
1827 .RE

1829 .sp
1830 .ne 2
1831 .na
1832 \fB\fBM-w\fR\fR
1833 .ad
1834 .RS 21n
1835 copy-region-as-kill
1836 .RE

1838 .sp
```

```
.ne 2
.na
\fB\fB^Y\fR\fR
.ad
.RS 21n
yank
.RE

.sp
.ne 2
.na
\fB\fB^P\fR\fR
.ad
.RS 21n
up-history
.RE

.sp
.ne 2
.na
\fB\fB^N\fR\fR
.ad
.RS 21n
down-history
.RE

.sp
.ne 2
.na
\fB\fBM-p\fR\fR
.ad
.RS 21n
history-search-backward
.RE

.sp
.ne 2
.na
\fB\fBM-n\fR\fR
.ad
.RS 21n
history-search-forward
.RE

.sp
.ne 2
.na
\fB\fB^I\fR\fR
.ad
.RS 21n
complete-word
.RE

.sp
.ne 2
.na
\fB\fB^X*\fR\fR
.ad
.RS 21n
expand-filename
.RE

.sp
.ne 2
.na
\fB\fB^X^F\fR\fR
```



```
.ad
.RS 21n
read-from-file
.RE

.sp
.ne 2
.na
\fB\fB^X^R\fR\fR
.ad
.RS 21n
read-init-files
.RE

.sp
.ne 2
.na
\fB\fB^Xg\fR\fR
.ad
.RS 21n
list-glob
.RE

.sp
.ne 2
.na
\fB\fB^Xh\fR\fR
.ad
.RS 21n
list-history
.RE

.sp
.ne 2
.na
\fB\fBM-<\fR\fR
.ad
.RS 21n
beginning-of-history
.RE

.sp
.ne 2
.na
\fB\fBM->\fR\fR
.ad
.RS 21n
end-of-history
.RE

.sp
.ne 2
.na
\fB\fB\en\fR\fR
.ad
.RS 21n
newline
.RE

.sp
.ne 2
.na
\fB\fB\er\fR\fR
.ad
.RS 21n
newline
```

```
1971 .RE

1973 .sp
1974 .ne 2
1975 .na
1976 \fB\fBM-o\fR\fR
1977 .ad
1978 .RS 21n
1979 repeat-history
1980 .RE

1982 .sp
1983 .ne 2
1984 .na
1985 \fB\fBM-^V\fR\fR
1986 .ad
1987 .RS 21n
1988 \fBvi\fR-mode
1989 .RE

1991 .sp
1992 .ne 2
1993 .na
1994 \fB\fBM-0, M-1, ... M-9\fR\fR
1995 .ad
1996 .RS 21n
1997 digit-argument (see below)
1998 .RE

2000 .sp
2001 .LP
2002 Note that \fB^I\fR is what the TAB key generates, and that \fB^@\fR can be
2003 generated not only by pressing the CONTROL key and the @ key simultaneously,
2004 but also by pressing the CONTROL key and the space bar at the same time.
2005 .SS "Default Key Bindings in \fBvi\fR Mode"
2006 .LP
2007 The following default key bindings are designed to mimic the \fBvi\fR style of
2008 editing as closely as possible. This means that very few editing functions are
2009 provided in the initial character input mode, editing functions instead being
2010 provided by the \fBvi\fR command mode. The \fBvi\fR command mode is entered
2011 whenever the ESCAPE character is pressed, or whenever a key sequence that
2012 starts with a meta character is entered. In addition to mimicing \fBvi\fR,
2013 \fBlibtecla\fR provides bindings for tab completion, wild-card expansion of
2014 file names, and historical line recall.
2015 .sp
2016 .LP
2017 To learn how to tell the tecla library to use \fBvi\fR mode instead of the
2018 default \fBemacs\fR editing mode, see the earlier section entitled The Tecla
2019 Configuration File.
2020 .sp
2021 .LP
2022 Under UNIX the terminal driver sets a number of special keys for certain
2023 functions. The tecla library attempts to use the same key bindings to maintain
2024 consistency, binding them both in input mode and in command mode. The key
2025 sequences shown for the following 6 bindings are thus just examples of what
2026 they will probably be set to. If you have used the \fBstty\fR command to change
2027 these keys, then the default bindings should match.
2028 .sp
2029 .ne 2
2030 .na
2031 \fB\fB^C\fR\fR
2032 .ad
2033 .RS 8n
2034 user-interrupt
2035 .RE
```



```
2037 .sp
2038 .ne 2
2039 .na
2040 \fB^\e\fR
2041 .ad
2042 .RS 8n
2043 abort
2044 .RE

2046 .sp
2047 .ne 2
2048 .na
2049 \fB\fB^Z\fR\fR
2050 .ad
2051 .RS 8n
2052 suspend
2053 .RE

2055 .sp
2056 .ne 2
2057 .na
2058 \fB\fB^Q\fR\fR
2059 .ad
2060 .RS 8n
2061 start-output
2062 .RE

2064 .sp
2065 .ne 2
2066 .na
2067 \fB\fB^S\fR\fR
2068 .ad
2069 .RS 8n
2070 stop-output
2071 .RE

2073 .sp
2074 .ne 2
2075 .na
2076 \fB\fB^V\fR\fR
2077 .ad
2078 .RS 8n
2079 literal-next
2080 .RE

2082 .sp
2083 .ne 2
2084 .na
2085 \fB\fBM-^C\fR\fR
2086 .ad
2087 .RS 8n
2088 user-interrupt
2089 .RE

2091 .sp
2092 .ne 2
2093 .na
2094 \fBM-^\e\fR
2095 .ad
2096 .RS 8n
2097 abort
2098 .RE

2100 .sp
2101 .ne 2
2102 .na
```

```
2103 \fB\fBM-^Z\fR\fR
2104 .ad
2105 .RS 8n
2106 suspend
2107 .RE

2109 .sp
2110 .ne 2
2111 .na
2112 \fB\fBM-^Q\fR\fR
2113 .ad
2114 .RS 8n
2115 start-output
2116 .RE

2118 .sp
2119 .ne 2
2120 .na
2121 \fB\fBM-^S\fR\fR
2122 .ad
2123 .RS 8n
2124 stop-output
2125 .RE

2127 .sp
2128 .LP
2129 Note that above, most of the bindings are defined twice, once as a raw control
2130 code like \fB^C\fR and then a second time as a META character like \fBM-^C\fR.
2131 The former is the binding for \fBvi\fR input mode, whereas the latter is the
2132 binding for \fBvi\fR command mode. Once in command mode all key sequences that
2133 the user types that they don't explicitly start with an ESCAPE or a META key,
2134 have their first key secretly converted to a META character before the key
2135 sequence is looked up in the key binding table. Thus, once in command mode,
2136 when you type the letter i, for example, the tecla library actually looks up
2137 the binding for \fBM-i\fR.
2138 .sp
2139 .LP
2140 The cursor keys are referred to by name, as follows. This is necessary because
2141 different types of terminals generate different key sequences when their cursor
2142 keys are pressed.
2143 .sp
2144 .ne 2
2145 .na
2146 \fB\fBright\fR\fR
2147 .ad
2148 .RS 9n
2149 cursor-right
2150 .RE

2152 .sp
2153 .ne 2
2154 .na
2155 \fB\fBleft\fR\fR
2156 .ad
2157 .RS 9n
2158 cursor-left
2159 .RE

2161 .sp
2162 .ne 2
2163 .na
2164 \fB\fBup\fR\fR
2165 .ad
2166 .RS 9n
2167 up-history
2168 .RE
```



```
2170 .sp
2171 .ne 2
2172 .na
2173 \fB\fBdown\fR\fR
2174 .ad
2175 .RS 9n
2176 down-history
2177 .RE

2179 .sp
2180 .LP
2181 The cursor keys normally generate a key sequence that start with an ESCAPE
2182 character, so beware that using the arrow keys will put you into command mode
2183 (if you aren't already in command mode).
2184 .sp
2185 .LP
2186 The following are the terminal-independent key bindings for \fBvi\fR input
2187 mode.
2188 .sp
2189 .ne 2
2190 .na
2191 \fB\fB^D\fR\fR
2192 .ad
2193 .RS 8n
2194 list-or-eof
2195 .RE

2197 .sp
2198 .ne 2
2199 .na
2200 \fB\fB^G\fR\fR
2201 .ad
2202 .RS 8n
2203 list-glob
2204 .RE

2206 .sp
2207 .ne 2
2208 .na
2209 \fB\fB^H\fR\fR
2210 .ad
2211 .RS 8n
2212 backward-delete-char
2213 .RE

2215 .sp
2216 .ne 2
2217 .na
2218 \fB\fB^I\fR\fR
2219 .ad
2220 .RS 8n
2221 complete-word
2222 .RE

2224 .sp
2225 .ne 2
2226 .na
2227 \fB\fB\er\fR\fR
2228 .ad
2229 .RS 8n
2230 newline
2231 .RE

2233 .sp
2234 .ne 2
```

```
.na
\fB\fB\en\fR\fR
.ad
.RS 8n
newline
.RE

.sp
.ne 2
.na
\fB\fB^L\fR\fR
.ad
.RS 8n
clear-screen
.RE

.sp
.ne 2
.na
\fB\fB^N\fR\fR
.ad
.RS 8n
down-history
.RE

.sp
.ne 2
.na
\fB\fB^P\fR\fR
.ad
.RS 8n
up-history
.RE

.sp
.ne 2
.na
\fB\fB^R\fR\fR
.ad
.RS 8n
redisplay
.RE

.sp
.ne 2
.na
\fB\fB^U\fR\fR
.ad
.RS 8n
backward-kill-line
.RE

.sp
.ne 2
.na
\fB\fB^W\fR\fR
.ad
.RS 8n
backward-delete-word
.RE

.sp
.ne 2
.na
\fB\fB^X*\fR\fR
.ad
```



```
.RS 8n
expand-filename
.RE

.sp
.ne 2
.na
\fB\fB^X^F\fR\fR
.ad
.RS 8n
read-from-file
.RE

.sp
.ne 2
.na
\fB\fB^X^R\fR\fR
.ad
.RS 8n
read-init-files
.RE

.sp
.ne 2
.na
\fB\fB^?\fR\fR
.ad
.RS 8n
backward-delete-char
.RE

.sp
.LP
The following are the key bindings that are defined in \fBvi\fR command mode,
this being specified by them all starting with a META character. As mentioned
above, once in command mode the initial meta character is optional. For
example, you might enter command mode by typing ESCAPE, and then press 'H'
twice to move the cursor two positions to the left. Both 'H' characters get
quietly converted to \fBM-h\fR before being compared to the key binding table,
the first one because ESCAPE followed by a character is always converted to the
equivalent META character, and the second because command mode was already
active.
.sp
.ne 2
.na
\fBM-<space>\fR
.ad
.RS 21n
cursor-right (META-space)
.RE

.sp
.ne 2
.na
\fB\fBM-$\fR\fR
.ad
.RS 21n
end-of-line
.RE

.sp
.ne 2
.na
\fB\fBM-*\fR\fR
.ad
.RS 21n
```

```
2367 expand-filename
2368 .RE

2370 .sp
2371 .ne 2
2372 .na
2373 \fB\fBM-+\fR\fR
2374 .ad
2375 .RS 21n
2376 down-history
2377 .RE

2379 .sp
2380 .ne 2
2381 .na
2382 \fB\fBM--\fR\fR
2383 .ad
2384 .RS 21n
2385 up-history
2386 .RE

2388 .sp
2389 .ne 2
2390 .na
2391 \fB\fBM-<\fR\fR
2392 .ad
2393 .RS 21n
2394 beginning-of-history
2395 .RE

2397 .sp
2398 .ne 2
2399 .na
2400 \fB\fBM->\fR\fR
2401 .ad
2402 .RS 21n
2403 end-of-history
2404 .RE

2406 .sp
2407 .ne 2
2408 .na
2409 \fB\fBM-^\fR\fR
2410 .ad
2411 .RS 21n
2412 beginning-of-line
2413 .RE

2415 .sp
2416 .ne 2
2417 .na
2418 \fB\fBM-\fR\fR
2419 .ad
2420 .RS 21n
2421 repeat-find-char
2422 .RE

2424 .sp
2425 .ne 2
2426 .na
2427 \fB\fBM-,\fR\fR
2428 .ad
2429 .RS 21n
2430 invert-refind-char
2431 .RE
```



```
2433 .sp
2434 .ne 2
2435 .na
2436 \fB\fBM-|\fR\fR
2437 .ad
2438 .RS 21n
2439 goto-column
2440 .RE

2442 .sp
2443 .ne 2
2444 .na
2445 \fB\fBM-~\fR\fR
2446 .ad
2447 .RS 21n
2448 change-case
2449 .RE

2451 .sp
2452 .ne 2
2453 .na
2454 \fB\fBM-.\fR\fR
2455 .ad
2456 .RS 21n
2457 vi-repeat-change
2458 .RE

2460 .sp
2461 .ne 2
2462 .na
2463 \fB\fBM-%\fR\fR
2464 .ad
2465 .RS 21n
2466 find-parenthesis
2467 .RE

2469 .sp
2470 .ne 2
2471 .na
2472 \fB\fBM-a\fR\fR
2473 .ad
2474 .RS 21n
2475 vi-append
2476 .RE

2478 .sp
2479 .ne 2
2480 .na
2481 \fB\fBM-A\fR\fR
2482 .ad
2483 .RS 21n
2484 vi-append-at-eol
2485 .RE

2487 .sp
2488 .ne 2
2489 .na
2490 \fB\fBM-b\fR\fR
2491 .ad
2492 .RS 21n
2493 backward-word
2494 .RE

2496 .sp
2497 .ne 2
2498 .na
```

```
2499 \fB\fBM-B\fR\fR
2500 .ad
2501 .RS 21n
2502 backward-word
2503 .RE

2505 .sp
2506 .ne 2
2507 .na
2508 \fB\fBM-C\fR\fR
2509 .ad
2510 .RS 21n
2511 vi-change-rest-of-line
2512 .RE

2514 .sp
2515 .ne 2
2516 .na
2517 \fB\fBM-cb\fR\fR
2518 .ad
2519 .RS 21n
2520 vi-backward-change-word
2521 .RE

2523 .sp
2524 .ne 2
2525 .na
2526 \fB\fBM-cB\fR\fR
2527 .ad
2528 .RS 21n
2529 vi-backward-change-word
2530 .RE

2532 .sp
2533 .ne 2
2534 .na
2535 \fB\fBM-cc\fR\fR
2536 .ad
2537 .RS 21n
2538 vi-change-line
2539 .RE

2541 .sp
2542 .ne 2
2543 .na
2544 \fB\fBM-ce\fR\fR
2545 .ad
2546 .RS 21n
2547 vi-forward-change-word
2548 .RE

2550 .sp
2551 .ne 2
2552 .na
2553 \fB\fBM-cE\fR\fR
2554 .ad
2555 .RS 21n
2556 vi-forward-change-word
2557 .RE

2559 .sp
2560 .ne 2
2561 .na
2562 \fB\fBM-cw\fR\fR
2563 .ad
2564 .RS 21n
```



```
2565 vi-forward-change-word
2566 .RE

2568 .sp
2569 .ne 2
2570 .na
2571 \fB\fBM-cW\fR\fR
2572 .ad
2573 .RS 21n
2574 vi-forward-change-word
2575 .RE

2577 .sp
2578 .ne 2
2579 .na
2580 \fB\fBM-cF\fR\fR
2581 .ad
2582 .RS 21n
2583 vi-backward-change-find
2584 .RE

2586 .sp
2587 .ne 2
2588 .na
2589 \fB\fBM-cf\fR\fR
2590 .ad
2591 .RS 21n
2592 vi-forward-change-find
2593 .RE

2595 .sp
2596 .ne 2
2597 .na
2598 \fB\fBM-cT\fR\fR
2599 .ad
2600 .RS 21n
2601 vi-backward-change-to
2602 .RE

2604 .sp
2605 .ne 2
2606 .na
2607 \fB\fBM-ct\fR\fR
2608 .ad
2609 .RS 21n
2610 vi-forward-change-to
2611 .RE

2613 .sp
2614 .ne 2
2615 .na
2616 \fB\fBM-c;\fR\fR
2617 .ad
2618 .RS 21n
2619 vi-change-refind
2620 .RE

2622 .sp
2623 .ne 2
2624 .na
2625 \fB\fBM-c,\fR\fR
2626 .ad
2627 .RS 21n
2628 vi-change-invert-refind
2629 .RE
```

```
2631 .sp
2632 .ne 2
2633 .na
2634 \fB\fBM-ch\fR\fR
2635 .ad
2636 .RS 21n
2637 vi-backward-change-char
2638 .RE

2640 .sp
2641 .ne 2
2642 .na
2643 \fB\fBM-c^H\fR\fR
2644 .ad
2645 .RS 21n
2646 vi-backward-change-char
2647 .RE

2649 .sp
2650 .ne 2
2651 .na
2652 \fB\fBM-c^?\fR\fR
2653 .ad
2654 .RS 21n
2655 vi-backward-change-char
2656 .RE

2658 .sp
2659 .ne 2
2660 .na
2661 \fB\fBM-cl\fR\fR
2662 .ad
2663 .RS 21n
2664 vi-forward-change-char
2665 .RE

2667 .sp
2668 .ne 2
2669 .na
2670 \fBM-c<space>\fR
2671 .ad
2672 .RS 21n
2673 vi-forward-change-char (META-c-space)
2674 .RE

2676 .sp
2677 .ne 2
2678 .na
2679 \fB\fBM-c^\fR\fR
2680 .ad
2681 .RS 21n
2682 vi-change-to-bol
2683 .RE

2685 .sp
2686 .ne 2
2687 .na
2688 \fB\fBM-c0\fR\fR
2689 .ad
2690 .RS 21n
2691 vi-change-to-bol
2692 .RE

2694 .sp
2695 .ne 2
2696 .na
```



```
2697 \fB\fBM-c$\fR\fR
2698 .ad
2699 .RS 21n
2700 vi-change-rest-of-line
2701 .RE

2703 .sp
2704 .ne 2
2705 .na
2706 \fB\fBM-c|\fR\fR
2707 .ad
2708 .RS 21n
2709 vi-change-to-column
2710 .RE

2712 .sp
2713 .ne 2
2714 .na
2715 \fB\fBM-c%\fR\fR
2716 .ad
2717 .RS 21n
2718 vi-change-to-parenthesis
2719 .RE

2721 .sp
2722 .ne 2
2723 .na
2724 \fB\fBM-dh\fR\fR
2725 .ad
2726 .RS 21n
2727 backward-delete-char
2728 .RE

2730 .sp
2731 .ne 2
2732 .na
2733 \fB\fBM-d^H\fR\fR
2734 .ad
2735 .RS 21n
2736 backward-delete-char
2737 .RE

2739 .sp
2740 .ne 2
2741 .na
2742 \fB\fBM-d^?\fR\fR
2743 .ad
2744 .RS 21n
2745 backward-delete-char
2746 .RE

2748 .sp
2749 .ne 2
2750 .na
2751 \fB\fBM-dl\fR\fR
2752 .ad
2753 .RS 21n
2754 forward-delete-char
2755 .RE

2757 .sp
2758 .ne 2
2759 .na
2760 \fBM-d<space>\fR
2761 .ad
2762 .RS 21n
```

```
2763 forward-delete-char (META-d-space)
2764 .RE

2766 .sp
2767 .ne 2
2768 .na
2769 \fB\fBM-dd\fR\fR
2770 .ad
2771 .RS 21n
2772 delete-line
2773 .RE

2775 .sp
2776 .ne 2
2777 .na
2778 \fB\fBM-db\fR\fR
2779 .ad
2780 .RS 21n
2781 backward-delete-word
2782 .RE

2784 .sp
2785 .ne 2
2786 .na
2787 \fB\fBM-dB\fR\fR
2788 .ad
2789 .RS 21n
2790 backward-delete-word
2791 .RE

2793 .sp
2794 .ne 2
2795 .na
2796 \fB\fBM-de\fR\fR
2797 .ad
2798 .RS 21n
2799 forward-delete-word
2800 .RE

2802 .sp
2803 .ne 2
2804 .na
2805 \fB\fBM-dE\fR\fR
2806 .ad
2807 .RS 21n
2808 forward-delete-word
2809 .RE

2811 .sp
2812 .ne 2
2813 .na
2814 \fB\fBM-dw\fR\fR
2815 .ad
2816 .RS 21n
2817 forward-delete-word
2818 .RE

2820 .sp
2821 .ne 2
2822 .na
2823 \fB\fBM-dW\fR\fR
2824 .ad
2825 .RS 21n
2826 forward-delete-word
2827 .RE
```



```
2829 .sp
2830 .ne 2
2831 .na
2832 \fB\fBM-dF\fR\fR
2833 .ad
2834 .RS 21n
2835 backward-delete-find
2836 .RE

2838 .sp
2839 .ne 2
2840 .na
2841 \fB\fBM-df\fR\fR
2842 .ad
2843 .RS 21n
2844 forward-delete-find
2845 .RE

2847 .sp
2848 .ne 2
2849 .na
2850 \fB\fBM-dT\fR\fR
2851 .ad
2852 .RS 21n
2853 backward-delete-to
2854 .RE

2856 .sp
2857 .ne 2
2858 .na
2859 \fB\fBM-dt\fR\fR
2860 .ad
2861 .RS 21n
2862 forward-delete-to
2863 .RE

2865 .sp
2866 .ne 2
2867 .na
2868 \fB\fBM-d;\fR\fR
2869 .ad
2870 .RS 21n
2871 delete-refind
2872 .RE

2874 .sp
2875 .ne 2
2876 .na
2877 \fB\fBM-d,\fR\fR
2878 .ad
2879 .RS 21n
2880 delete-invert-refind
2881 .RE

2883 .sp
2884 .ne 2
2885 .na
2886 \fB\fBM-d^\fR\fR
2887 .ad
2888 .RS 21n
2889 backward-kill-line
2890 .RE

2892 .sp
2893 .ne 2
2894 .na
```

```
2895 \fB\fBM-d0\fR\fR
2896 .ad
2897 .RS 21n
2898 backward-kill-line
2899 .RE

2901 .sp
2902 .ne 2
2903 .na
2904 \fB\fBM-d$\fR\fR
2905 .ad
2906 .RS 21n
2907 kill-line
2908 .RE

2910 .sp
2911 .ne 2
2912 .na
2913 \fB\fBM-D\fR\fR
2914 .ad
2915 .RS 21n
2916 kill-line
2917 .RE

2919 .sp
2920 .ne 2
2921 .na
2922 \fB\fBM-d|\fR\fR
2923 .ad
2924 .RS 21n
2925 delete-to-column
2926 .RE

2928 .sp
2929 .ne 2
2930 .na
2931 \fB\fBM-d%\fR\fR
2932 .ad
2933 .RS 21n
2934 delete-to-parenthesis
2935 .RE

2937 .sp
2938 .ne 2
2939 .na
2940 \fB\fBM-e\fR\fR
2941 .ad
2942 .RS 21n
2943 forward-word
2944 .RE

2946 .sp
2947 .ne 2
2948 .na
2949 \fB\fBM-E\fR\fR
2950 .ad
2951 .RS 21n
2952 forward-word
2953 .RE

2955 .sp
2956 .ne 2
2957 .na
2958 \fB\fBM-f\fR\fR
2959 .ad
2960 .RS 21n
```



```
2961 forward-find-char
2962 .RE

2964 .sp
2965 .ne 2
2966 .na
2967 \fB\fBM-F\fR\fR
2968 .ad
2969 .RS 21n
2970 backward-find-char
2971 .RE

2973 .sp
2974 .ne 2
2975 .na
2976 \fB\fBM--\fR\fR
2977 .ad
2978 .RS 21n
2979 up-history
2980 .RE

2982 .sp
2983 .ne 2
2984 .na
2985 \fB\fBM-h\fR\fR
2986 .ad
2987 .RS 21n
2988 cursor-left
2989 .RE

2991 .sp
2992 .ne 2
2993 .na
2994 \fB\fBM-H\fR\fR
2995 .ad
2996 .RS 21n
2997 beginning-of-history
2998 .RE

3000 .sp
3001 .ne 2
3002 .na
3003 \fB\fBM-i\fR\fR
3004 .ad
3005 .RS 21n
3006 vi-insert
3007 .RE

3009 .sp
3010 .ne 2
3011 .na
3012 \fB\fBM-I\fR\fR
3013 .ad
3014 .RS 21n
3015 vi-insert-at-bol
3016 .RE

3018 .sp
3019 .ne 2
3020 .na
3021 \fB\fBM-j\fR\fR
3022 .ad
3023 .RS 21n
3024 down-history
3025 .RE
```

```
3027 .sp
3028 .ne 2
3029 .na
3030 \fB\fBM-J\fR\fR
3031 .ad
3032 .RS 21n
3033 history-search-forward
3034 .RE

3036 .sp
3037 .ne 2
3038 .na
3039 \fB\fBM-k\fR\fR
3040 .ad
3041 .RS 21n
3042 up-history
3043 .RE

3045 .sp
3046 .ne 2
3047 .na
3048 \fB\fBM-K\fR\fR
3049 .ad
3050 .RS 21n
3051 history-search-backward
3052 .RE

3054 .sp
3055 .ne 2
3056 .na
3057 \fB\fBM-l\fR\fR
3058 .ad
3059 .RS 21n
3060 cursor-right
3061 .RE

3063 .sp
3064 .ne 2
3065 .na
3066 \fB\fBM-L\fR\fR
3067 .ad
3068 .RS 21n
3069 end-of-history
3070 .RE

3072 .sp
3073 .ne 2
3074 .na
3075 \fB\fBM-n\fR\fR
3076 .ad
3077 .RS 21n
3078 history-re-search-forward
3079 .RE

3081 .sp
3082 .ne 2
3083 .na
3084 \fB\fBM-N\fR\fR
3085 .ad
3086 .RS 21n
3087 history-re-search-backward
3088 .RE

3090 .sp
3091 .ne 2
3092 .na
```



```
3093 \fB\fBM-p\fR\fR
3094 .ad
3095 .RS 21n
3096 append-yank
3097 .RE

3099 .sp
3100 .ne 2
3101 .na
3102 \fB\fBM-P\fR\fR
3103 .ad
3104 .RS 21n
3105 yank
3106 .RE

3108 .sp
3109 .ne 2
3110 .na
3111 \fB\fBM-r\fR\fR
3112 .ad
3113 .RS 21n
3114 vi-replace-char
3115 .RE

3117 .sp
3118 .ne 2
3119 .na
3120 \fB\fBM-R\fR\fR
3121 .ad
3122 .RS 21n
3123 vi-overwrite
3124 .RE

3126 .sp
3127 .ne 2
3128 .na
3129 \fB\fBM-s\fR\fR
3130 .ad
3131 .RS 21n
3132 vi-forward-change-char
3133 .RE

3135 .sp
3136 .ne 2
3137 .na
3138 \fB\fBM-S\fR\fR
3139 .ad
3140 .RS 21n
3141 vi-change-line
3142 .RE

3144 .sp
3145 .ne 2
3146 .na
3147 \fB\fBM-t\fR\fR
3148 .ad
3149 .RS 21n
3150 forward-to-char
3151 .RE

3153 .sp
3154 .ne 2
3155 .na
3156 \fB\fBM-T\fR\fR
3157 .ad
3158 .RS 21n
```

```
3159 backward-to-char
3160 .RE

3162 .sp
3163 .ne 2
3164 .na
3165 \fB\fBM-u\fR\fR
3166 .ad
3167 .RS 21n
3168 vi-undo
3169 .RE

3171 .sp
3172 .ne 2
3173 .na
3174 \fB\fBM-w\fR\fR
3175 .ad
3176 .RS 21n
3177 forward-to-word
3178 .RE

3180 .sp
3181 .ne 2
3182 .na
3183 \fB\fBM-W\fR\fR
3184 .ad
3185 .RS 21n
3186 forward-to-word
3187 .RE

3189 .sp
3190 .ne 2
3191 .na
3192 \fB\fBM-x\fR\fR
3193 .ad
3194 .RS 21n
3195 forward-delete-char
3196 .RE

3198 .sp
3199 .ne 2
3200 .na
3201 \fB\fBM-X\fR\fR
3202 .ad
3203 .RS 21n
3204 backward-delete-char
3205 .RE

3207 .sp
3208 .ne 2
3209 .na
3210 \fB\fBM-yh\fR\fR
3211 .ad
3212 .RS 21n
3213 backward-copy-char
3214 .RE

3216 .sp
3217 .ne 2
3218 .na
3219 \fB\fBM-y^H\fR\fR
3220 .ad
3221 .RS 21n
3222 backward-copy-char
3223 .RE
```



```
3225 .sp
3226 .ne 2
3227 .na
3228 \fB\fBM-y^?\fR\fR
3229 .ad
3230 .RS 21n
3231 backward-copy-char
3232 .RE

3234 .sp
3235 .ne 2
3236 .na
3237 \fB\fBM-yl\fR\fR
3238 .ad
3239 .RS 21n
3240 forward-copy-char
3241 .RE

3243 .sp
3244 .ne 2
3245 .na
3246 \fBM-y<space>\fR
3247 .ad
3248 .RS 21n
3249 forward-copy-char (META-y-space)
3250 .RE

3252 .sp
3253 .ne 2
3254 .na
3255 \fB\fBM-ye\fR\fR
3256 .ad
3257 .RS 21n
3258 forward-copy-word
3259 .RE

3261 .sp
3262 .ne 2
3263 .na
3264 \fB\fBM-yE\fR\fR
3265 .ad
3266 .RS 21n
3267 forward-copy-word
3268 .RE

3270 .sp
3271 .ne 2
3272 .na
3273 \fB\fBM-yw\fR\fR
3274 .ad
3275 .RS 21n
3276 forward-copy-word
3277 .RE

3279 .sp
3280 .ne 2
3281 .na
3282 \fB\fBM-yW\fR\fR
3283 .ad
3284 .RS 21n
3285 forward-copy-word
3286 .RE

3288 .sp
3289 .ne 2
3290 .na
```

```
\fB\fBM-yb\fR\fR
.ad
.RS 21n
backward-copy-word
.RE

.sp
.ne 2
.na
\fB\fBM-yB\fR\fR
.ad
.RS 21n
backward-copy-word
.RE

.sp
.ne 2
.na
\fB\fBM-yf\fR\fR
.ad
.RS 21n
forward-copy-find
.RE

.sp
.ne 2
.na
\fB\fBM-yF\fR\fR
.ad
.RS 21n
backward-copy-find
.RE

.sp
.ne 2
.na
\fB\fBM-yt\fR\fR
.ad
.RS 21n
forward-copy-to
.RE

.sp
.ne 2
.na
\fB\fBM-yT\fR\fR
.ad
.RS 21n
backward-copy-to
.RE

.sp
.ne 2
.na
\fB\fBM-y;\fR\fR
.ad
.RS 21n
copy-refind
.RE

.sp
.ne 2
.na
\fB\fBM-y,\fR\fR
.ad
.RS 21n
```



```
copy-invert-refind
.RE

.sp
.ne 2
.na
\fB\fBM-y^\fR\fR
.ad
.RS 21n
copy-to-bol
.RE

.sp
.ne 2
.na
\fB\fBM-y0\fR\fR
.ad
.RS 21n
copy-to-bol
.RE

.sp
.ne 2
.na
\fB\fBM-y$\fR\fR
.ad
.RS 21n
copy-rest-of-line
.RE

.sp
.ne 2
.na
\fB\fBM-yy\fR\fR
.ad
.RS 21n
copy-line
.RE

.sp
.ne 2
.na
\fB\fBM-Y\fR\fR
.ad
.RS 21n
copy-line
.RE

.sp
.ne 2
.na
\fB\fBM-y|\fR\fR
.ad
.RS 21n
copy-to-column
.RE

.sp
.ne 2
.na
\fB\fBM-y%\fR\fR
.ad
.RS 21n
copy-to-parenthesis
.RE
```

```
3423 .sp
3424 .ne 2
3425 .na
3426 \fB\fBM-^E\fR\fR
3427 .ad
3428 .RS 21n
3429 emacs-mode
3430 .RE

3432 .sp
3433 .ne 2
3434 .na
3435 \fB\fBM-^H\fR\fR
3436 .ad
3437 .RS 21n
3438 cursor-left
3439 .RE

3441 .sp
3442 .ne 2
3443 .na
3444 \fB\fBM-^?\fR\fR
3445 .ad
3446 .RS 21n
3447 cursor-left
3448 .RE

3450 .sp
3451 .ne 2
3452 .na
3453 \fB\fBM-^L\fR\fR
3454 .ad
3455 .RS 21n
3456 clear-screen
3457 .RE

3459 .sp
3460 .ne 2
3461 .na
3462 \fB\fBM-^N\fR\fR
3463 .ad
3464 .RS 21n
3465 down-history
3466 .RE

3468 .sp
3469 .ne 2
3470 .na
3471 \fB\fBM-^P\fR\fR
3472 .ad
3473 .RS 21n
3474 up-history
3475 .RE

3477 .sp
3478 .ne 2
3479 .na
3480 \fB\fBM-^R\fR\fR
3481 .ad
3482 .RS 21n
3483 redisplay
3484 .RE

3486 .sp
3487 .ne 2
3488 .na
```



```
3489 \fB\fBM-^D\fR\fR
3490 .ad
3491 .RS 21n
3492 list-or-eof
3493 .RE

3495 .sp
3496 .ne 2
3497 .na
3498 \fB\fBM-^I\fR\fR
3499 .ad
3500 .RS 21n
3501 complete-word
3502 .RE

3504 .sp
3505 .ne 2
3506 .na
3507 \fBM-\er\fR
3508 .ad
3509 .RS 21n
3510 newline
3511 .RE

3513 .sp
3514 .ne 2
3515 .na
3516 \fB\fBM-\en\fR\fR
3517 .ad
3518 .RS 21n
3519 newline
3520 .RE

3522 .sp
3523 .ne 2
3524 .na
3525 \fB\fBM-^X^R\fR\fR
3526 .ad
3527 .RS 21n
3528 read-init-files
3529 .RE

3531 .sp
3532 .ne 2
3533 .na
3534 \fB\fBM-^Xh\fR\fR
3535 .ad
3536 .RS 21n
3537 list-history
3538 .RE

3540 .sp
3541 .ne 2
3542 .na
3543 \fB\fBM-0, M-1, ... M-9\fR\fR
3544 .ad
3545 .RS 21n
3546 digit-argument (see below)
3547 .RE

3549 .sp
3550 .LP
3551 Note that \fB^I\fR is what the TAB key generates.
3552 .SS "Entering Repeat Counts"
3553 .LP
3554 Many of the key binding functions described previously, take an optional count,
```

typed in before the target key sequence. This is interpreted as a repeat count by most bindings. A notable exception is the goto-column binding, which interprets the count as a column number.

.LP
By default you can specify this count argument by pressing the META key while typing in the numeric count. This relies on the digit-argument action being bound to 'META-0', 'META-1' etc. Once any one of these bindings has been activated, you can optionally take your finger off the META key to type in the rest of the number, since every numeric digit thereafter is treated as part of the number, unless it is preceded by the literal-next binding. As soon as a non-digit, or literal digit key is pressed the repeat count is terminated and either causes the just typed character to be added to the line that many times, or causes the next key binding function to be given that argument.

.LP
For example, in \fBemacs\fR mode, typing:

```
M-12a
```

.LP
causes the letter 'a' to be added to the line 12 times, whereas

```
M-4M-c
```

.LP
Capitalizes the next 4 words.

.LP
In \fBvi\fR command mode the meta modifier is automatically added to all characters typed in, so to enter a count in \fBvi\fR command-mode, just involves typing in the number, just as it does in the \fBvi\fR editor itself. So for example, in vi command mode, typing:

```
4w2x
```

moves the cursor four words to the right, then deletes two characters.

.LP
You can also bind digit-argument to other key sequences. If these end in a numeric digit, that digit gets appended to the current repeat count. If it doesn't end in a numeric digit, a new repeat count is started with a value of zero, and can be completed by typing in the number, after letting go of the key which triggered the digit-argument action.

FILES

\fB/usr/lib/libtecla.so\fR
: The tecla library

\fB/usr/include/libtecla.h\fR
: The tecla header file

\fB~/.teclarc\fR
: The personal tecla customization file

ATTRIBUTES

See \fBattributes\fR(5) for descriptions of the following attributes:

ATTRIBUTE TYPE	ATTRIBUTE VALUE
Interface Stability	Evolving

SEE ALSO

\fBvi\fR(1), \fBcpl_complete_word\fR(3TECLA), \fBef_expand_file\fR(3TECLA), \fBgl_get_line\fR(3TECLA), \fBgl_io_mode\fR(3TECLA), \fBlibtecla\fR(3LIB), \fBpca_lookup_file\fR(3TECLA), \fBattributes\fR(5)

```
**********************************************************
   16276 Sun Sep 16 19:22:58 2018
new/usr/src/man/man5/threads.5
9842 man page typos and spelling
**********************************************************
'\" te
.\" Copyright (c) 2008, Sun Microsystems, Inc.  All Rights Reserved.
.\" Copyright 2016 Joyent, Inc.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH THREADS 5 "Mar 27, 2016"
.SH NAME
threads, pthreads \- POSIX pthreads, c11, and illumos threads concepts
.SH SYNOPSIS
.SS "POSIX"
.LP
.nf
gcc -D_REENTRANT [ \fIflag\fR... ] \fIfile\fR... [ \fIlibrary\fR... ]
.fi

.LP
.nf
#include <pthread.h>
.fi

.SS "C11"
.LP
.nf
gcc -std=c11 -D_REENTRANT [ \fIflag\fR... ] \fIfile\fR... [ \fIlibrary\fR... ]
.fi

.LP
.nf
#include <threads.h>
.fi

.SS "illumos"
.LP
.nf
gcc -D_REENTRANT [ \fIflag\fR... ] \fIfile\fR... [ \fIlibrary\fR... ]
.fi

.LP
.nf
#include <sched.h>
.fi

.LP
.nf
#include <thread.h>
.fi

.SH DESCRIPTION
.LP
A thread is an independent source of execution within a process. Every process
is created with a single thread, which calls the
.B main
function. A process may have multiple threads, all of which are scheduled
independently by the system and may run concurrently. Threads within a process
all use the same address space and as a result can access all data in the
process; however, each thread is created with its own attributes and its own
stack. When a thread is created, it inherits the signal mask of the thread which
created it, but it has no pending signals.
.sp
.LP
```



```
All threads of execution have their own, independent life time, though it is
ultimately bounded by the life time of the process. If the process terminates
for any reason, whether due to a call to \fBexit\fR(3C), the receipt of a fatal
signal, or some other reason, then all threads within the process are
terminated. Threads may themselves exit and status information of them may be
obtained, for more information, see the \fBpthread_detach\fR(3C),
\fBpthread_join\fR(3C), and \fBpthread_exit\fR(3C) functions, and their
equivalents as described in the tables later on in the manual.
.sp
.LP
Most hardware platforms do not have any special synchronization for data objects
which may be accessed concurrently from multiple threads of execution. To avoid
such problems, programs may use atomic operations (see \fBatomic_ops\fR(3C)) and
locking primitives, such as mutexes, readers/writer locks, condition variables,
and semaphores. Note, that depending on the hardware platform, memory
synchronization may be necessary, for more information, see \fBmembar_ops\fR(3C)
.LP
POSIX, C11, and illumos threads each have their own implementation within
\fBlibc\fR(3LIB). All implementations are interoperable, their functionality
similar, and can be used within the same application. Only POSIX threads are
guaranteed to be fully portable to other POSIX-compliant environments. C11
threads are an optional part of ISO C11 and may not exist on every ISO C11
platform. POSIX, C11, and illumos threads require different source and include
files. See \fBSYNOPSIS\fR.
.SS "Similarities"
.LP
Most of the POSIX and illumos threading functions have counterparts with each
other. POSIX function names, with the exception of the semaphore names, have a
"\fBpthread\fR" prefix. Function names for similar POSIX and illumos functions
have similar endings. Typically, similar POSIX and illumos functions have the
same number and use of arguments.
.SS "Differences"
.LP
POSIX pthreads and illumos threads differ in the following ways:
.RS +4
.TP
.ie t \(bu
.el o
POSIX threads are more portable.
.RE
.RS +4
.TP
.ie t \(bu
.el o
POSIX threads establish characteristics  for each thread according to
configurable attribute objects.
.RE
.RS +4
.TP
.ie t \(bu
.el o
POSIX pthreads implement thread cancellation.
.RE
.RS +4
.TP
.ie t \(bu
.el o
POSIX pthreads enforce scheduling algorithms.
.RE
.RS +4
.TP
.ie t \(bu
.el o
POSIX pthreads allow for clean-up handlers for \fBfork\fR(2) calls.
.RE
.RS +4
```

```
.TP
.ie t \(bu
.el o
illumos threads can be suspended and continued.
.RE
.RS +4
.TP
.ie t \(bu
.el o
illumos threads implement daemon threads, for whose demise the process does not
wait.
.RE
.SS "Comparison to C11 Threads"
.LP
C11 threads are not as functional as either POSIX or illumos threads. C11
threads only support intra-process locking and do not have any form of
readers/writer locking or semaphores. In general, POSIX threads will be more
portable than C11 threads, all POSIX-compliant systems support pthreads;
however, not all C environments support C11 Threads.
.sp
.LP
In addition to lacking other common synchronization primitives, the ISO/IEC
standard for C11 threads does not have rich error semantics. In an effort to not
extend the set of error numbers standardized in ISO/IEC C11, none of the
routines set errno and instead multiple distinguishable errors, aside from the
equivalent to ENOMEM and EBUSY, are all squashed into one. As such, users of the
platform are encouraged to use POSIX threads, unless a portability concern
dictates otherwise.

.SH FUNCTION COMPARISON
.LP
The following table compares the POSIX pthreads, C11 threads, and illumos
threads functions.  When a comparable interface is not available either in POSIX
pthreads, C11 threads  or illumos threads, a hyphen (\fB-\fR) appears in the
column.
.SS "Functions Related to Creation"

.TS
l l l
l l l .
\fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
\fBpthread_create()\fR  \fBthr_create()\fR      \fBthrd_create()\fR
\fBpthread_attr_init()\fR       \fB-\fR \fB-\fR
\fBpthread_attr_setdetachstate()\fR     \fB-\fR \fB-\fR
\fBpthread_attr_getdetachstate()\fR     \fB-\fR \fB-\fR
\fBpthread_attr_setinheritsched()\fR    \fB-\fR \fB-\fR
\fBpthread_attr_getinheritsched()\fR    \fB-\fR \fB-\fR
\fBpthread_attr_setschedparam()\fR      \fB-\fR \fB-\fR
\fBpthread_attr_getschedparam()\fR      \fB-\fR \fB-\fR
\fBpthread_attr_setschedpolicy()\fR     \fB-\fR \fB-\fR
\fBpthread_attr_getschedpolicy()\fR     \fB-\fR \fB-\fR
\fBpthread_attr_setscope()\fR   \fB-\fR \fB-\fR
\fBpthread_attr_getscope()\fR   \fB-\fR \fB-\fR
\fBpthread_attr_setstackaddr()\fR       \fB-\fR \fB-\fR
\fBpthread_attr_getstackaddr()\fR       \fB-\fR \fB-\fR
\fBpthread_attr_setstacksize()\fR       \fB-\fR \fB-\fR
\fBpthread_attr_getstacksize()\fR       \fB-\fR \fB-\fR
\fBpthread_attr_getguardsize()\fR       \fB-\fR \fB-\fR
\fBpthread_attr_setguardsize()\fR       \fB-\fR \fB-\fR
\fBpthread_attr_destroy()\fR    \fB-\fR \fB-\fR
\fB-\fR \fBthr_min_stack()\fR   \fB-\fR
.TE

.SS "Functions Related to Exit"

.TS
```



```
l l l
l l l .
\fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
\fBpthread_exit()\fR    \fBthr_exit()\fR        \fBthrd_exit()\fR
\fBpthread_join()\fR    \fBthr_join()\fR        \fBthrd_join()\fR
\fBpthread_detach()\fR  \fB-\fR \fBthrd_detach()\fR
.TE

.SS "Functions Related to Thread Specific Data"

.TS
l l l
l l l .
\fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
\fBpthread_key_create()\fR      \fBthr_keycreate()\fR   \fBtss_create()\fR
\fBpthread_setspecific()\fR     \fBthr_setspecific()\fR \fBtss_set()\fR
\fBpthread_getspecific()\fR     \fBthr_getspecific()\fR \fBtss_get()\fR
\fBpthread_key_delete()\fR      \fB-\fR \fBtss_delete()\fR
.TE

.SS "Functions Related to Signals"

.TS
l l l
l l l .
\fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
\fBpthread_sigmask()\fR \fBthr_sigsetmask()\fR  \fB-\fR
\fBpthread_kill()\fR    \fBthr_kill()\fR        \fB-\fR
.TE

.SS "Functions Related to IDs"

.TS
l l l
l l l .
\fBPOSIX\fR     \fBillumos\fR   \fBc11\fR
\fBpthread_self()\fR    \fBthr_self()\fR        \fBthrd_current()\fR
\fBpthread_equal()\fR   \fB-\fR \fBthrd_equal()\fR
\fB-\fR \fBthr_main()\fR        \fB-\fR
.TE

.SS "Functions Related to Scheduling"

.TS
l l l
l l l .
\fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
\fB-\fR \fBthr_yield()\fR       \fBthrd_yield()\fR
\fB-\fR \fBthr_suspend()\fR     \fB-\fR
\fB-\fR \fBthr_continue()\fR    \fB-\fR
\fBpthread_setconcurrency()\fR  \fBthr_setconcurrency()\fR      \fB-\fR
\fBpthread_getconcurrency()\fR  \fBthr_getconcurrency()\fR      \fB-\fR
\fBpthread_setschedparam()\fR   \fBthr_setprio()\fR     \fB-\fR
\fBpthread_setschedprio()\fR    \fBthr_setprio()\fR     \fB-\fR
\fBpthread_getschedparam()\fR   \fBthr_getprio()\fR     \fB-\fR
.TE

.SS "Functions Related to Cancellation"

.TS
l l l
l l l .
\fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
\fBpthread_cancel()\fR  \fB-\fR \fB-\fR
\fBpthread_setcancelstate()\fR  \fB-\fR \fB-\fR
\fBpthread_setcanceltype()\fR   \fB-\fR \fB-\fR
```

```
\fBpthread_testcancel()\fR      \fB-\fR \fB-\fR
\fBpthread_cleanup_pop()\fR     \fB-\fR \fB-\fR
\fBpthread_cleanup_push()\fR    \fB-\fR \fB-\fR
.TE

.SS "Functions Related to Mutexes"

.TS
l l l
l l l .
\fBPOSIX\fR     \fBillumos\fR   \fBc11\fR
\fBpthread_mutex_init()\fR      \fBmutex_init()\fR      \fBmtx_init()\fR
\fBpthread_mutexattr_init()\fR  \fB-\fR \fB-\fR
\fBpthread_mutexattr_setpshared()\fR    \fB-\fR \fB-\fR
\fBpthread_mutexattr_getpshared()\fR    \fB-\fR \fB-\fR
\fBpthread_mutexattr_setprotocol()\fR   \fB-\fR \fB-\fR
\fBpthread_mutexattr_getprotocol()\fR   \fB-\fR \fB-\fR
\fBpthread_mutexattr_setprioceiling()\fR        \fB-\fR \fB-\fR
\fBpthread_mutexattr_getprioceiling()\fR        \fB-\fR \fB-\fR
\fBpthread_mutexattr_settype()\fR       \fB-\fR \fB-\fR
\fBpthread_mutexattr_gettype()\fR       \fB-\fR \fB-\fR
\fBpthread_mutexattr_setrobust()\fR     \fB-\fR \fB-\fR
\fBpthread_mutexattr_getrobust()\fR     \fB-\fR \fB-\fR
\fBpthread_mutexattr_destroy()\fR       \fB-\fR \fBmtx_destroy()\fR
\fBpthread_mutex_setprioceiling()\fR    \fB-\fR \fB-\fR
\fBpthread_mutex_getprioceiling()\fR    \fB-\fR \fB-\fR
\fBpthread_mutex_lock()\fR      \fBmutex_lock()\fR      \fBmtx_lock()\fR
\fBpthread_mutex_timedlock()\fR \fB-\fR \fBmtx_timedlock()\fR
\fBpthread_mutex_trylock()\fR   \fBmutex_trylock()\fR   \fBmtx_trylock()\fR
\fBpthread_mutex_unlock()\fR    \fBmutex_unlock()\fR    \fBmtx_unlock()\fR
\fIfBpthread_mutex_unlock()\fR    \fIfBmutex_unlock()\fR    \fIfBmtx_unlcok()\fR
\fBpthread_mutex_destroy()\fR   \fBmutex_destroy()\fR   \fBmtx_destroy()\fR
.TE

.SS "Functions Related to Condition Variables"

.TS
l l l
l l l .
\fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
\fBpthread_cond_init()\fR       \fBcond_init()\fR       \fBcnd_init()\fR
\fBpthread_condattr_init()\fR   \fB-\fR \fB-\fR
\fBpthread_condattr_setpshared()\fR     \fB-\fR \fB-\fR
\fBpthread_condattr_getpshared()\fR     \fB-\fR \fB-\fR
\fBpthread_condattr_destroy()\fR        \fB-\fR \fB-\fR
\fBpthread_cond_wait()\fR       \fBcond_wait()\fR       \fBcnd_wait()\fR
\fBpthread_cond_timedwait()\fR  \fBcond_timedwait()\fR  \fBcond_timedwait()\fR
\fBpthread_cond_signal()\fR     \fBcond_signal()\fR     \fBcnd_signal()\fR
\fBpthread_cond_broadcast()\fR  \fBcond_broadcast()\fR  \fBcnd_broadcast()\fR
\fBpthread_cond_destroy()\fR    \fBcond_destroy()\fR    \fBcnd_destroy()\fR
.TE

.SS "Functions Related to Reader/Writer Locking"

.TS
l l l
l l l .
\fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
\fBpthread_rwlock_init()\fR     \fBrwlock_init()\fR     \fB-\fR
\fBpthread_rwlock_rdlock()\fR   \fBrw_rdlock()\fR       \fB-\fR
\fBpthread_rwlock_tryrdlock()\fR        \fBrw_tryrdlock()\fR    \fB-\fR
\fBpthread_rwlock_wrlock()\fR   \fBrw_wrlock()\fR       \fB-\fR
\fBpthread_rwlock_trywrlock()\fR        \fBrw_trywrlock()\fR    \fB-\fR
\fBpthread_rwlock_unlock()\fR   \fBrw_unlock()\fR       \fB-\fR
\fBpthread_rwlock_destroy()\fR  \fBrwlock_destroy()\fR  \fB-\fR
\fBpthread_rwlockattr_init()\fR \fB-\fR \fB-\fR
```



```
\fBpthread_rwlockattr_destroy()\fR      \fB-\fR \fB-\fR
\fBpthread_rwlockattr_getpshared()\fR   \fB-\fR \fB-\fR
\fBpthread_rwlockattr_setpshared()\fR   \fB-\fR \fB-\fR
.TE

.SS "Functions Related to Semaphores"

.TS
l l l
l l l .
\fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
\fBsem_init()\fR        \fBsema_init()\fR       \fB-\fR
\fBsem_open()\fR        \fB-\fR \fB-\fR
\fBsem_close()\fR       \fB-\fR \fB-\fR
\fBsem_wait()\fR        \fBsema_wait()\ \fB-\fR
\fBsem_trywait()\fR     \fBsema_trywait()\fR    \fB-\fR
\fBsem_post()\fR        \fBsema_post()\fR       \fB-\fR
\fBsem_getvalue()\fR    \fB-\fR \fB-\fR
\fBsem_unlink()\fR      \fB-\fR \fB-\fR
\fBsem_destroy()\fR     \fBsema_destroy()\fR    \fB-\fR
.TE

.SS "Functions Related to fork(\|) Clean Up"

.TS
l l l
l l l .
\fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
\fBpthread_atfork()\fR  \fB-\fR \fB-\fR
.TE

.SS "Functions Related to Limits"

.TS
l l l
l l l .
\fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
\fBpthread_once()\fR    \fB-\fR \fBcall_once()\fR
.TE

.SS "Functions Related to Debugging"

.TS
l l l
l l l .
\fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
\fB-\fR \fBthr_stksegment()\fR  \fB-\fR
.TE

.SH LOCKING
.SS "Synchronization"
.LP
Multithreaded behavior is asynchronous, and therefore,  optimized for
concurrent and parallel processing. As threads, always from within the same
process and  sometimes from multiple processes, share global data with each
other, they are not guaranteed exclusive access to the shared data at any point
in time. Securing mutually exclusive access to shared data requires
synchronization among the threads. Both POSIX and illumos implement four
synchronization mechanisms: mutexes, condition variables, reader/writer locking
(\fIoptimized frequent-read occasional-write mutex\fR), and semaphores, where as
C11 threads only implement two mechanisms: mutexes and condition variables.
.sp
.LP
Synchronizing multiple threads diminishes their concurrency. The coarser the
grain of synchronization, that is, the larger the block of code that is locked,
the lesser the concurrency.
```

```
391 .SS "MT \fBfork()\fR"
392 .LP
393 If a threads program calls \fBfork\fR(2), it implicitly calls \fBfork1\fR(2),
394 which replicates only the calling thread. Should there be any outstanding
395 mutexes throughout the process, the application should call
396 \fBpthread_atfork\fR(3C) to wait for and acquire those mutexes prior to calling
397 \fBfork()\fR.
398 .SH SCHEDULING
399 .SS "POSIX Threads"
400 .LP
401 illumos supports the following three POSIX scheduling policies:
402 .sp
403 .ne 2
404 .na
405 \fB\fBSCHED_OTHER\fR\fR
406 .ad
407 .RS 15n
408 Traditional Timesharing scheduling policy. It is based on the timesharing (TS)
409 scheduling class.
410 .RE

412 .sp
413 .ne 2
414 .na
415 \fB\fBSCHED_FIFO\fR\fR
416 .ad
417 .RS 15n
418 First-In-First-Out scheduling policy. Threads scheduled to this policy, if not
419 preempted by a higher priority, will proceed until completion. Such threads are
420 in real-time (RT) scheduling class. The calling process must have a effective
421 user \fBID\fR of \fB0\fR.
422 .RE

424 .sp
425 .ne 2
426 .na
427 \fB\fBSCHED_RR\fR\fR
428 .ad
429 .RS 15n
430 Round-Robin scheduling policy. Threads scheduled to this policy, if not
431 preempted by a higher priority, will execute for a time period determined by
432 the system. Such threads are in real-time (RT) scheduling class and the calling
433 process must have a effective user \fBID\fR of \fB0\fR.
434 .RE

436 .sp
437 .LP
438 In addition to the POSIX-specified scheduling policies above, illumos also
439 supports these scheduling policies:
440 .sp
441 .ne 2
442 .na
443 \fB\fBSCHED_IA\fR\fR
444 .ad
445 .RS 13n
446 Threads are scheduled according to the Inter-Active Class (IA) policy as
447 described in \fBpriocntl\fR(2).
448 .RE

450 .sp
451 .ne 2
452 .na
453 \fB\fBSCHED_FSS\fR\fR
454 .ad
455 .RS 13n
456 Threads are scheduled according to the Fair-Share Class (FSS) policy as
```



```
457 described in \fBpriocntl\fR(2).
458 .RE

460 .sp
461 .ne 2
462 .na
463 \fB\fBSCHED_FX\fR\fR
464 .ad
465 .RS 13n
466 Threads are scheduled according to the Fixed-Priority Class (FX) policy as
467 described in \fBpriocntl\fR(2).
468 .RE

470 .SS "illumos Threads"
471 .LP
472 Only scheduling policy supported is \fBSCHED_OTHER\fR, which is timesharing,
473 based on the \fBTS\fR scheduling class.
474 .SH ERRORS
475 .LP
476 In a multithreaded application, \fBEINTR\fR can be returned from blocking
477 system calls when another thread calls \fBforkall\fR(2).
478 .SH USAGE
479 .SS "\fB-mt\fR compiler option"
480 .LP
481 The \fB-mt\fR compiler option compiles and links for multithreaded code. It
482 compiles source files with \(mi\fBD_REENTRANT\fR and augments the set of
483 support libraries properly.
484 .sp
485 .LP
486 Users of other compilers such as gcc and clang should manually set
487 \(mi\fBD_REENTRANT\fR on the compilation line. There are no other libraries or
488 flags necessary.
489 .SH ATTRIBUTES
490 .LP
491 See \fBattributes\fR(5) for descriptions of the following attributes:
492 .sp

494 .sp
495 .TS
496 box;
497 c | c
498 l | l .
499 ATTRIBUTE TYPE  ATTRIBUTE VALUE
500 _
501 MT-Level        MT-Safe, Fork 1-Safe
502 .TE

504 .SH SEE ALSO
505 .LP
506 \fBcrle\fR(1), \fBfork\fR(2), \fBpriocntl\fR(2), \fBlibpthread\fR(3LIB),
507 \fBlibrt\fR(3LIB), \fBlibthread\fR(3LIB), \fBpthread_atfork\fR(3C),
508 \fBpthread_create\fR(3C), \fBattributes\fR(5), \fBstandards\fR(5)
509 .sp
510 .LP
511 \fILinker and Libraries Guide\fR
```

```
**********************************************************
    8128 Sun Sep 16 19:22:58 2018
new/usr/src/man/man7/FSS.7
9842 man page typos and spelling
**********************************************************
'\" te
.\" Copyright (c) 2001, Sun Microsystems, Inc. All Rights Reserved
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH FSS 7 "May 13, 2017"
.SH NAME
FSS \- Fair share scheduler
.SH DESCRIPTION
.LP
The fair share scheduler (FSS) guarantees application performance by explicitly
allocating shares of CPU resources to projects. A share indicates a project's
entitlement to available CPU resources. Because shares are meaningful only in
comparison with other project's shares, the absolute quantity of shares is not
important. Any number that is in proportion with the desired CPU entitlement
can be used.
.sp
.LP
The goals of the FSS scheduler differ from the traditional time-sharing
scheduling class (TS). In addition to scheduling individual LWPs, the FSS
scheduler schedules projects against each other, making it impossible for any
project to acquire more CPU cycles simply by running more processes
concurrently.
.sp
.LP
A project's entitlement is individually calculated by FSS independently for
each processor set if the project contains processes bound to them. If a
project is running on more than one processor set, it can have different
entitlements on every set. A project's entitlement is defined as a ratio
between the number of shares given to a project and the sum of shares of all
active projects running on the same processor set. An active project is one
that has at least one running or runnable process. Entitlements are recomputed
whenever any project becomes active or inactive, or whenever the number of
shares is changed.
.sp
.LP
Processor sets represent virtual machines in the FSS scheduling class and
processes are scheduled independently in each processor set. That is, processes
compete with each other only if they are running on the same processor set.
When a processor set is destroyed, all processes that were bound to it are
moved to the default processor set, which always exists. Empty processor sets
(that is, sets without processors in them) have no impact on the FSS scheduler
behavior.
.sp
.LP
If a processor set contains a mix of TS/IA and FSS processes, the fairness of
the FSS scheduling class can be compromised because these classes use the same
range of priorities. Fairness is most significantly affected if processes
running in the TS scheduling class are CPU-intensive and are bound to
processors within the processor set. As a result, you should avoid having
processes from TS/IA and FSS classes share the same processor set. RT and FSS
processes use disjoint priority ranges and therefore can share processor sets.
.sp
.LP
As projects execute, their CPU usage is accumulated over time. The FSS
scheduler periodically decays CPU usages of every project by multiplying it
with a decay factor, ensuring that more recent CPU usage has greater weight
when taken into account for scheduling. The FSS scheduler continually adjusts
priorities of all processes to make each project's relative CPU usage converge
with its entitlement.
.sp
```



```
.LP
While FSS is designed to fairly allocate cycles over a long-term time period,
it is possible that projects will not receive their allocated shares worth of
CPU cycles due to uneven demand. This makes one-shot, instantaneous analysis of
FSS performance data unreliable.
.sp
.LP
Note that share is not the same as utilization. A project may be allocated 50%
of the system, although on the average, it uses just 20%. Shares serve to cap a
project's CPU usage only when there is competition from other projects running
on the same processor set. When there is no competition, utilization may be
larger than entitlement based on shares. Allocating a small share to a busy
project slows it down but does not prevent it from completing its work if the
system is not saturated.
.sp
.LP
The configuration of CPU shares is managed by the name server as a property of
the \fBproject\fR(4) database. In the following example, an entry in the
\fB/etc/project\fR file sets the number of shares for project \fBx-files\fR to
10:
.sp
.in +2
.nf
x-files:100::::project.cpu-shares=(privileged,10,none)
.fi
.in -2

.sp
.LP
Projects with undefined number of shares are given one share each. This means
that such projects are treated with equal importance. Projects with 0 shares
only run when there are no projects with non-zero shares competing for the same
processor set. The maximum number of shares that can be assigned to one project
is 65535.
.sp
.LP
You can use the \fBprctl\fR(1) command to determine the current share
assignment for a given project:
.sp
.in +2
.nf
$ prctl -n project.cpu-shares -i project x-files
.fi
.in -2

.sp
.LP
or to change the amount of shares if you have root privileges:
.sp
.in +2
.nf
# prctl -r -n project.cpu-shares -v 5 -i project x-files
.fi
.in -2

.sp
.LP
See the \fBprctl\fR(1) man page for additional information on how to modify and
examine resource controls associated with active processes, tasks, or projects
on the system. See \fBresource_controls\fR(5) for a description of the resource
controls supported in the current release of the Solaris operating system.
.sp
.LP
By default, project \fBsystem\fR (project ID 0) includes all system daemons
started by initialization scripts and has an "unlimited" amount of shares. That
is, it is always scheduled first no matter how many shares are given to other
```

```
projects.
.sp
.LP
The following command sets FSS as the default scheduler for the system:
.sp
.in +2
.nf
# dispadmin -d FSS
.fi
.in -2

.sp
.LP
This change will take effect on the next reboot. Alternatively, you can move
processes from the time-share scheduling class (as well as the special case of
init) into the FSS class without changing your default scheduling class and
rebooting by becoming \fBroot\fR, and then using the \fBpriocntl\fR(1) command,
as shown in the following example:
.sp
.in +2
.nf
# priocntl -s -c FSS -i class TS
# priocntl -s -c FSS -i pid 1
.fi
.in -2

.SH CONFIGURING SCHEDULER WITH DISPADMIN
.LP
You can use the \fBdispadmin\fR(1M) command to examine and tune the FSS
scheduler's time quantum value. Time quantum is the amount of time that a
thread is allowed to run before it must relinquish the processor. The following
example dumps the current time quantum for the fair share scheduler:
.sp
.in +2
.nf
$ dispadmin -g -c FSS
        #
        # Fair Share Scheduler Configuration
        #
        RES=1000
        #
        # Time Quantum
        #
        QUANTUM=110
.fi
.in -2

.sp
.LP
The value of the QUANTUM represents some fraction of a second with the
fractional value determined by the reciprocal value of RES. With the default
fractional value determied by the reciprocal value of RES. With the default
value of RES = 1000, the reciprocal of 1000 is .001, or milliseconds. Thus, by
default, the QUANTUM value represents the time quantum in milliseconds.
.sp
.LP
If you change the RES value using \fBdispadmin\fR with the \fB-r\fR option, you
also change the QUANTUM value. For example, instead of quantum of 110 with RES
of 1000, a quantum of 11 with a RES of 100 results. The fractional unit is
different while the amount of time is the same.
.sp
.LP
You can use the \fB-s\fR option to change the time quantum value. Note that
such changes are not preserved across reboot. Please refer to the
\fBdispadmin\fR(1M) man page for additional information.
```



```
.SH SEE ALSO
.LP
\fBprctl\fR(1), \fBpriocntl\fR(1), \fBdispadmin\fR(1M), \fBpsrset\fR(1M),
\fBpriocntl\fR(2), \fBproject\fR(4), \fBresource_controls\fR(5)
.sp
.LP
\fISystem Administration Guide:  Virtualization Using the Solaris Operating
System\fR
```