```
**********************************************************
    3816 Mon Aug 26 06:56:02 2019
new/usr/src/man/man1/audioctl.1
11622 clean up rarer mandoc lint warnings
**********************************************************
```

```
'\" t
.\"
.\" 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 2011 Nexenta Systems, Inc.  All rights reserved.
.\"
.TH AUDIOCTL 1 "Mar 18, 2011"
.SH NAME
audioctl - audio device control command line interface
.SH SYNOPSIS
.LP
.nf
\fBaudioctl\fR \fBlist-devices\fR
.fi

.LP
.nf
\fBaudioctl\fR \fBshow-device\fR [\fB-v\fR] [\fB-d\fR \fIdevice\fR]
.fi

.LP
.nf
\fBaudioctl\fR \fBshow-control\fR [\fB-v\fR] [\fB-d\fR \fIdevice\fB] [\fIcontrol
.fi

.LP
.nf
\fBaudioctl\fR \fBset-control\fR [\fB-v\fR] [\fB-d\fR \fIdevice\fB] \fIcontrol\f
.fi

.LP
.nf
\fBaudioctl\fR \fBsave-controls\fR [\fB-d\fR \fIdevice\fB] [\fB-f\fR] \fIfile\fR
.fi

.LP
.nf
\fBaudioctl\fR \fBload-controls\fR [\fB-d\fR \fIdevice\fB] \fIfile\fR
.fi

.SH DESCRIPTION
The \fBaudioctl\fP command is used to control various settings and features
of audio devices, including mixer settings such as playback volume and
record gain.
.LP

.SH SUBCOMMANDS
The \fBaudioctl\fR command supports the following subcommands.
.sp
.ne 2
.na
\fBlist-devices\fR
.ad
```



```
.sp .6
.RS 4n
List all the audio devices on the system.
.RE

.sp
.ne 2
.na
\fBshow-device\fR [\fB-v\fR] [\fB-d\fR \fIdevice\fR] [\fBcontrol\fR ... ]
.ad
.sp .6
.RS 4n
Display an informational overview of \fIdevice\fR (or the default device if
not specified).  If the \fB-v\fR option is specified, then more detail
will be displayed.
.RE

.sp
.ne 2
.na
\fBshow-control\fR [\fB-v\fR] [\fB-d\fR \fIdevice\fR]
.ad
.sp .6
.RS 4n
Display the control settings for one or more \fIcontrols\fR of a
\fIdevice\fR (or the default device if not specified).  If no \fIcontrol\fR is
specified, then the settings for all controls will be displayed. If
the \fB-v\fR option is specified, then more detail will be displayed.
.RE

.sp
.ne 2
.na
\fBset-control\fR [\fB-v\fR] [\fB-d\fR \fIdevice\fB] \fIcontrol\fR \fIvalue\fR
.ad
.sp .6
.RS 4n
Set the control settings for a \fIcontrol\fR of a \fIdevice\fR (or the
default device if not specified) to \fIvalue\fR.  If the \fB-v\fR
option is specified, then more verbose output be displayed.
.RE

.sp
.ne 2
.na
\fBsave-controls\fR [\fB-f\fR] [\fB-d\fR \fIdevice\fB] \fIfile\fR
.ad
.sp .6
.RS 4n
Save all control settings of the \fIdevice\fR (or the default device if not
specified) to the named \fIfile\fR.  This subcommand will not replace
or modify an existing file unless \fB-f\fR (force) is specified.
.RE

.sp
.ne 2
.na
\fBload-controls\fR [\fB-d\fR \fIdevice\fB] \fIfile\fR
.ad
.sp .6
.RS 4n
Load previously saved control settings from a \fIfile\fR
into a \fIdevice\fR (or all audio devices if not
specified).
.RE
```

```
125 .SH ENVIRONMENT VARIABLES
126 .ne 2
127 .na
128 \fB\fBAUDIODEV\fR\fR
129 .ad
130 .RS 12n
131 The full path name of the default audio device to use if one
132 is not specified on the command line.  If this variable is not set,
133 \fB/dev/audio\fR is used.
134 .RE


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

141 .sp
142 .TS
143 box;
144 c | c
145 l | l .
146 ATTRIBUTE TYPE  ATTRIBUTE VALUE
147 _
148 Interface Stability     See below.
149 .TE
150 .LP
151 The \fBaudioctl\fR command and its subcommands are Committed.  The
152 names of controls, their values, and device names are Uncommitted.
153 The display output is intended for human consumption, and is Not An
154 Interface.  The format of the state files used by the
155 \fBsave-controls\fR and \fBload-controls\fR subcommands is Committed
156 Private.
161 .LP

157 .SH SEE ALSO
158 \fBaudioplay\fR(1), \fBaudiorecord\fR(1), \fBdsp\fR(7I),
159 \fBmixer\fR(7I), \fBattributes\fR(5)
```

```
**********************************************************
    9534 Mon Aug 26 06:56:02 2019
new/usr/src/man/man1/bc.1
11622 clean up rarer mandoc lint warnings
**********************************************************
.\"
.\" 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) 2003, Sun Microsystems, Inc.  All Rights Reserved.
.\"
.TH BC 1 "Aug 29, 2003"
.SH NAME
bc \- arbitrary precision arithmetic language
.SH SYNOPSIS
.LP
.nf
\fB/usr/bin/bc\fR [\fB-c\fR] [\fB-l\fR] [\fIfile\fR]...
.fi

.LP
.nf
\fB/usr/xpg6/bin/bc\fR [\fB-c\fR] [\fB-l\fR] [\fIfile\fR]...
.fi

.SH DESCRIPTION
```



```
.sp
.LP
The \fBbc\fR utility implements an arbitrary precision calculator. It takes
input from any files given, then reads from the standard input. If the standard
input and standard output to \fBbc\fR are attached to a terminal, the
invocation of \fBbc\fR is  \fIinteractive\fR, causing behavioral constraints
described in the following sections. \fBbc\fR processes a language that
resembles C and is a preprocessor for the desk calculator program \fBdc\fR,
which it invokes automatically unless the \fB-c\fR option is specified. In this
case the \fBdc\fR input is sent to the standard output instead.
.SH USAGE
.sp
.LP
The syntax for  \fBbc\fR programs is as follows:
.sp
.ne 2
.na
\fB\fIL\fR\fR
.ad
.RS 5n
Means a letter \fBa\fR\(mi\fBz\fR,
.RE

.sp
.ne 2
.na
\fB\fIE\fR\fR
.ad
.RS 5n
Means an expression: a (mathematical or logical) value, an operand  that takes
a value, or a combination of operands and operators that evaluates to a value,
.RE

.sp
.ne 2
.na
\fB\fIS\fR\fR
.ad
.RS 5n
Means a statement.
.RE

.SS "Comments"
.sp
.LP
Enclosed in \fB/*\fR and \fB*/\fR.
.SS "Names (Operands)"
.br
.in +2
Simple variables: \fIL\fR.
.in -2
.br
.in +2
Array elements: \fIL\fR [ \fIE\fR ] (up to \fBBC_DIM_MAX\fR dimensions).
.in -2
.br
.in +2
The words \fBibase\fR, \fBobase\fR (limited to \fBBC_BASE_MAX\fR), and
\fBscale\fR (limited to \fBBC_SCALE_MAX\fR).
.in -2
.SS "Other Operands"
.sp
.LP
Arbitrarily long numbers with optional sign and decimal point. Strings of fewer
than \fBBC_STRING_MAX\fR characters, between double quotes ("). \fB(\fR \fIE\fR
\fB)\fR
```

```
.sp
.ne 2
.na
\fB\fBsqrt ( \fR\fIE\fR\fB )\fR\fR
.ad
.RS 21n
Square root
.RE

.sp
.ne 2
.na
\fB\fBlength ( \fR\fIE\fR\fB )\fR\fR
.ad
.RS 21n
Number of significant decimal digits.
.RE

.sp
.ne 2
.na
\fB\fBscale ( \fR\fIE\fR\fB )\fR\fR
.ad
.RS 21n
Number of digits right of decimal point.
.RE

.sp
.ne 2
.na
\fB\fIL \fR\fB( \fR\fIE\fR , ... , \fIE\fR\fB )\fR\fR
.ad
.RS 21n

.RE

.SS "Operators"
.sp
.ne 2
.na
\fB\fB+   \(mi   *   /   %   ^\fR\fR
.ad
.sp .6
.RS 4n
(\fB%\fR is remainder; \fB^\fR is power)
.RE

.sp
.ne 2
.na
\fB\fB++   \(mi\(mi \fR\fR
.ad
.sp .6
.RS 4n
(prefix and postfix; apply to names)
.RE

.sp
.ne 2
.na
\fB\fB==   <=   >=   !=   <   >\fR\fR
.ad
.sp .6
.RS 4n

.RE
```



```
.sp
.ne 2
.na
\fB\fB=   =+   =\(mi   =*   =/   =%   =^\fR\fR
.ad
.sp .6
.RS 4n

.RE

.SS "Statements"
.br
.in +2
\fIE\fR
.in -2
.br
.in +2
\fB{\fR \fIS\fR \fB;\fR.\|.\|. \fB;\fR \fIS\fR \fB}\fR
.in -2
.br
.in +2
\fBif (\fR \fIE\fR \fB)\fR \fIS\fR
.in -2
.br
.in +2
\fBwhile (\fR \fIE\fR \fB)\fR \fIS\fR
.in -2
.br
.in +2
\fBfor (\fR \fIE\fR \fB;\fR \fIE\fR \fB;\fR \fIE\fR \fB)\fR \fIS\fR
.in -2
.br
.in +2
null statement
.in -2
.br
.in +2
\fBbreak\fR
.in -2
.br
.in +2
\fBquit\fR
.in -2
.sp
.LP
\&.string
.SS "Function Definitions"
.br
.in +2
\fBdefine\fR \fIL\fR \fB(\fR \fIL\fR \fB,\fR.\|.\|.\|\fB,\fR \fIL\fR \fB) {\fR
.in -2
.br
.in +2
\fBauto\fR \fIL\fR \fB,\fR.\|.\|.\|\fB,\fR \fIL\fR
.in -2
.br
.in +2
\fIS\fR \fB;\fR.\|.\|.\| \fIS\fR
.in -2
.br
.in +2
\fBreturn (\fR \fIE\fR \fB)\fR
.in -2
.br
.in +2
```

```
247 \fB}\fR
248 .in -2
249 .SS "Functions in \fB-l\fR Math Library"
263 .sp
250 .ne 2
251 .na
252 \fB\fBs(\fR\fIx\fR\fB)\fR\fR
253 .ad
254 .RS 10n
255 sine
256 .RE

258 .sp
259 .ne 2
260 .na
261 \fB\fBc(\fR\fIx\fR\fB)\fR\fR
262 .ad
263 .RS 10n
264 cosine
265 .RE

267 .sp
268 .ne 2
269 .na
270 \fB\fBe(\fR\fIx\fR\fB)\fR\fR
271 .ad
272 .RS 10n
273 exponential
274 .RE

276 .sp
277 .ne 2
278 .na
279 \fB\fBl(\fR\fIx\fR\fB)\fR\fR
280 .ad
281 .RS 10n
282 log
283 .RE

285 .sp
286 .ne 2
287 .na
288 \fB\fBa(\fR\fIx\fR\fB)\fR\fR
289 .ad
290 .RS 10n
291 arctangent
292 .RE

294 .sp
295 .ne 2
296 .na
297 \fB\fBj(\fR\fIn\fR\fB,\fR\fIx)\fR\fR
298 .ad
299 .RS 10n
300 Bessel function
301 .RE

303 .sp
304 .LP
305 All function arguments are passed by value.
306 .sp
307 .LP
308 The value of a statement that is an expression is printed unless the main
309 operator is an assignment. Either semicolons or new-lines may separate
310 statements. Assignment to \fBscale\fR influences the number of digits to be
311 retained on arithmetic operations in the manner of \fBdc\fR. Assignments to
```



```
312 \fBibase\fR or \fBobase\fR set the input and output number radix respectively.
313 .sp
314 .LP
315 The same letter may be used as an array, a function, and a simple variable
316 simultaneously. All variables are global to the program. \fBauto\fR variables
317 are stacked during function calls. When using arrays as function arguments or
318 defining them as automatic variables, empty square brackets must follow the
319 array name.
320 .SH OPTIONS
335 .sp
336 .LP
321 The following operands are supported:
322 .sp
323 .ne 2
324 .na
325 \fB\fB-c\fR\fR
326 .ad
327 .RS 6n
328 Compiles only. The output is \fBdc\fR commands that are sent to the standard
329 output.
330 .RE

332 .SS "/usr/bin/bc"
349 .sp
333 .ne 2
334 .na
335 \fB\fB-l\fR\fR
336 .ad
337 .RS 6n
338 Defines the math functions and initializes \fBscale\fR to \fB20\fR, instead of
339 the default zero.
340 .RE

342 .SS "/usr/xpg6/bin/bc"
360 .sp
343 .ne 2
344 .na
345 \fB\fB-l\fR\fR
346 .ad
347 .RS 6n
348 Defines the math functions and initializes \fBscale\fR to \fB20\fR, instead of
349 the default zero. All math results have the scale of \fB20\fR.
350 .RE

352 .SH OPERANDS
371 .sp
372 .LP
353 The following operands are supported:
354 .sp
355 .ne 2
356 .na
357 \fB\fIfile\fR\fR
358 .ad
359 .RS 8n
360 A pathname of a text file containing \fBbc\fR program statements. After all
361 cases of \fIfile\fR have been read, \fBbc\fR reads the standard input.
362 .RE

364 .SH EXAMPLES
385 .LP
365 \fBExample 1 \fRSetting the precision of a variable
366 .sp
367 .LP
368 In the shell, the following assigns an approximation of the first ten digits of
369 \fBn\fR to the variable \fIx\fR:
```

```
.sp
.in +2
.nf
\fBx=$(printf "%s\en" 'scale = 10; 104348/33215' | bc)\fR
.fi
.in -2
.sp

.LP
\fBExample 2 \fRDefining a computing function
.sp
.LP
Defines a function to compute an approximate value of the exponential function:

.sp
.in +2
.nf
\fBscale = 20
define e(x){
        auto a, b, c, i, s
        a = 1
        b = 1
        s = 1
        for(i=1; 1==1; i++){
                a = a*x
                b = b*i
                c = a/b
                if(c == 0) return(s)
                s = s+c
        }
}\fR
.fi
.in -2
.sp

.LP
\fBExample 3 \fRPrinting the approximate values of the function
.sp
.LP
Prints approximate values of the exponential function of the first ten
integers:

.sp
.in +2
.nf
\fBfor(i=1; i<=10; i++) e(i)\fR
.fi
.in -2
.sp

.sp
.LP
or

.sp
.in +2
.nf
\fBfor (i = 1; i <= 10; ++i) {         e(i) }\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 \fBbc\fR: \fBLANG\fR, \fBLC_ALL\fR,
\fBLC_CTYPE\fR, \fBLC_MESSAGES\fR, and \fBNLSPATH\fR.
.SH EXIT STATUS
.sp
.LP
The following exit values are returned:
.sp
.ne 2
.na
\fB\fB0\fR\fR
.ad
.RS 15n
All input files were processed successfully.
.RE

.sp
.ne 2
.na
\fB\fBunspecified\fR\fR
.ad
.RS 15n
An error occurred.
.RE

.SH FILES
.sp
.ne 2
.na
\fB\fB/usr/lib/lib.b\fR\fR
.ad
.RS 25n
mathematical library
.RE

.sp
.ne 2
.na
\fB\fB/usr/include/limits.h\fR\fR
.ad
.RS 25n
to define BC_ parameters
.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     Standard
.TE

.SH SEE ALSO
.sp
.LP
\fBdc\fR(1), \fBawk\fR(1), \fBattributes\fR(5), \fBenviron\fR(5),
\fBstandards\fR(5)
.SH NOTES
.sp
```

```
 524 .LP
 493 The \fBbc\fR command does not recognize the logical operators \fB&&\fR and
 494 \fB|\||\fR.
 495 .sp
 496 .LP
 497 The \fBfor\fR statement must have all three expressions (\fIE\fR's).
```

```
**********************************************************
    9599 Mon Aug 26 06:56:02 2019
new/usr/src/man/man1/localedef.1
11622 clean up rarer mandoc lint warnings
**********************************************************
```

```
.\"
.\" 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]
\fIlocaledef\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.
```

```
125 .RE

127 .SH OPTIONS
130 .LP
128 The following options are supported:
129 .sp
130 .ne 2
131 .na
132 \fB\fB-c\fR\fR
133 .ad
134 .RS 23n
135 Creates permanent output even if warning messages have been issued.
136 .RE

138 .sp
139 .ne 2
140 .na
141 \fB\fB-v\fR\fR
142 .ad
143 .RS 23n
144 Emit verbose debugging output on standard output.
145 .RE

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

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

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

189 .sp
```



```
190 .ne 2
191 .na
192 \fB\fB-i\fR \fIsourcefile\fR\fR
193 .ad
194 .RS 23n
195 The path name of a file containing the source definitions. If this option is
196 not present, source definitions will be read from standard input.
197 .RE

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

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

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

247 .sp
248 .ne 2
249 .na
250 \fB\fB1\fR\fR
251 .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.
```

```
**********************************************************
   16986 Mon Aug 26 06:56:02 2019
new/usr/src/man/man1/sed.1
11622 clean up rarer mandoc lint warnings
**********************************************************
```

```
.\" Copyright (c) 1992, 1993
.\"     The Regents of the University of California.  All rights reserved.
.\"
.\" Copyright 2011 Nexenta Systems, Inc. All rights reserved.
.\"
.\" This code is derived from software contributed to Berkeley by
.\" the Institute of Electrical and Electronics Engineers, Inc.
.\"
.\" 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.
.\" 4. Neither the name of the University nor the names of its contributors
.\"    may be used to endorse or promote products derived from this software
.\"    without specific prior written permission.
.\"
.\" THIS SOFTWARE IS PROVIDED BY THE REGENTS 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 REGENTS 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.
.\"
.TH SED 1 "Feb 14, 2015"
.SH NAME
\fBsed\fP
\- stream editor
.SH SYNOPSIS
\fBsed\fP
[\fB\-Ealnr\fP]
\fIcommand\fP
[\fIfile ...\fP]
.br
\fBsed\fP
[\fB\-Ealnr\fP]
[\fB\-e\fP \fIcommand\fP]
[\fB\-f\fP \fIcommand_file\fP]
[\fB\-I\fP[\fIextension\fP] | \fB\-i\fP[\fIextension\fP]]
[\fIfile ...\fP]
.SH DESCRIPTION
The
\fBsed\fP
utility reads the specified files, or the standard input if no files
are specified, modifying the input as specified by a list of commands.
The input is then written to the standard output.

A single command may be specified as the first argument to
\fB.\fP
Multiple commands may be specified by using the
\fB\-e\fP
or
\fB\-f\fP
```



```
options.
All commands are applied to the input in the order they are specified
regardless of their origin.

The following options are available:
.TP
\fB\-E\fP
Interpret regular expressions as extended (modern) regular expressions
rather than basic regular expressions (BRE's).
The
\fBregex\fP(5)
manual page fully describes both formats.
.TP
\fB\-a\fP
The files listed as parameters for the
``w''
functions are created (or truncated) before any processing begins,
by default.
The
\fB\-a\fP
option causes
\fBsed\fP
to delay opening each file until a command containing the related
``w''
function is applied to a line of input.
.TP
\fB\-e\fP \fIcommand\fP
Append the editing commands specified by the
\fIcommand\fP
argument
to the list of commands.
.TP
\fB\-f\fP \fIcommand_file\fP
Append the editing commands found in the file
\fIcommand_file\fP
to the list of commands.
The editing commands should each be listed on a separate line.
.TP
\fB\-I\fP[\fIextension\fP]
Edit files in-place, saving backups if \fIextension\fP was specified.
It is not recommended to omit saving backups when in-place editing files,
as you risk corruption or partial content in situations where disk
space is exhausted, etc.

Note that in-place editing with
\fB\-I\fP
still takes place in a single continuous line address space covering
all files, although each file preserves its individuality instead of
forming one output stream.
The line counter is never reset between files, address ranges can span
file boundaries, and the
``$''
address matches only the last line of the last file.
(See
.B "Sed Addresses" . )
That can lead to unexpected results in many cases of in-place editing,
where using
\fB\-i\fP
is desired.
.TP
\fB\-i\fP[\fIextension\fP]
Edit files in-place similarly to
\fB\-I\fP,
but treat each file independently from other files.
In particular, line numbers in each file start at 1,
the
```

```
128 ‘‘$’’
129 address matches the last line of the current file,
130 and address ranges are limited to the current file.
131 (See
132 .B "Sed Addresses" . )
133 The net result is as though each file were edited by a separate
134 \fBsed\fP
135 instance.
136 .TP
137 \fB\-l\fP
138 Make output line buffered.
139 .TP
140 \fB\-n\fP
141 By default, each line of input is echoed to the standard output after
142 all of the commands have been applied to it.
143 The
144 \fB\-n\fP
145 option suppresses this behavior.
146 .TP
147 \fB\-r\fP
148 Same as
149 \fB\-E\fP
150 for compatibility with GNU sed.

152 The form of a
153 \fBsed\fP
154 command is as follows:

156 [address[,address]]function[arguments]

158 Whitespace may be inserted before the first address and the function
159 portions of the command.

161 Normally,
162 \fBsed\fP
163 cyclically copies a line of input, not including its terminating newline
164 character, into a
165 .IR "pattern space" ,
166 (unless there is something left after a
167 ‘‘D’’
168 function),
169 applies all of the commands with addresses that select that pattern space,
170 copies the pattern space to the standard output, appending a newline, and
171 deletes the pattern space.

173 Some of the functions use a
174 .IR "hold space"
175 to save all or part of the pattern space for subsequent retrieval.
176 .SH "Sed Addresses"
177 An address is not required, but if specified must have one of the
178 following formats:
179 .IP \(bu
180 a number that counts
181 input lines
182 cumulatively across input files (or in each file independently
183 if a
184 \fB\-i\fP
185 option is in effect);
186 .IP \(bu
187 a dollar
188 (‘‘$’’)
189 character that addresses the last line of input (or the last line
190 of the current file if a
191 \fB\-i\fP
192 option was specified);
193 .IP \(bu
```



```
194 a context address
195 that consists of a regular expression preceded and followed by a
196 delimiter. The closing delimiter can also optionally be followed by the
197 ‘‘I’’
198 character, to indicate that the regular expression is to be matched
199 in a case-insensitive way.

201 A command line with no addresses selects every pattern space.

203 A command line with one address selects all of the pattern spaces
204 that match the address.

206 A command line with two addresses selects an inclusive range.
207 This
208 range starts with the first pattern space that matches the first
209 address.
210 The end of the range is the next following pattern space
211 that matches the second address.
212 If the second address is a number
213 less than or equal to the line number first selected, only that
214 line is selected.
215 The number in the second address may be prefixed with a
216 (‘‘\&+’’)
217 to specify the number of lines to match after the first pattern.
218 In the case when the second address is a context
219 address,
220 \fBsed\fP
221 does not re-match the second address against the
222 pattern space that matched the first address.
223 Starting at the
224 first line following the selected range,
225 \fBsed\fP
226 starts looking again for the first address.

228 Editing commands can be applied to non-selected pattern spaces by use
229 of the exclamation character
230 (‘‘\&!’’)
231 function.
232 .SH "Sed Regular Expressions"
233 The regular expressions used in
234 \fB,\fP
235 by default, are basic regular expressions (BREs, see
236 \fBregex\fP(5)
237 for more information), but extended (modern) regular expressions can be used
238 instead if the
239 \fB\-E\fP
240 flag is given.
241 In addition,
242 \fBsed\fP
243 has the following two additions to regular expressions:

245 .IP 1.
246 In a context address, any character other than a backslash
247 (‘‘\e’’)
248 or newline character may be used to delimit the regular expression.
249 The opening delimiter needs to be preceded by a backslash
250 unless it is a slash.
251 For example, the context address
252 \exabcx
253 is equivalent to
254 /abc/ .
255 Also, putting a backslash character before the delimiting character
256 within the regular expression causes the character to be treated literally.
257 For example, in the context address
258 \exabc\exdefx ,
259 the RE delimiter is an
```

```
260 ‘‘x’’
261 and the second
262 ‘‘x’’
263 stands for itself, so that the regular expression is
264 ‘‘abcxdef’’.

266 .IP 2.
267 The escape sequence \en matches a newline character embedded in the
268 pattern space.
269 You cannot, however, use a literal newline character in an address or
270 in the substitute command.

272 One special feature of
273 \fBsed\fP
274 regular expressions is that they can default to the last regular
275 expression used.
276 If a regular expression is empty, i.e., just the delimiter characters
277 are specified, the last regular expression encountered is used instead.
278 The last regular expression is defined as the last regular expression
279 used as part of an address or substitute command, and at run-time, not
280 compile-time.
281 For example, the command
282 ‘‘/abc/s//XXX/’’
283 will substitute
284 ‘‘XXX’’
285 for the pattern
286 ‘‘abc’’.
287 .SH "Sed Functions"
288 In the following list of commands, the maximum number of permissible
289 addresses for each command is indicated by [0addr], [1addr], or [2addr],
290 representing zero, one, or two addresses.

292 The argument
293 .IR text
294 consists of one or more lines.
295 To embed a newline in the text, precede it with a backslash.
296 Other backslashes in text are deleted and the following character
297 taken literally.

299 The
300 ‘‘r’’
301 and
302 ‘‘w’’
303 functions take an optional file parameter, which should be separated
304 from the function letter by white space.
305 Each file given as an argument to
306 \fBsed\fP
307 is created (or its contents truncated) before any input processing begins.

309 The
310 ‘‘b’’,
311 ‘‘r’’,
312 ‘‘s’’,
313 ‘‘t’’,
314 ‘‘w’’,
315 ‘‘y’’,
316 ‘‘\&!’’,
317 and
318 ‘‘\&:’’
319 functions all accept additional arguments.
320 The following synopses indicate which arguments have to be separated from
321 the function letters by white space characters.

323 Two of the functions take a function-list.
324 This is a list of
325 \fBsed\fP
```



```
326 functions separated by newlines, as follows:

328 { function
329 .br
329   function
331 .br
330   ...
333 .br
331   function
332 .br
333 }
337 .br

335 The
336 ‘‘{’’
337 can be preceded by white space and can be followed by white space.
338 The function can be preceded by white space.
339 The terminating
340 ‘‘}’’
341 must be preceded by a newline or optional white space.

343 .TP
344 [2addr] function-list
345 Execute function-list only when the pattern space is selected.

347 .TP
348 [1addr]a\e
349 .TP
350 text
351 Write
352 .IR text
353 to standard output immediately before each attempt to read a line of input,
354 whether by executing the
355 ‘‘N’’
356 function or by beginning a new cycle.

358 .TP
359 [2addr]b[label]
360 Branch to the
361 ‘‘\&:’’
362 function with the specified label.
363 If the label is not specified, branch to the end of the script.

365 .TP
366 [2addr]c\e
367 .TP
368 text
369 Delete the pattern space.
370 With 0 or 1 address or at the end of a 2-address range,
371 .IR text
372 is written to the standard output.

374 .TP
375 [2addr]d
376 Delete the pattern space and start the next cycle.

378 .TP
379 [2addr]D
380 Delete the initial segment of the pattern space through the first
381 newline character and start the next cycle.

383 .TP
384 [2addr]g
385 Replace the contents of the pattern space with the contents of the
386 hold space.
```

```
.TP
[2addr]G
Append a newline character followed by the contents of the hold space
to the pattern space.

.TP
[2addr]h
Replace the contents of the hold space with the contents of the
pattern space.

.TP
[2addr]H
Append a newline character followed by the contents of the pattern space
to the hold space.

.TP
[1addr]i\e
.TP
text
Write
.IR text
to the standard output.

.TP
[2addr]l
(The letter ell.)
Write the pattern space to the standard output in a visually unambiguous
form.
This form is as follows:

.TP
backslash
\e\e
.TP
alert
\ea
.TP
form-feed
\ef
.TP
carriage-return
\er
.TP
tab
\et
.TP
vertical tab
\ev

Nonprintable characters are written as three-digit octal numbers (with a
preceding backslash) for each byte in the character (most significant byte
first).
Long lines are folded, with the point of folding indicated by displaying
a backslash followed by a newline.
The end of each line is marked with a
‘‘$’’.

.TP
[2addr]n
Write the pattern space to the standard output if the default output has
not been suppressed, and replace the pattern space with the next line of
input.

.TP
[2addr]N
Append the next line of input to the pattern space, using an embedded
```



```
newline character to separate the appended material from the original
contents.
Note that the current line number changes.

.TP
[2addr]p
Write the pattern space to standard output.

.TP
[2addr]P
Write the pattern space, up to the first newline character to the
standard output.

.TP
[1addr]q
Branch to the end of the script and quit without starting a new cycle.

.TP
[1addr]r file
Copy the contents of
.IR file
to the standard output immediately before the next attempt to read a
line of input.
If
.IR file
cannot be read for any reason, it is silently ignored and no error
condition is set.

.TP
[2addr]s/regular expression/replacement/flags
Substitute the replacement string for the first instance of the regular
expression in the pattern space.
Any character other than backslash or newline can be used instead of
a slash to delimit the RE and the replacement.
Within the RE and the replacement, the RE delimiter itself can be used as
a literal character if it is preceded by a backslash.

An ampersand
(‘‘&’’)
appearing in the replacement is replaced by the string matching the RE.
The special meaning of
‘‘&’’
in this context can be suppressed by preceding it by a backslash.
The string
‘‘\e#’’,
where
‘‘#’’
is a digit, is replaced by the text matched
by the corresponding backreference expression (see
\fBregex\fP(5)) .

A line can be split by substituting a newline character into it.
To specify a newline character in the replacement string, precede it with
a backslash.

The value of
.IR flags
in the substitute function is zero or more of the following:
.TP
\fIN\fP
Make the substitution only for the
\fIN\fP’th
occurrence of the regular expression in the pattern space.
.TP
g
Make the substitution for all non-overlapping matches of the
```

```
520 regular expression, not just the first one.
521 .TP
522 p
523 Write the pattern space to standard output if a replacement was made.
524 If the replacement string is identical to that which it replaces, it
525 is still considered to have been a replacement.
526 .TP
527 w file
528 Append the pattern space to
529 .IR file
530 if a replacement was made.
531 If the replacement string is identical to that which it replaces, it
532 is still considered to have been a replacement.
533 .TP
534 I
535 Match the regular expression in a case-insensitive way.

537 .TP
538 [2addr]t [label]
539 Branch to the
540 ‘‘\&:’’
541 function bearing the label if any substitutions have been made since the
542 most recent reading of an input line or execution of a
543 ‘‘t’’
544 function.
545 If no label is specified, branch to the end of the script.

547 .TP
548 [2addr]w file
549 Append the pattern space to the
550 .IR file .

552 .TP
553 [2addr]x
554 Swap the contents of the pattern and hold spaces.

556 .TP
557 [2addr]y/string1/string2/
558 Replace all occurrences of characters in
559 .IR string1
560 in the pattern space with the corresponding characters from
561 .IR string2 .
562 Any character other than a backslash or newline can be used instead of
563 a slash to delimit the strings.
564 Within
565 .IR string1
566 and
567 .IR string2 ,
568 a backslash followed by any character other than a newline is that literal
569 character, and a backslash followed by an ‘‘n’’ is replaced by a newline
570 character.

572 .TP
573 [2addr]!function
574 .TP
575 [2addr]!function-list
576 Apply the function or function-list only to the lines that are
577 .IR not
578 selected by the address(es).

580 .TP
581 [0addr]:label
582 This function does nothing; it bears a label to which the
583 ‘‘b’’
584 and
585 ‘‘t’’
```



```
586 commands may branch.

588 .TP
589 [1addr]=
590 Write the line number to the standard output followed by a newline
591 character.

593 .TP
594 [0addr]
595 Empty lines are ignored.

597 .TP
598 [0addr]#
599 The
600 ‘‘#’’
601 and the remainder of the line are ignored (treated as a comment), with
602 the single exception that if the first two characters in the file are
603 ‘‘#n’’,
604 the default output is suppressed.
605 This is the same as specifying the
606 \fB\-n\fP
607 option on the command line.
608 .SH ENVIRONMENT
609 The
610 .IR COLUMNS , LANG , LC_ALL , LC_CTYPE
611 and
612 .IR LC_COLLATE
613 environment variables affect the execution of
614 \fBsed\fP
615 as described in
616 \fBenviron\fP(5).
617 .SH EXIT STATUS
618 The \fBsed\fP utility exits 0 on success, and >0 if an error occurs.
619 .SH SEE ALSO
620 \fBawk\fP(1),
621 \fBed\fP(1),
622 \fBgrep\fP(1),
623 \fBregex\fP(5)
624 .SH STANDARDS
625 The
626 \fBsed\fP
627 utility is expected to be a superset of the IEEE Std 1003.2 (‘‘POSIX.2’’)
628 specification.

630 The
631 \fB\-E\fP, I , a
632 and
633 \fB\-i\fP
634 options, the prefixing
635 ‘‘\&+’’
636 in the second member of an address range,
637 as well as the
638 ‘‘I’’
639 flag to the address regular expression and substitution command are
640 non-standard extensions and may not be available on other operating systems.
641 .SH HISTORY
642 A
643 \fBsed\fP
644 command, written by L. E. McMahon, appeared in Version 7 AT&T UNIX.
645 .SH AUTHORS

647 "Diomidis D. Spinellis" <dds@FreeBSD.org>
648 .SH BUGS
649 Multibyte characters containing a byte with value 0x5C (ASCII ‘\e’)
650 may be incorrectly treated as line continuation characters in arguments to the
651 ‘‘a’’,
```

‘‘c’’
and
‘‘i’’
commands.
Multibyte characters cannot be used as delimiters with the
‘‘s’’
and
‘‘y’’
commands.

```
**********************************************************
    5369 Mon Aug 26 06:56:02 2019
new/usr/src/man/man1/whois.1
11622 clean up rarer mandoc lint warnings
**********************************************************
.TH WHOIS 1 "Oct 2, 2009"
.SH NAME
\fBwhois\fP
\- Internet domain name and network number directory service
.SH SYNOPSIS
.br
\fBwhois\fP
[\fB\-aAbfgiIklmQr\fP]
[\fB\-c\fP \fIcountry-code\fP | \fIFl\fP h \fIhost\fP]
[\fB\-p\fP \fIport\fP]
\fIname\fP...
.SH DESCRIPTION
The
\fBwhois\fP
utility looks up records in the databases maintained by several
Network Information Centers (NICs).

The options are as follows:
.TP
\fB\-a\fP
Use the American Registry for Internet Numbers (ARIN) database.
It contains network numbers used in those parts of the world covered neither by
APNIC, AfriNIC, LACNIC, nor by RIPE.

(Hint: All point of contact handles in the ARIN whois database end with -ARIN.)

.TP
\fB\-A\fP
Use the Asia/Pacific Network Information Center (APNIC) database.
It contains network numbers used in East Asia, Australia,
New Zealand, and the Pacific islands.
.TP
\fB\-b\fP
Use the Network Abuse Clearinghouse database.
It contains addresses to which network abuse should be reported,
indexed by domain name.
.TP
\fB\-c\fP \fIcountry-code\fP
This is the equivalent of using the
\fB\-h\fP
option with an argument of \fIcountry-code\fP.whois-servers.net.
.TP
\fB\-f\fP
Use the African Network Information Centre (AfriNIC) database.
It contains network numbers used in Africa and the islands of the
western Indian Ocean.
.TP
\fB\-g\fP
Use the US non-military federal government database, which contains points of
contact for subdomains of
\fI\&.GOV\fP.
.TP
\fB\-h\fP \fIhost\fP
Use the specified host instead of the default variant.
Either a host name or an IP address may be specified.

By default
\fBwhois\fP
constructs the name of a whois server to use from the top-level domain (TLD)
of the supplied (single) argument, and appending .whois-servers.net .
This effectively allows a suitable whois server to be selected
```



```
automatically for a large number of TLDs.

In the event that an IP
address is specified, the whois server will default to the American
Registry for Internet Numbers (ARIN).
If a query to ARIN references APNIC, AfriNIC, LACNIC, or RIPE,
that server will be queried also, provided that the
\fB\-Q\fP
option is not specified.

If the query is not a domain name or IP address,
\fBwhois\fP
will fall back to
\fIwhois.crsnic.net\fP.
.TP
\fB\-i\fP
Use the Network Solutions Registry for Internet Numbers
(\fIwhois.networksolutions.com\fP)
database.
It contains network numbers and domain contact information for most of
\fI\&.COM\fP,.NET,.ORG
and
\fI\&.EDU\fP
domains.

.B NOTE !
The registration of these domains is now done by a number of
independent and competing registrars and this database holds no information
on the domains registered by organizations other than Network Solutions, Inc.
Also, note that the InterNIC database
(\fIwhois.internic.net\fP)
is no longer handled by Network Solutions, Inc.
For details, see
\fIhttp://www.internic.net/\fP.

(Hint: Contact information, identified by the term
.IR handle ,
can be looked up by prefixing "handle" to the NIC
handle in the query.)
.TP
\fB\-I\fP
Use the Internet Assigned Numbers Authority (IANA) database.
It contains network information for top-level domains.
.TP
\fB\-k\fP
Use the National Internet Development Agency of Korea’s (KRNIC)
database.
It contains network numbers and domain contact information
for Korea.
.TP
\fB\-l\fP
Use the Latin American and Caribbean IP address Regional Registry
(Tn LACNIC)
database.
It contains network numbers used in much of Latin America and the
Caribbean.
.TP
\fB\-m\fP
Use the Route Arbiter Database (RADB) database.
It contains route policy specifications for a large
number of operators’ networks.
.TP
\fB\-p\fP \fIport\fP
Connect to the whois server on
\fIport\fP.
If this option is not specified,
```

```
 127 \fBwhois\fP
 128 defaults to port 43.
 129 .TP
 130 \fB\-Q\fP
 131 Do a quick lookup.
 132 This means that
 133 \fBwhois\fP
 134 will not attempt to lookup the name in the authoritative whois
 135 server (if one is listed).
 136 This option has no effect when combined with any other options.
 137 .TP
 138 \fB\-r\fP
 139 Use the R\(aaeseaux IP Europ\(aaeens (RIPE) database.
 140 It contains network numbers and domain contact information
 141 for Europe.

 143 The operands specified to
 144 \fBwhois\fP
 145 are treated independently and may be used
 146 as queries on different whois servers.
 147 .SH EXIT STATUS
 148 The \fBwhois\fP utility exits 0 on success, and >0 if an error occurs.
 149 .SH EXAMPLES
 150 Most types of data, such as domain names and IP addresses, can be used as
 151 arguments to
 152 \fBwhois\fP
 153 without any options, and
 154 \fBwhois\fP
 155 will choose the correct whois server to query.
 156 Some exceptions, where
 157 \fBwhois\fP
 158 will not be able to handle data correctly, are detailed below.

 160 To obtain contact information about an
 161 administrator located in the Russian TLD domain RU,
 162 use the
 163 \fB\-c\fP
 164 option as shown in the following example, where
 165 \fICONTACT-ID\fP
 166 is substituted with the actual contact identifier.

 168 whois -c RU CONTACT-ID

 170 (Note: This example is specific to the TLD RU,
 171 but other TLDs can be queried by using a similar syntax.)

 173 The following example demonstrates how to query
 174 a whois server using a non-standard port, where
 175 ‘‘query-data’’
 176 is the query to be sent to
 177 ‘‘whois.example.com’’
 178 on port
 179 ‘‘rwhois’’
 180 (written numerically as 4321).

 182 whois -h whois.example.com -p rwhois query-data
 183 .SH SEE ALSO

 185 Vic White and Ken Harrenstien, \fINICNAME/WHOIS\fP, 1 March 1982, RFC 812.
 186 .SH HISTORY
 187 The
 188 \fBwhois\fP
 189 command appeared in
 190 4.3BSD.
```

```
***********************************************************
   15614 Mon Aug 26 06:56:02 2019
new/usr/src/man/man1m/beadm.1m
11622 clean up rarer mandoc lint warnings
***********************************************************
   1 '\" te
   2 .\" Copyright 2013 Nexenta Systems, Inc. All rights reserved.
   3 .\" Copyright 2016 Toomas Soome <tsoome@me.com>
   4 .TH BEADM 1M "Feb 21, 2016"
   5 .SH NAME
   6 beadm \- utility for managing zfs boot environments
   7 .SH SYNOPSIS
   8 .LP
   8 .nf
   9 \fBbeadm\fR \fBcreate\fR [\fB-a\fR] [\fB-d\fR \fIdescription\fR]
  10      [\fB-e\fR \fInon-activeBeName\fR | \fIbeName@snapshot\fR]
  11      [\fB-o\fR \fIproperty=value\fR] ... [\fB-p\fR \fIzpool\fR]
  12      [\fB-v\fR] \fIbeName\fR
  13 .fi

  15 .LP
  16 .nf
  17 \fBbeadm\fR \fBcreate\fR [\fB-v\fR] \fIbeName@snapshot\fR
  18 .fi

  20 .LP
  21 .nf
  22 \fBbeadm\fR \fBdestroy\fR [\fB-fFsv\fR] \fIbeName\fR | \fIbeName@snapshot\fR
  23 .fi

  25 .LP
  26 .nf
  27 \fBbeadm\fR \fBlist\fR [\fB-a\fR | \fB-ds\fR] [\fB-H\fR]
  28     [\fB-k\fR|\fB-K\fR \fBdate\fR | \fBname\fR | \fBspace\fR] [\fB-v\fR] [\fIbeN
  29 .fi

  31 .LP
  32 .nf
  33 \fBbeadm\fR \fBmount\fR [\fB-s\fR \fBro\fR|\fBrw\fR] [\fB-v\fR] \fIbeName\fR \fI
  34 .fi

  36 .LP
  37 .nf
  38 \fBbeadm\fR \fBunmount\fR [\fB-fv\fR] \fIbeName\fR | \fImountpoint\fR
  39 .fi

  41 .LP
  42 .nf
  43 \fBbeadm\fR \fBrename\fR [\fB-v\fR] \fIbeName\fR \fInewBeName\fR
  44 .fi

  46 .LP
  47 .nf
  48 \fBbeadm\fR \fBactivate\fR [\fB-v\fR] \fIbeName\fR
  49 .fi

  51 .LP
  52 .nf
  53 \fBbeadm\fR \fBrollback\fR [\fB-v\fR] \fIbeName\fR \fIsnapshot\fR
  54 .fi

  56 .LP
  57 .nf
  58 \fBbeadm\fR \fBrollback\fR [\fB-v\fR] \fIbeName@snapshot\fR
  59 .fi
```



```
  61 .SH DESCRIPTION
  62 The \fBbeadm\fR command is the user interface for managing zfs Boot
  63 Environments (BEs). This utility is intended to be used by System
  64 Administrators who want to manage multiple Solaris Instances on a single
  65 system.
  66 .sp
  67 The \fBbeadm\fR command supports the following operations:
  68 .RS +4
  69 .TP
  70 .ie t \(bu
  71 .el -
  72 Create a new BE, based on the active BE.
  73 .RE
  74 .RS +4
  75 .TP
  76 .ie t \(bu
  77 .el -
  78 Create a new BE, based on an inactive BE.
  79 .RE
  80 .RS +4
  81 .TP
  82 .ie t \(bu
  83 .el -
  84 Create a snapshot of an existing BE.
  85 .RE
  86 .RS +4
  87 .TP
  88 .ie t \(bu
  89 .el -
  90 Create a new BE, based on an existing snapshot.
  91 .RE
  92 .RS +4
  93 .TP
  94 .ie t \(bu
  95 .el -
  96 Create a new BE, and copy it to a different zpool.
  97 .RE
  98 .RS +4
  99 .TP
 100 .ie t \(bu
 101 .el -
 102 Activate an existing, inactive BE.
 103 .RE
 104 .RS +4
 105 .TP
 106 .ie t \(bu
 107 .el -
 108 Mount a BE.
 109 .RE
 110 .RS +4
 111 .TP
 112 .ie t \(bu
 113 .el -
 114 Unmount a BE.
 115 .RE
 116 .RS +4
 117 .TP
 118 .ie t \(bu
 119 .el -
 120 Destroy a BE.
 121 .RE
 122 .RS +4
 123 .TP
 124 .ie t \(bu
 125 .el -
 126 Destroy a snapshot of a BE.
```

```
.RE
.RS +4
.TP
.ie t \(bu
.el -
Rename an existing, inactive BE.
.RE
.RS +4
.TP
.ie t \(bu
.el -
Roll back a BE to an existing snapshot of a BE.
.RE
.RS +4
.TP
.ie t \(bu
.el -
Display information about your snapshots and datasets.
.RE

.SH SUBCOMMANDS
The \fBbeadm\fR command has the subcommands and options listed
below. Also see
EXAMPLES below.
.sp
.ne 2
.na
\fBbeadm\fR
.ad
.sp .6
.RS 4n
Displays command usage.
.RE

.sp
.ne 2
.na
\fBbeadm\fR \fBcreate\fR [\fB-a\fR] [\fB-d\fR \fIdescription\fR]
     [\fB-e\fR \fInon-activeBeName\fR | \fIbeName@snapshot\fR]
     [\fB-o\fR \fIproperty=value\fR] ... [\fB-p\fR \fIzpool\fR]
     [\fB-v\fR] \fIbeName\fR

.ad
.sp .6
.RS 4n
Creates a new boot environment named \fIbeName\fR.  If the \fB-e\fR option is
not
provided, the new boot environment will be created as a clone of the
currently
running boot environment. If the \fB-d\fR option is provided then the
description is
also used as the title for the BE’s entry in the boot menu. If the \fB-d\fR
option is not provided, \fIbeName\fR will be used as the title.
.sp
.ne 2
.na
\fB-a\fR
.ad
.sp .6
.RS 4n
Activate the newly created BE upon creation.  The default is to not activate
the newly created BE.
.RE
.sp
.ne 2
.na
```



```
\fB-d\fR \fIdescription\fR
.ad
.sp .6
.RS 4n
Create a new BE with a description associated with it.
.RE
.sp
.ne 2
.na
\fB-e\fR \fInon-activeBeName\fR
.ad
.sp .6
.RS 4n
Create a new BE from an existing inactive BE.
.RE
.sp
.ne 2
.na
\fB-e\fR \fIbeName@snapshot\fR
.ad
.sp .6
.RS 4n
Create a new BE from an existing snapshot of the BE named beName.
.RE
.sp
.ne 2
.na
\fB-o\fR \fIproperty=value\fR
.ad
.sp .6
.RS 4n
Create the datasets for new BE with specific ZFS properties.  Multiple
\fB-o\fR
options can be specified.  See \fBzfs\fR(1M) for more information on
the
\fB-o\fR option.
.RE
.sp
.ne 2
.na
\fB-p\fR \fIzpool\fR
.ad
.sp .6
.RS 4n
Create the new BE in the specified zpool.  If this is not provided, the
default
behavior is to create the new BE in the same pool as as the origin BE.
This option is not supported in non-global zone.
.RE
.sp
.ne 2
.na
\fB-v\fR
.ad
.sp .6
.RS 4n
Verbose mode. Displays verbose error messages from \fBbeadm\fR.
.RE
.RE

.sp
.ne 2
.na
\fBbeadm\fR \fBcreate\fR [\fB-v\fR] \fIbeName@snapshot\fR
.ad
.sp .6
```

```
.RS 4n
Creates a snapshot of the existing BE named beName.
.sp
.ne 2
.na
\fB-v\fR
.ad
.sp .6
.RS 4n
Verbose mode. Displays verbose error messages from \fBbeadm\fR.
.RE
.RE

.sp
.ne 2
.na
\fBbeadm\fR \fBdestroy\fR [\fB-fFsv\fR] \fIbeName\fR | \fIbeName@snapshot\fR
.ad
.sp .6
.RS 4n
Destroys the boot environment named \fIbeName\fR or destroys an existing
snapshot of
the boot environment named \fIbeName@snapshot\fR.  Destroying a
boot environment
will also destroy all snapshots of that boot environment.  Use
this command
with caution.
.sp
.ne 2
.na
\fB-f\fR
.ad
.sp .6
.RS 4n
Forcefully unmount the boot environment if it is currently mounted.
.RE
.sp
.ne 2
.na
\fB-F\fR
.ad
.sp .6
.RS 4n
Force the action without prompting to verify the destruction of the boot
environment.
.RE
.sp
.ne 2
.na
\fB-s\fR
.ad
.sp .6
.RS 4n
Destroy all snapshots of the boot
environment.
.RE
.sp
.ne 2
.na
\fB-v\fR
.ad
.sp .6
.RS 4n
Verbose mode. Displays verbose error messages from \fBbeadm\fR.
.RE
.RE
```



```
.sp
.ne 2
.na
\fBbeadm\fR \fBlist\fR [\fB-a\fR | \fB-ds\fR] [\fB-H\fR]
    [\fB-k\fR|\fB-K\fR \fBdate\fR | \fBname\fR | \fBspace\fR] [\fB-v\fR] [\fIbeN
.ad
.sp .6
.RS 4n
Lists information about the existing boot environment named \fIbeName\fR, or
lists
information for all boot environments if \fIbeName\fR is not provided.
The 'Active'
field indicates whether the boot environment is active now,
represented
by 'N'; active on reboot, represented by 'R'; or both, represented
by 'NR'. In non-global zone the 'Active' field also indicates whether the
boot environment has a non-active parent BE, represented by 'x'; is active
on boot in a non-active parent BE, represented by 'b'. Activate, rollback
and snapshot operations for boot environments from non-active global parent
BE aren't supported, destroy is allowed if these boot environments aren't
active on boot.
.sp
Each line in the machine parsable output has the boot environment name as the
first field.  The 'Space' field is displayed in bytes and the 'Created' field
is displayed in UTC format.  The \fB-H\fR option used with no other options
gives
the boot environment's uuid in the second field.  This field will be
blank if
the boot environment does not have a uuid. See the EXAMPLES section.
In non-global zones, this field shows the uuid of the parent BE.
.sp
.ne 2
.na
\fB-a\fR
.ad
.sp .6
.RS 4n
Lists all available information about the boot environment.  This includes
subordinate file systems and snapshots.
.RE
.sp
.ne 2
.na
\fB-d\fR
.ad
.sp .6
.RS 4n
Lists information about all subordinate file systems belonging to the boot
environment.
.RE
.sp
.ne 2
.na
\fB-s\fR
.ad
.sp .6
.RS 4n
Lists information about the snapshots of the boot environment.
.RE
.sp
.ne 2
.na
\fB-H\fR
.ad
.sp .6
```

```
.RS 4n
Do not list header information.  Each field in the list information is
separated by a semicolon.
.RE
.sp
.ne 2
.na
\fB-k\fR \fBdate\fR
.ad
.sp .6
.RS 4n
Sort BEs by date of creation in ascending order. This is default.
.RE
.sp
.ne 2
.na
\fB-k\fR \fBname\fR
.ad
.sp .6
.RS 4n
Lexical sort of BEs by name in ascending order.
.RE
.sp
.ne 2
.na
\fB-k\fR \fBspace\fR
.ad
.sp .6
.RS 4n
Sort BEs by space in ascending order.
.RE
.sp
.ne 2
.na
\fB-K date\fR | \fBname\fR | \fBspace\fR
.ad
.sp .6
.RS 4n
Same as the \fB-k\fR option, but sorts in descending order.
.RE
.sp
.ne 2
.na
\fB-v\fR
.ad
.sp .6
.RS 4n
Verbose mode. Displays verbose error messages from \fBbeadm\fR.
.RE
.RE

.sp
.ne 2
.na
\fBbeadm\fR \fBmount\fR [\fB-s\fR \fBro\fR|\fBrw\fR] [\fB-v\fR] \fIbeName\fR \fI
.ad
.sp .6
.RS 4n
Mounts a boot environment named beName at mountpoint.  mountpoint must be an
already existing empty directory.
.sp
.ne 2
.na
\fB-s\fR \fBro\fR|\fBrw\fR
.ad
.sp .6
```



```
.RS 4n
Mount the shared filesystems of the BE in read-only or read-write mode.
.RE
.sp
.ne 2
.na
\fB-v\fR
.ad
.sp .6
.RS 4n
Verbose mode. Displays verbose error messages from \fBbeadm\fR.
.RE
.RE

.sp
.ne 2
.na
\fBbeadm\fR \fBunmount\fR [\fB-fv\fR] \fIbeName\fR | \fImountpoint\fR
.ad
.sp .6
.RS 4n
Unmounts the boot environment named beName. The command can also be given a path
beName mount point on the system.
.sp
.ne 2
.na
\fB-f\fR
.ad
.sp .6
.RS 4n
Forcefully unmount the boot environment even if its currently busy.
.RE
.sp
.ne 2
.na
\fB-v\fR
.ad
.sp .6
.RS 4n
Verbose mode. Displays verbose error messages from \fBbeadm\fR.
.RE
.RE

.sp
.ne 2
.na
\fBbeadm\fR \fBrename\fR [\fB-v\fR] \fIbeName\fR \fInewBeName\fR
.ad
.sp .6
.RS 4n
Renames the boot environment named \fIbeName\fR to \fInewBeName\fR.
.sp
.ne 2
.na
\fB-v\fR
.ad
.sp .6
.RS 4n
Verbose mode. Displays verbose error messages from \fBbeadm\fR.
.RE
.RE

.sp
.ne 2
.na
\fBbeadm\fR \fBrollback\fR [\fB-v\fR] \fIbeName\fR \fIsnapshot\fR | \fIbeName@sn
```

```
.ad
.sp .6
.RS 4n
Roll back the boot environment named \fIbeName\fR to existing snapshot
of the boot environment named \fIbeName@snapshot\fR.
.sp
.ne 2
.na
\fB-v\fR
.ad
.sp .6
.RS 4n
Verbose mode. Displays verbose error messages from \fBbeadm\fR.
.RE
.RE

.sp
.ne 2
.na
\fBbeadm\fR \fBactivate\fR [\fB-v\fR] \fIbeName\fR
.ad
.sp .6
.RS 4n
Makes beName the active BE on next reboot.
.sp
.ne 2
.na
\fB-v\fR
.ad
.sp .6
.RS 4n
Verbose mode. Displays verbose error messages from \fBbeadm\fR.
.RE
.RE

.SH ALTERNATE BE LOCATION
.LP
The alternate BE location outside rpool/ROOT can be configured
by modifying the BENAME_STARTS_WITH parameter in /etc/default/be.
For example: BENAME_STARTS_WITH=rootfs

.SH EXAMPLES
.LP
\fBExample 1\fR: Create a new BE named BE1, by cloning the current live BE.
.sp
.in +2
.nf
\fB# beadm create BE1\fR
.fi
.in -2
.sp

.LP
\fBExample 2\fR: Create a new BE named BE2, by cloning the existing inactive
BE
named BE1.
.sp
.in +2
.nf
\fB# beadm create -e BE1 BE2\fR
.fi
.in -2
.sp

.LP
\fBExample 3\fR: Create a snapshot named now of the existing BE named BE1.
```



```
.sp
.in +2
.nf
\fB# beadm create BE1@now\fR
.fi
.in -2
.sp

.LP
\fBExample 4\fR: Create a new BE named BE3, by cloning an existing snapshot of
BE1.
.sp
.in +2
.nf
\fB# beadm create -e BE1@now BE3\fR
.fi
.in -2
.sp

.LP
\fBExample 5\fR: Create a new BE named BE4 based on the currently running BE.
Create the new BE in rpool2.
.sp
.in +2
.nf
\fB# beadm create -p rpool2 BE4\fR
.fi
.in -2
.sp

.LP
\fBExample 6\fR: Create a new BE named BE5 based on the currently running BE.
Create the new BE in rpool2, and create its datasets with compression turned
on.
.sp
.in +2
.nf
\fB# beadm create -p rpool2 -o compression=on BE5\fR
.fi
.in -2
.sp

.LP
\fBExample 7\fR: Create a new BE named BE6 based on the currently running BE
and provide a description for it.
.sp
.in +2
.nf
\fB# beadm create -d "BE6 used as test environment" BE6\fR
.fi
.in -2
.sp

.LP
\fBExample 8\fR: Activate an existing, inactive BE named BE3.
.sp
.in +2
.nf
\fB# beadm activate BE3\fR
.fi
.in -2
.sp

.LP
\fBExample 9\fR: Mount the BE named BE3 at /mnt.
.sp
```

```
653 .in +2
654 .nf
655 \fB# beadm mount BE3 /mnt\fR
656 .fi
657 .in -2
658 .sp

660 .LP
661 \fBExample 10\fR: Unmount the mounted BE named BE3.
662 .sp
663 .in +2
664 .nf
665 \fB# beadm unmount BE3\fR
666 .fi
667 .in -2
668 .sp

670 .LP
671 \fBExample 11\fR: Destroy the BE named BE3 without verification.
672 .sp
673 .in +2
674 .nf
675 \fB# beadm destroy -f BE3\fR
676 .fi
677 .in -2
678 .sp

680 .LP
681 \fBExample 12\fR: Destroy the snapshot named now of BE1.
682 .sp
683 .in +2
684 .nf
685 \fB# beadm destroy BE1@now\fR
686 .fi
687 .in -2
688 .sp

690 .LP
691 \fBExample 13\fR: Rename the existing, inactive BE named BE1 to BE3.
692 .sp
693 .in +2
694 .nf
695 \fB# beadm rename BE1 BE3\fR
696 .fi
697 .in -2
698 .sp

700 .LP
701 \fBExample 14\fR: Roll back the BE named BE1 to snapshot BE1@now.
702 .sp
703 .in +2
704 .nf
705 \fB# beadm rollback BE1 BE1@now\fR
706 .fi
707 .in -2
708 .sp

710 .LP
711 \fBExample 15\fR: List all existing boot environments.

713 .sp
714 .in +2
715 .nf
716 \fB# beadm list\fR
717 BE  Active Mountpoint Space  Policy Created
718 --  ------ ---------- -----  ------ -------
```



```
719 BE2 -      -          72.0K  static 2008-05-21 12:26
720 BE3 -      -          332.0K static 2008-08-26 10:28
721 BE4 -      -          15.78M static 2008-09-05 18:20
722 BE5 NR     /          7.25G  static 2008-09-09 16:53
723 .fi
724 .in -2
725 .sp

727 .LP
728 \fBExample 16\fR: List all existing boot environmets and list all dataset and
729 snapshot information about those bootenvironments.

731 .sp
732 .in +2
733 .nf
734 \fB# beadm list -d -s\fR

736 BE/Dataset/Snapshot     Active Mountpoint Space   Policy Created
737 --------------------     ------ ---------- -----   ------ -------
738 BE2
739    p/ROOT/BE2           -      -          36.0K   static 2008-05-21 12:26
740    p/ROOT/BE2/opt       -      -          18.0K   static 2008-05-21 16:26
741    p/ROOT/BE2/opt@now   -      -          0       static 2008-09-08 22:43
742    p/ROOT/BE2@now       -      -          0       static 2008-09-08 22:43
743 BE3
744    p/ROOT/BE3           -      -          192.0K  static 2008-08-26 10:28
745    p/ROOT/BE3/opt       -      -          86.0K   static 2008-08-26 10:28
746    p/ROOT/BE3/opt/local -      -          36.0K   static 2008-08-28 10:58
747 BE4
748    p/ROOT/BE4           -      -          15.78M  static 2008-09-05 18:20
749 BE5
750    p/ROOT/BE5           NR     /          6.10G   static 2008-09-09 16:53
751    p/ROOT/BE5/opt       -      /opt       24.55M  static 2008-09-09 16:53
752    p/ROOT/BE5/opt@bar   -      -          18.38M  static 2008-09-10 00:59
753    p/ROOT/BE5/opt@foo   -      -          18.38M  static 2008-06-10 16:37
754    p/ROOT/BE5@bar       -      -          139.44M static 2008-09-10 00:59
755    p/ROOT/BE5@foo       -      -          912.85M static 2008-06-10 16:37
756 .fi
757 .in -2
758 .sp

760 \fBExample 17\fR: List all dataset and snapshot information about BE5

762 .sp
763 .in +2
764 .nf
765 \fB# beadm list -a BE5\fR

767 BE/Dataset/Snapshot   Active Mountpoint Space   Policy Created
768 --------------------   ------ ---------- -----   ------ -------
769 BE5
770    p/ROOT/BE5         NR     /          6.10G   static 2008-09-09 16:53
771    p/ROOT/BE5/opt     -      /opt       24.55M  static 2008-09-09 16:53
772    p/ROOT/BE5/opt@bar -      -          18.38M  static 2008-09-10 00:59
773    p/ROOT/BE5/opt@foo -      -          18.38M  static 2008-06-10 16:37
774    p/ROOT/BE5@bar     -      -          139.44M static 2008-09-10 00:59
775    p/ROOT/BE5@foo     -      -          912.85M static 2008-06-10 16:37
776 .fi
777 .in -2
778 .sp

780 .LP
781 \fBExample 18\fR: List machine parsable information about all boot
782 environments.

784 .sp
```

```
.in +2
.nf
\fB# beadm list -H\fR

BE2;;;;55296;static;1211397974
BE3;;;;339968;static;1219771706
BE4;;;;16541696;static;1220664051
BE5;215b8387-4968-627c-d2d0-f4a011414bab;NR;/;7786206208;static;1221004384
.fi
.in -2
.sp

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

.sp
.ne 2
.na
\fB>0\fR
.ad
.sp .6
.RS 4n
Failure
.RE


.SH FILES
.LP
.sp
.ne 2
.na
\fB/var/log/beadm/<beName>/create.log.<yyyymmdd_hhmmss>\fR
.ad
.sp .6
.RS 4n
Log used for capturing beadm create output
.sp
.nf
\fIyyyymmdd_hhmmss\fR - 20071130_140558
\fIyy\fR - year; 2007
\fImm\fR - month; 11
\fIdd\fR - day; 30
\fIhh\fR - hour; 14
\fImm\fR - minute; 05
\fIss\fR - second; 58
.fi
.in -2
.sp
.RE
.sp
.LP
.sp
.ne 2
.na
\fB/etc/default/be\fR
.ad
```



```
.sp .6
.RS 4n
Contains default value for BENAME_STARTS_WITH parameter
.sp
.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     Uncommitted
.TE


.SH SEE ALSO
.LP
.BR zfs (1M)
```

```
**********************************************************
   19599 Mon Aug 26 06:56:02 2019
new/usr/src/man/man1m/cryptoadm.1m
11622 clean up rarer mandoc lint warnings
**********************************************************
'\" te
.\" 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 limitati
.\" fields enclosed by brackets "[]" replaced with your own identifying informat
.TH CRYPTOADM 1M "Sep 1, 2009"
.SH NAME
cryptoadm \- cryptographic framework administration
.SH SYNOPSIS
.LP
.nf
\fBcryptoadm\fR list [\fB-mpv\fR] [provider=\fIprovider-name\fR]
     [mechanism=\fImechanism-list\fR]
.fi

.LP
.nf
\fBcryptoadm\fR disable
     provider=\fIprovider-name\fR mechanism=\fImechanism-list\fR | random | all
.fi

.LP
.nf
\fBcryptoadm\fR enable
     provider=\fIprovider-name\fR mechanism=\fImechanism-list\fR | random | all
.fi

.LP
.nf
\fBcryptoadm\fR install provider=\fIprovider-name\fR
.fi

.LP
.nf
\fBcryptoadm\fR install provider=\fIprovider-name\fR
     [mechanism=\fImechanism-list\fR]
.fi

.LP
.nf
\fBcryptoadm\fR uninstall provider=\fIprovider-name\fR
.fi

.LP
.nf
\fBcryptoadm\fR unload provider=\fIprovider-name\fR
.fi

.LP
.nf
\fBcryptoadm\fR disable fips-140
.fi

.LP
.nf
\fBcryptoadm\fR enable fips-140
.fi

.LP
.nf
\fBcryptoadm\fR list fips-140
```



```
.fi

.LP
.nf
\fBcryptoadm\fR refresh
.fi

.LP
.nf
\fBcryptoadm\fR start
.fi

.LP
.nf
\fBcryptoadm\fR stop
.fi

.LP
.nf
\fBcryptoadm\fR \fB-\fR\fB-help\fR
.fi

.SH DESCRIPTION
.sp
.LP
The \fBcryptoadm\fR utility displays cryptographic provider information for a
system, configures the mechanism policy for each provider, and installs or
uninstalls a cryptographic provider. The cryptographic framework supports three
types of providers: a user-level provider (a PKCS11 shared library), a kernel
software provider (a loadable kernel software module), and a kernel hardware
provider (a cryptographic hardware device).
.sp
.LP
For kernel software providers, the \fBcryptoadm\fR utility provides the
\fBunload\fR subcommand. This subcommand instructs the kernel to unload a
kernel software providers.
.sp
.LP
For the cryptographic framework's metaslot, the \fBcryptoadm\fR utility
provides subcommands to enable and disable the metaslot's features, list
metaslot's configuration, specify alternate persistent object storage, and
configure the metaslot's mechanism policy.
.sp
.LP
The \fBcryptoadm\fR utility provides subcommands to enable and disable FIPS-140
mode in the Cryptographic Framework. It also provides a \fBlist\fR subcommand
to display the current status of FIPS-140 mode.
.sp
.LP
Administrators will find it useful to use \fBsyslog\fR facilities (see
\fBsyslogd\fR(1M) and \fBlogadm\fR(1M)) to maintain the cryptographic
subsystem. Logging can be especially useful under the following circumstances:
.RS +4
.TP
.ie t \(bu
.el o
If kernel-level daemon is dead, all applications fail. You can learn this from
syslog and use \fBsvcadm\fR(1M) to restart the \fBsvc:/system/cryptosvc\fR
service.
.RE
.RS +4
.TP
.ie t \(bu
.el o
If there are bad providers plugged into the framework, you can learn this from
syslog and remove the bad providers from the framework.
```

```
125 .RE
126 .sp
127 .LP
128 With the exception of the subcommands or options listed below, the
129 \fBcryptoadm\fR command needs to be run by a privileged user.
130 .RS +4
131 .TP
132 .ie t \(bu
133 .el o
134 subcommand \fBlist\fR, any options
135 .RE
136 .RS +4
137 .TP
138 .ie t \(bu
139 .el o
140 subcommand \fB-\fR\fB-help\fR
141 .RE
142 .SH OPTIONS
146 .sp
147 .LP
143 The \fBcryptoadm\fR utility has the various combinations of subcommands and
144 options shown below.
145 .sp
146 .ne 2
147 .na
148 \fB\fBcryptoadm\fR \fBlist\fR\fR
149 .ad
150 .sp .6
151 .RS 4n
152 Display the list of installed providers.
153 .RE

155 .sp
156 .ne 2
157 .na
158 \fB\fBcryptoadm\fR \fBlist metaslot\fR\fR
159 .ad
160 .sp .6
161 .RS 4n
162 Display the system-wide configuration for metaslot.
163 .RE

165 .sp
166 .ne 2
167 .na
168 \fB\fBcryptoadm\fR \fBlist\fR \fB-m\fR \fB[ provider=\fIprovider-name\fR |
169 metaslot ]\fR\fR
170 .ad
171 .sp .6
172 .RS 4n
173 Display a list of mechanisms that can be used with the installed providers or
174 metaslot. If a provider is specified, display the name of the specified
175 provider and the mechanism list that can be used with that provider. If the
176 metaslot keyword is specified, display the list of mechanisms that can be used
177 with metaslot.
178 .RE

180 .sp
181 .ne 2
182 .na
183 \fB\fBcryptoadm\fR \fBlist\fR \fB-p\fR \fB[ provider=\fIprovider-name\fR |
184 metaslot ]\fR\fR
185 .ad
186 .sp .6
187 .RS 4n
188 Display the mechanism policy (that is, which mechanisms are available and which
```



```
189 are not) for the installed providers. Also display the provider feature policy
190 or metaslot. If a provider is specified, display the name of the provider with
191 the mechanism policy enforced on it only. If the metaslot keyword is specified,
192 display the mechanism policy enforced on the metaslot.
193 .RE

195 .sp
196 .ne 2
197 .na
198 \fB\fBcryptoadm\fR \fBlist\fR \fB-v\fR \fBprovider=\fIprovider-name\fR |
199 metaslot\fR\fR
200 .ad
201 .sp .6
202 .RS 4n
203 Display details about the specified provider if a provider is specified. If the
204 metaslot keyword is specified, display details about the metaslot.
205 .RE

207 .sp
208 .ne 2
209 .na
210 \fB\fB-v\fR\fR
211 .ad
212 .sp .6
213 .RS 4n
214 For the various \fBlist\fR subcommands described above (except for \fBlist\fR
215 \fB-p\fR), the \fB-v\fR (verbose) option provides details about providers,
216 mechanisms and slots.
217 .RE

219 .sp
220 .ne 2
221 .na
222 \fB\fBcryptoadm\fR \fBdisable provider=\fIprovider-name\fR\fR\fR
223 .ad
224 .br
225 .na
226 \fB[ mechanism=\fImechanism-list\fR | \fIprovider-feature\fR \fB\&... |\fR
227 \fBall\fR ]\fR
228 .ad
229 .sp .6
230 .RS 4n
231 Disable the mechanisms or provider features specified for the provider. See
232 OPERANDS for a description of \fImechanism\fR, \fIprovider-feature\fR, and the
233 \fBall\fR keyword.
234 .RE

236 .sp
237 .ne 2
238 .na
239 \fB\fBcryptoadm\fR \fB[ mechanism=\fImechanism-list\fR ] [ auto-key-migrate
240 ]\fR\fR
241 .ad
242 .sp .6
243 .RS 4n
244 Disable the metaslot feature in the cryptographic framework or disable some of
245 metaslot's features. If no operand is specified, this command disables the
246 metaslot feature in the cryptographic framework. If a list of mechanisms is
247 specified, disable mechanisms specified for metaslot. If all mechanisms are
248 disabled for metaslot, the metaslot will be disabled. See OPERANDS for a
249 description of mechanism. If the \fBauto-key-migrate\fR keyword is specified,
250 it disables the migration of sensitive token objects to other slots even if it
251 is necessary for performing crypto operations. See OPERANDS for a description
252 of \fBauto-key-migrate\fR.
253 .RE
```

```
.sp
.ne 2
.na
\fB\fBcryptoadm\fR \fBenable provider=\fIprovider-name\fR\fR\fR
.ad
.br
.na
\fB[ mechanism=\fImechanism-list\fR | \fIprovider-feature\fR \fB\&... |\fR
\fBall\fR ]\fR
.ad
.sp .6
.RS 4n
Enable the mechanisms or provider features specified for the provider. See
OPERANDS for a description of \fImechanism\fR, \fIprovider-feature\fR, and the
\fBall\fR keyword.
.RE

.sp
.ne 2
.na
\fB\fBcryptoadm\fR \fBenable metaslot [ mechanism=\fImechanism-list\fR ]
|\fR\fR
.ad
.br
.na
\fB\fB[ [ token=\fItoken-label\fR] [ slot=\fIslot-description\fR] |\fR\fR
.ad
.br
.na
\fB\fBdefault-keystore ] | [ auto-key-migrate ]\fR\fR
.ad
.sp .6
.RS 4n
If no operand is specified, this command enables the metaslot feature in the
cryptographic framework. If a list of mechanisms is specified, it enables only
the list of specified mechanisms for metaslot. If \fItoken-label\fR is
specified, the specified token will be used as the persistent object store. If
the \fIslot-description\fR is specified, the specified slot will be used as the
persistent object store. If both the \fItoken-label\fR and the
\fIslot-description\fR are specified, the provider with the matching token
label and slot description is used as the persistent object store. If the
\fBdefault-keystore\fR keyword is specified, metaslot will use the default
persistent object store. If the \fBauto-key-migrate\fR keyword is specified,
sensitive token objects will automatically migrate to other slots as needed to
complete certain crypto operations. See OPERANDS for a description of
mechanism, token, slot, \fBdefault-keystore\fR, and \fBauto-key-migrate\fR.
.RE

.sp
.ne 2
.na
\fB\fBcryptoadm\fR \fBinstall provider=\fIprovider-name\fR\fR\fR
.ad
.sp .6
.RS 4n
Install a user-level provider into the system. The \fIprovider\fR operand must
be an absolute pathname of the corresponding shared library. If there are both
32-bit and 64-bit versions for a library, this command should be run once only
with the path name containing \fB$ISA\fR. Note that \fB$ISA\fR is not a
reference to an environment variable. Note also that \fB$ISA\fR must be quoted
(with single quotes [for example, \fB\&'$ISA'\fR]) or the \fB$\fR must be
escaped to keep it from being incorrectly expanded by the shell. The user-level
framework expands \fB$ISA\fR to an empty string or an architecture-specific
directory, for example, \fBsparcv9\fR.
.sp
The preferred way of installing a user-level provider is to build a package for
```



```
the provider. For more information, see the \fISolaris Security for Developer's
Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fBcryptoadm\fR \fBinstall provider=\fIprovider-name\fR\fR\fR
.ad
.br
.na
\fBmechanism=\fImechanism-list\fR\fR
.ad
.sp .6
.RS 4n
Install a kernel software provider into the system. The provider should contain
the base name only. The \fImechanism-list\fR operand specifies the complete
list of mechanisms to be supported by this provider.
.sp
The preferred way of installing a kernel software provider is to build a
package for providers. For more information, see the \fISolaris Security for
Developer's Guide\fR.
.RE

.sp
.ne 2
.na
\fB\fBcryptoadm\fR \fBuninstall provider=\fIprovider-name\fR\fR\fR
.ad
.sp .6
.RS 4n
Uninstall the specified \fIprovider\fR and the associated mechanism policy from
the system. This subcommand applies only to a user-level provider or a kernel
software provider.
.RE

.sp
.ne 2
.na
\fB\fBcryptoadm\fR \fBunload provider=\fIprovider-name\fR\fR\fR
.ad
.sp .6
.RS 4n
Unload the kernel software module specified by \fIprovider\fR.
.RE

.sp
.ne 2
.na
\fB\fBcryptoadm\fR \fBdisable fips-140\fR\fR
.ad
.sp .6
.RS 4n
Disable FIPS-140 mode in the Cryptographic Framework.
.RE

.sp
.ne 2
.na
\fB\fBcryptoadm\fR \fBenable fips-140\fR\fR
.ad
.sp .6
.RS 4n
Enable FIPS-140 mode in the Cryptographic Framework. This subcommand does not
disable the non-FIPS approved algorithms from the user-level
\fBpkcs11_softtoken\fR library and the kernel software providers. It is the
```

```
387 consumers of the framework that are responsible for using only FIPS-approved
388 algorithms.
389 .sp
390 Upon completion of this subcommand, a message is issued to inform the
391 administrator that any plugins added that are not within the boundary might
392 invalidate FIPS compliance and to check the Security Policies for those
393 plugins. In addition, a warning message is issued to indicate that, in this
394 release, the Cryptographic Framework has not been FIPS 140-2 certified.
395 .sp
396 The system will require a reboot to perform Power-Up Self Tests that include a
397 cryptographic algorithm test and a software integrity test.
398 .RE

400 .sp
401 .ne 2
402 .na
403 \fB\fBcryptoadm\fR \fBlist fips-140\fR\fR
404 .ad
405 .sp .6
406 .RS 4n
407 Display the current setting of FIPS-140 mode in the Cryptographic Framework.
408 The status of FIPS-140 mode is \fBenabled\fR or \fBdisabled\fR. The default
409 FIPS-140 mode is \fBdisabled\fR.
410 .RE

412 .sp
413 .ne 2
414 .na
415 \fB\fBcryptoadm\fR \fBrefresh\fR\fR
416 .ad
417 .br
418 .na
419 \fB\fBcryptoadm\fR \fBstart\fR\fR
420 .ad
421 .br
422 .na
423 \fB\fBcryptoadm\fR \fBstop\fR\fR
424 .ad
425 .sp .6
426 .RS 4n
427 Private interfaces for use by \fBsmf\fR(5), these must not be used directly.
428 .RE

430 .sp
431 .ne 2
432 .na
433 \fB\fBcryptoadm\fR \fB-help\fR\fR
434 .ad
435 .sp .6
436 .RS 4n
437 Display the command usage.
438 .RE

440 .SH OPERANDS
446 .sp
441 .ne 2
442 .na
443 \fBprovider=\fIprovider-name\fR\fR
444 .ad
445 .sp .6
446 .RS 4n
447 A user-level provider (a PKCS11 shared library), a kernel software provider (a
448 loadable kernel software module), or a kernel hardware provider (a
449 cryptographic hardware device).
450 .sp
451 A valid value of the \fIprovider\fR operand is one entry from the output of a
```



```
452 command of the form: \fBcryptoadm\fR \fIlist\fR. A \fIprovider\fR operand for a
453 user-level provider is an absolute pathname of the corresponding shared
454 library. A \fIprovider\fR operand for a kernel software provider contains a
455 base name only. A \fIprovider\fR operand for a kernel hardware provider is in a
456 "\fIname\fR/\fInumber\fR" form.
457 .RE

459 .sp
460 .ne 2
461 .na
462 \fBmechanism=\fImechanism-list\fR\fR
463 .ad
464 .sp .6
465 .RS 4n
466 A comma separated list of one or more PKCS #11 mechanisms. A process for
467 implementing a cryptographic operation as defined in PKCS #11 specification.
468 You can substitute \fBall\fR for \fImechanism-list\fR, to specify all
469 mechanisms on a provider. See the discussion of the \fBall\fR keyword, below.
470 .RE

472 .sp
473 .ne 2
474 .na
475 \fB\fIprovider-feature\fR\fR
476 .ad
477 .sp .6
478 .RS 4n
479 A cryptographic framework feature for the given provider. Currently only
480 \fBrandom\fR is accepted as a feature. For a user-level provider, disabling the
481 random feature makes the PKCS #11 routines \fBC_GenerateRandom\fR and
482 \fBC_SeedRandom\fR unavailable from the provider. For a kernel provider,
483 disabling the random feature prevents \fB/dev/random\fR from gathering random
484 numbers from the provider.
485 .RE

487 .sp
488 .ne 2
489 .na
490 \fB\fBall\fR\fR
491 .ad
492 .sp .6
493 .RS 4n
494 The keyword all can be used with with the \fBdisable\fR and \fBenable\fR
495 subcommands to operate on all provider features.
496 .RE

498 .sp
499 .ne 2
500 .na
501 \fB\fBtoken=\fR\fItoken-label\fR\fR
502 .ad
503 .sp .6
504 .RS 4n
505 The label of a token in one of the providers in the cryptographic framework.
506 .sp
507 A valid value of the token operand is an item displayed under "Token Label"
508 from the output of the command \fBcryptoadm list\fR \fB-v\fR.
509 .RE

511 .sp
512 .ne 2
513 .na
514 \fB\fBslot=\fR\fIslot-description\fR\fR
515 .ad
516 .sp .6
517 .RS 4n
```

```
518 The description of a slot in one of the providers in the cryptographic
519 framework.
520 .sp
521 A valid value of the slot operand is an item displayed under "Description" from
522 the output of the command \fBcryptoadm list\fR \fB-v\fR.
523 .RE

525 .sp
526 .ne 2
527 .na
528 \fB\fBdefault-keystore\fR\fR
529 .ad
530 .sp .6
531 .RS 4n
532 The keyword \fBdefault-keystore\fR is valid only for metaslot. Specify this
533 keyword to set the persistent object store for metaslot back to using the
534 default store.
535 .RE

537 .sp
538 .ne 2
539 .na
540 \fB\fBauto-key-migrate\fR\fR
541 .ad
542 .sp .6
543 .RS 4n
544 The keyword auto-key-migrate is valid only for metaslot. Specify this keyword
545 to configure whether metaslot is allowed to move sensitive token objects from
546 the token object slot to other slots for performing cryptographic operations.
547 .RE

549 .sp
550 .LP
551 The keyword \fBall\fR can be used in two ways with the \fBdisable\fR and
552 \fBenable\fR subcommands:
553 .RS +4
554 .TP
555 .ie t \(bu
556 .el o
557 You can substitute \fBall\fR for \fBmechanism\fR=\fImechanism-list\fR, as in:
558 .sp
559 .in +2
560 .nf
561 # \fBcryptoadm enable provider=dca/0 all\fR
562 .fi
563 .in -2
564 .sp

566 This command enables the mechanisms on the provider \fBand\fR any other
567 provider-features, such as \fBrandom\fR.
568 .sp
569 .in +2
570 .nf
571 # \fBcryptoadm enable provider=des mechanism=all\fR
572 .fi
573 .in -2
574 .sp

576 .RE
577 .RS +4
578 .TP
579 .ie t \(bu
580 .el o
581 You can also use \fBall\fR as an argument to \fBmechanism\fR, as in:
582 .sp
583 .in +2
```



```
584 .nf
585 # \fBcryptoadm enable provider=des mechanism=all\fR
586 .fi
587 .in -2
588 .sp

590 \&...which enables all mechanisms on the provider, but enables no other
591 provider-features, such as \fBrandom\fR.
592 .RE
593 .SH EXAMPLES
600 .LP
594 \fBExample 1 \fRDisplay List of Providers Installed in System
595 .sp
596 .LP
597 The following command displays a list of all installed providers:

599 .sp
600 .in +2
601 .nf
602 example% \fBcryptoadm list\fR
603 user-level providers:
604 /usr/lib/security/$ISA/pkcs11_kernel.so
605 /usr/lib/security/$ISA/pkcs11_softtoken.so
606 /opt/lib/libcryptoki.so.1
607 /opt/SUNWconn/lib/$ISA/libpkcs11.so.1

609 kernel software providers:
610     des
611     aes
612     bfish
613     sha1
614     md5

616 kernel hardware providers:
617     dca/0
618 .fi
619 .in -2
620 .sp

622 .LP
623 \fBExample 2 \fRDisplay Mechanism List for \fBmd5\fR Provider
624 .sp
625 .LP
626 The following command is a variation of the \fBlist\fR subcommand:

628 .sp
629 .in +2
630 .nf
631 example% \fBcryptoadm list -m provider=md5\fR
632 md5: CKM_MD5,CKM_MD5_HMAC,CKM_MD5_HMAC_GENERAL
633 .fi
634 .in -2
635 .sp

637 .LP
638 \fBExample 3 \fRDisable Specific Mechanisms for Kernel Software Provider
639 .sp
640 .LP
641 The following command disables mechanisms \fBCKM_DES3_ECB\fR and
642 \fBCKM_DES3_CBC\fR for the kernel software provider \fBdes\fR:

644 .sp
645 .in +2
646 .nf
647 example# \fBcryptoadm disable provider=des\fR
648 .fi
```

```
649 .in -2
650 .sp

652 .LP
653 \fBExample 4 \fRDisplay Mechanism Policy for a Provider
654 .sp
655 .LP
656 The following command displays the mechanism policy for the \fBdes\fR provider:

658 .sp
659 .in +2
660 .nf
661 example% \fBcryptoadm list -p provider=des\fR
662 des: All mechanisms are enabled, except CKM_DES3_ECB, CKM_DES3_CBC
663 .fi
664 .in -2
665 .sp

667 .LP
668 \fBExample 5 \fREnable Specific Mechanism for a Provider
669 .sp
670 .LP
671 The following command enables the \fBCKM_DES3_ECB\fR mechanism for the kernel
672 software provider \fBdes\fR:

674 .sp
675 .in +2
676 .nf
677 example# \fBcryptoadm enable provider=des mechanism=CKM_DES3_ECB\fR
678 .fi
679 .in -2
680 .sp

682 .LP
683 \fBExample 6 \fRInstall User-Level Provider
684 .sp
685 .LP
686 The following command installs a user-level provider:

688 .sp
689 .in +2
690 .nf
691 example# \fBcryptoadm install provider=/opt/lib/libcryptoki.so.1\fR
692 .fi
693 .in -2
694 .sp

696 .LP
697 \fBExample 7 \fRInstall User-Level Provider That Contains 32- and 64-bit
698 Versions
699 .sp
700 .LP
701 The following command installs a user-level provider that contains both 32-bit
702 and 64-bit versions:

704 .sp
705 .in +2
706 .nf
707 example# \fBcryptoadm install \e\fR
708 provider=/opt/SUNWconn/lib/'$ISA'/libpkcs11.so.1
709 .fi
710 .in -2
711 .sp

713 .LP
714 \fBExample 8 \fRUninstall a Provider
```



```
715 .sp
716 .LP
717 The following command uninstalls the \fBmd5\fR provider:

719 .sp
720 .in +2
721 .nf
722 example# \fBcryptoadm uninstall provider=md5\fR
723 .fi
724 .in -2
725 .sp

727 .LP
728 \fBExample 9 \fRDisable metaslot
729 .sp
730 .LP
731 The following command disables the metaslot feature in the cryptographic
732 framework.

734 .sp
735 .in +2
736 .nf
737 example# \fBcryptoadm disable metaslot\fR
738 .fi
739 .in -2
740 .sp

742 .LP
743 \fBExample 10 \fRSpecify metaslot to Use Specified Token as Persistent Object
744 Store
745 .sp
746 .LP
747 The following command specifies that metaslot use the Venus token as the
748 persistent object store.

750 .sp
751 .in +2
752 .nf
753 example# \fBcryptoadm enable metaslot token="SUNW,venus"\fR
754 .fi
755 .in -2
756 .sp

758 .SH EXIT STATUS
766 .sp
767 .LP
759 The following exit values are returned:
760 .sp
761 .ne 2
762 .na
763 \fB\fB0\fR\fR
764 .ad
765 .sp .6
766 .RS 4n
767 Successful completion.
768 .RE

770 .sp
771 .ne 2
772 .na
773 \fB\fB>0\fR\fR
774 .ad
775 .sp .6
776 .RS 4n
777 An error occurred.
778 .RE
```

```
780 .SH ATTRIBUTES
790 .sp
791 .LP
781 See \fBattributes\fR(5) for descriptions of the following attributes:
782 .sp

784 .sp
785 .TS
786 box;
787 c | c
788 l | l .
789 ATTRIBUTE TYPE  ATTRIBUTE VALUE
790 _
791 Interface Stability     See below
792 .TE

794 .sp
795 .LP
796 The \fBstart\fR, \fBstop\fR, and \fBrefresh\fR options are Private interfaces.
797 All other options are Evolving. The utility name is Stable.
798 .SH SEE ALSO
810 .sp
811 .LP
799 \fBlogadm\fR(1M), \fBsvcadm\fR(1M), \fBsyslogd\fR(1M), \fBlibpkcs11\fR(3LIB),
800 \fBexec_attr\fR(4), \fBprof_attr\fR(4), \fBattributes\fR(5), \fBsmf\fR(5),
801 \fBrandom\fR(7D)
815 .sp
816 .LP

803 .sp
804 .LP
805 \fISolaris Security for Developer's Guide\fR
806 .SH NOTES
822 .sp
823 .LP
807 If a hardware provider's policy was made explicitly (that is, some of its
808 mechanisms were disabled) and the hardware provider has been detached, the
809 policy of this hardware provider is still listed.
810 .sp
811 .LP
812 \fBcryptoadm\fR assumes that, minimally, a 32-bit shared object is delivered
813 for each user-level provider. If both a 32-bit and 64-bit shared object are
814 delivered, the two versions must provide the same functionality. The same
815 mechanism policy applies to both.
```

```
**********************************************************
   31301 Mon Aug 26 06:56:03 2019
new/usr/src/man/man1m/monitor.1m
11622 clean up rarer mandoc lint warnings
**********************************************************
```

```
   1 '\" te
   2 .\" Copyright (c) 2003, Sun Microsystems, Inc.
   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 MONITOR 1M "Jul 24, 2003"
   7 .SH NAME
   8 monitor \- SPARC system PROM monitor
   9 .SH SYNOPSIS
  10 .LP
  10 .nf
  11 \fBSTOP\(miA\fR
  12 .fi

  14 .LP
  15 .nf
  16 \fBBREAK\fR
  17 .fi

  19 .LP
  20 .nf
  21 \fBinitial system power-on\fR
  22 .fi

  24 .LP
  25 .nf
  26 \fBexit from a client program, e.g., the Operating System\fR
  27 .fi

  29 .SH DESCRIPTION
  31 .sp
  32 .LP
  30 The \fBCPU\fR board of a workstation contains one or more \fBEPROMs\fR or
  31 \fBEEPROMs.\fR The program which executes from the \fBPROMs\fR is referred to
  32 as "the monitor". Among other things, the monitor performs system
  33 initialization at power-on and provides a user interface.
  34 .SS "Monitor Prompt"
  38 .sp
  39 .LP
  35 The monitor of earlier workstations was known as the \fBSunMON\fR monitor and
  36 displayed the \fB>\fR for its prompt. See the \fBSunMON MONITOR USAGE\fR
  37 section for further details.
  38 .sp
  39 .LP
  40 Existing workstations use a monitor which is known as the OpenBoot monitor.
  41 The OpenBoot monitor typically displays \fBok\fR as its prompt, but it may also
  42 display the \fB>\fR prompt under certain circumstances.
  43 .sp
  44 .LP
  45 If the \fB\&'auto-boot?'\fR \fBNVRAM\fR parameter is set to \fB\&'false'\fR
  46 when the workstation is powered on, the system does not attempt to boot and the
  47 monitor issues its prompt. If \fB\&'auto-boot'\fR is set to \fB\&'true'\fR, the
  48 system initiates the boot sequence. The boot sequence can be aborted by
  49 simultaneously pressing two keys on the system's keyboard: \fBL1\fR and \fBA\fR
  50 (on older keyboards), or \fBStop\fR and \fBA\fR (on newer keyboards).  Either a
  51 lower case \fBa\fR or an upper case \fBA\fR works for the keyboard abort
  52 sequence.  If a console has been attached by way of one of the system's serial
  53 ports then the abort sequence can be accomplished by sending a \fBBREAK\fR. See
  54 \fBtip\fR(1).
  55 .sp
  56 .LP
```



```
  57 When the \fBNVRAM\fR \fB\&'security-mode'\fR parameter has been turned on, or
  58 when the value of the \fB\&'sunmon-compat?'\fR parameter is true, then the
  59 OpenBoot monitor displays the message: \fBType b (boot), c (continue), or n
  60 (new command mode)\fR
  61 .sp
  62 .LP
  63 and the \fB>\fR prompt appears.
  64 .SH OPENBOOT PROM USAGE
  70 .sp
  71 .LP
  65 Some of the more useful commands that can be issued from OpenBoot's \fBok \fR
  66 prompt are described here.  Refer to the   book for a complete list of
  67 commands.
  68 .SS "Help"
  76 .sp
  77 .LP
  69 Help for various functional areas of the OpenBoot monitor can be obtained by
  70 typing \fBhelp\fR. The help listing provides a number of other key words which
  71 can then be used in the help command to provide further details.
  72 .SS "NVRAM Parameters"
  82 .sp
  83 .LP
  73 Each workstation contains one or more \fBNVRAM\fR devices which contains unique
  74 system ID information, as well as a set of user-configurable parameters.  The
  75 \fBNVRAM\fR parameters allow the user a certain level of flexibility in
  76 configuring the system to act in a given manner under a specific set of
  77 circumstances.
  78 .sp
  79 .LP
  80 See \fBeeprom\fR(1M) for a description of the parameters and information
  81 regarding setting the parameters from the OS level.
  82 .sp
  83 .LP
  84 The following commands can be used at the OpenBoot monitor to access the
  85 \fBNVRAM\fR parameters.
  86 .sp
  87 .ne 2
  88 .na
  89 \fB\fBprintenv\fR\fR
  90 .ad
  91 .RS 18n
  92 Used to list the \fBNVRAM\fR parameters, along with their default values and
  93 current values.
  94 .RE

  96 .sp
  97 .ne 2
  98 .na
  99 \fB\fBsetenv\fR\fI pn pv\fR\fR
 100 .ad
 101 .RS 18n
 102 Used to set or modify a parameter.  The \fIpn\fR represents the parameter name,
 103 and \fIpv\fR represents the parameter value.
 104 .RE

 106 .sp
 107 .ne 2
 108 .na
 109 \fB\fBset-default\fR \fIpn\fR\fR
 110 .ad
 111 .RS 18n
 112 Used to set an individual parameter back to its default value.
 113 .RE

 115 .sp
 116 .ne 2
```

```
117 .na
118 \fB\fBset-defaults\fR\fR
119 .ad
120 .RS 18n
121 Used to reset all parameters to their default values.  (Note that
122 \fB\&'set-defaults'\fR only affects parameters that have assigned default
123 values.)
124 .RE

126 .SS "Security Parameters"
138 .sp
139 .LP
127 Newer OpenBoot monitors contain user interfaces that support the storage and
128 listing of keys for later use by client programs.
129 .sp
130 .ne 2
131 .na
132 \fB\fBlist-security-keys\fR\fR
133 .ad
134 .sp .6
135 .RS 4n
136 Lists the names of keys currently stored on a machine.
137 .RE

139 .sp
140 .ne 2
141 .na
142 \fB\fBset-security-key\fR \fIkeyname\fR [ \fIkeydata\fR ]\fR
143 .ad
144 .sp .6
145 .RS 4n
146 Stores key data \fIkeydata\fR in a key named \fIkeyname\fR. Actual key data can
147 be up to 32 bytes in length. The maximum length of \fIkeyname\fR is 64 bytes,
148 which allows for the hex-formatted ASCII used to present the key data. If
149 \fIkeydata\fR is not present, \fIkeyname\fR and its corresponding data is
150 deleted.
151 .RE

153 .SS "Hardware Checks and Diagnostics"
167 .sp
168 .LP
154 The following commands are available for testing or checking the system's
155 hardware.  If the \fB\&'diag-switch?'\fR \fBNVRAM\fR parameter is set to true
156 when the system is powered on, then a Power-On Self Test (POST) diagnostic is
157 run, if present, sending its results messages to the system's serial port A.
158 Not all of the commands shown are available on all workstations.
159 .sp
160 .ne 2
161 .na
162 \fB\fBtest-all\fR\fR
163 .ad
164 .RS 17n
165 Run the diagnostic tests on each device which has provided a self-test.
166 .RE

168 .sp
169 .ne 2
170 .na
171 \fB\fBtest\fR \fBfloppy\fR\fR
172 .ad
173 .RS 17n
174 Run diagnostics on the system's floppy device.
175 .RE

177 .sp
178 .ne 2
```



```
179 .na
180 \fB\fBtest\fR \fB/memory\fR\fR
181 .ad
182 .RS 17n
183 Run the main memory tests.  If the \fBNVRAM\fR parameter \fB\&'diag-switch?'\fR
184 is set to true, then all of main memory is tested.  If the parameter is false
185 then only the amount of memory specified in the \fB\&'selftest-#megs'\fR
186 \fBNVRAM\fR parameter is tested.
187 .RE

189 .sp
190 .ne 2
191 .na
192 \fB\fBtest\fR \fBnet\fR\fR
193 .ad
194 .RS 17n
195 Test the network connection for the on-board network controller.
196 .RE

198 .sp
199 .ne 2
200 .na
201 \fB\fBwatch-net\fR\fR
202 .ad
203 .RS 17n
204 Monitor the network attached to the on-board net controller.
205 .RE

207 .sp
208 .ne 2
209 .na
210 \fB\fBwatch-net-all\fR\fR
211 .ad
212 .RS 17n
213 Monitor the network attached to the on-board net controller, as well as the
214 network controllers installed in SBus slots.
215 .RE

217 .sp
218 .ne 2
219 .na
220 \fB\fBwatch-clock\fR\fR
221 .ad
222 .RS 17n
223 Test the system's clock function.
224 .RE

226 .SS "System Information"
242 .sp
243 .LP
227 The following commands are available for displaying information about the
228 system.  Not all commands are available on all workstations.
229 .sp
230 .ne 2
231 .na
232 \fB\fBbanner\fR\fR
233 .ad
234 .RS 18n
235 Display the power-on banner.
236 .RE

238 .sp
239 .ne 2
240 .na
241 \fB\fB\&.enet-addr\fR\fR
242 .ad
```

```
243 .RS 18n
244 Display the system’s Ethernet address.
245 .RE

247 .sp
248 .ne 2
249 .na
250 \fB\fB\&.idprom\fR\fR
251 .ad
252 .RS 18n
253 Display the formatted contents of the \fBIDPROM.\fR
254 .RE

256 .sp
257 .ne 2
258 .na
259 \fB\fBmodule-info\fR\fR
260 .ad
261 .RS 18n
262 Display information about the system’s processor(s).
263 .RE

265 .sp
266 .ne 2
267 .na
268 \fB\fBprobe-scsi\fR\fR
269 .ad
270 .RS 18n
271 Identify the devices attached to the on-board \fBSCSI\fR controller.
272 .RE

274 .sp
275 .ne 2
276 .na
277 \fB\fBprobe-scsi-all\fR\fR
278 .ad
279 .RS 18n
280 Identify the devices attached to the on-board \fBSCSI\fR controller as well as
281 those devices which are attached to SBus \fBSCSI\fR controllers.
282 .RE

284 .sp
285 .ne 2
286 .na
287 \fB\fBshow-disks\fR\fR
288 .ad
289 .RS 18n
290 Display a list of the device paths for installed \fBSCSI\fR disk controllers.
291 .RE

293 .sp
294 .ne 2
295 .na
296 \fB\fBshow-displays\fR\fR
297 .ad
298 .RS 18n
299 Display a list of the device paths for installed display devices.
300 .RE

302 .sp
303 .ne 2
304 .na
305 \fB\fBshow-nets\fR\fR
306 .ad
307 .RS 18n
308 Display a list of the device paths for installed Ethernet controllers.
```



```
309 .RE

311 .sp
312 .ne 2
313 .na
314 \fB\fBshow-sbus\fR\fR
315 .ad
316 .RS 18n
317 Display list of installed SBus devices.
318 .RE

320 .sp
321 .ne 2
322 .na
323 \fB\fBshow-tapes\fR\fR
324 .ad
325 .RS 18n
326 Display a list of the device paths for installed \fBSCSI\fR tape controllers.
327 .RE

329 .sp
330 .ne 2
331 .na
332 \fB\fBshow-ttys\fR\fR
333 .ad
334 .RS 18n
335 Display a list of the device paths for tty devices.
336 .RE

338 .sp
339 .ne 2
340 .na
341 \fB\fB\&.traps\fR\fR
342 .ad
343 .RS 18n
344 Display a list of the SPARC trap types.
345 .RE

347 .sp
348 .ne 2
349 .na
350 \fB\fB\&.version\fR\fR
351 .ad
352 .RS 18n
353 Display the version and date of the OpenBoot PROM.
354 .RE

356 .SS "Emergency Commands"
374 .sp
375 .LP
357 These commands must be typed from the keyboard, they do not work from a console
358 which is attached by way of the serial ports.  With the exception of the
359 \fBStop-A\fR command, these commands are issued by pressing and holding down
360 the indicated keys on the keyboard immediately after the system has been
361 powered on.  The keys must be held down until the monitor has checked their
362 status.  The \fBStop-A\fR command can be issued at any time after the console
363 display begins, and the keys do not need to be held down once they’ve been
364 pressed. The \fBStop-D,\fR \fBStop-F\fR and \fBStop-N\fR commands are not
365 allowed when one of the security modes has been set.  Not all commands are
366 available on all workstations.
367 .sp
368 .ne 2
369 .na
370 \fB\fBStop (L1)\fR\fR
371 .ad
372 .RS 17n
```

```
373 Bypass the Power-On Self Test (POST).  This is only effective if the system has
374 been placed into the diagnostic mode.
375 .RE

377 .sp
378 .ne 2
379 .na
380 \fB\fBStop-A (L1-A)\fR\fR
381 .ad
382 .RS 17n
383 Abort the current operation and return to the monitor's default prompt.
384 .RE

386 .sp
387 .ne 2
388 .na
389 \fB\fBStop-D (L1-D)\fR\fR
390 .ad
391 .RS 17n
392 Set the system's \fB\&'diag-switch?'\fR \fBNVRAM\fR parameter to
393 \fB\&'true'\fR, which places the system in diagnostic mode.  POST diagnostics,
394 if present, are run, and the messages are displayed by way of the system's
395 serial port A.
396 .RE

398 .sp
399 .ne 2
400 .na
401 \fB\fBStop-F (L1-F)\fR\fR
402 .ad
403 .RS 17n
404 Enter the OpenBoot monitor before the monitor has probed the system for
405 devices.  Issue the \fB\&'fexit'\fR command to continue with system
406 initialization.
407 .RE

409 .sp
410 .ne 2
411 .na
412 \fB\fBStop-N (L1-N)\fR\fR
413 .ad
414 .RS 17n
415 Causes the \fBNVRAM\fR parameters to be reset to their default values.  Note
416 that not all parameters have default values.
417 .RE

419 .SS "Line Editor Commands"
439 .sp
440 .LP
420 The following commands can be used while the monitor is displaying the \fBok\fR
421 prompt.  Not all of these editing commands are available on all workstations.
422 .sp
423 .ne 2
424 .na
425 \fB\fBCTRL-A\fR\fR
426 .ad
427 .RS 10n
428 Place the cursor at the start of line.
429 .RE

431 .sp
432 .ne 2
433 .na
434 \fB\fBCTRL-B\fR\fR
435 .ad
436 .RS 10n
```



```
437 Move the cursor backward one character.
438 .RE

440 .sp
441 .ne 2
442 .na
443 \fB\fBESC-B\fR\fR
444 .ad
445 .RS 10n
446 Move the cursor backward one word.
447 .RE

449 .sp
450 .ne 2
451 .na
452 \fB\fBCTRL-D\fR\fR
453 .ad
454 .RS 10n
455 Erase the character that the cursor is currently highlighting.
456 .RE

458 .sp
459 .ne 2
460 .na
461 \fB\fBESC-D\fR\fR
462 .ad
463 .RS 10n
464 Erase the portion of word from the cursor's present position to the end of the
465 word.
466 .RE

468 .sp
469 .ne 2
470 .na
471 \fB\fBCTRL-E\fR\fR
472 .ad
473 .RS 10n
474 Place the cursor at the end of line.
475 .RE

477 .sp
478 .ne 2
479 .na
480 \fB\fBCTRL-F\fR\fR
481 .ad
482 .RS 10n
483 Move the cursor forward one character.
484 .RE

486 .sp
487 .ne 2
488 .na
489 \fB\fBESC-F\fR\fR
490 .ad
491 .RS 10n
492 Move the cursor forward one word.
493 .RE

495 .sp
496 .ne 2
497 .na
498 \fB\fBCTRL-H\fR\fR
499 .ad
500 .RS 10n
501 Erase the character preceding the cursor (also use Delete or Back Space)
502 .RE
```

```
.sp
.ne 2
.na
\fB\fBESC-H\fR\fR
.ad
.RS 10n
Erase the portion of the word which precedes the cursor (use also \fBCTRL-W)\fR
.RE

.sp
.ne 2
.na
\fB\fBCTRL-K\fR\fR
.ad
.RS 10n
Erase from the cursor's present position to the end of the line.
.RE

.sp
.ne 2
.na
\fB\fBCTRL-L\fR\fR
.ad
.RS 10n
Show the command history list.
.RE

.sp
.ne 2
.na
\fB\fBCTRL-N\fR\fR
.ad
.RS 10n
Recall the next command from the command history list
.RE

.sp
.ne 2
.na
\fB\fBCTRL-P\fR\fR
.ad
.RS 10n
Recall a previous command from the command history list.
.RE

.sp
.ne 2
.na
\fB\fBCTRL-Q\fR\fR
.ad
.RS 10n
Quote the next character (used to type a control character).
.RE

.sp
.ne 2
.na
\fB\fBCTRL-R\fR\fR
.ad
.RS 10n
Retype the current line.
.RE

.sp
.ne 2
```



```
.na
\fB\fBCTRL-U\fR\fR
.ad
.RS 10n
Erase from the cursor's present position to the beginning of the line.
.RE

.sp
.ne 2
.na
\fB\fBCTRL-Y\fR\fR
.ad
.RS 10n
Insert the contents of the memory buffer into the line, in front (to the left)
of the cursor.
.RE

.SS "nvramrc"
.sp
.LP
The \fBnvramrc\fR is an area of the system's \fBNVRAM\fR where users may store
Forth programs. The programs which are stored in the \fBnvramrc\fR are executed
each time the system is reset, provided that the \fB\&'use-nvramrc?'\fR
\fBNVRAM\fR parameter has been set to \fB\&'true'\fR. Refer to the book for
information on how to edit and use the \fBnvramrc\fR.
.SS "Restricted Monitor"
.sp
.LP
The command \fB\&'old-mode'\fR is used to move OpenBoot into a restricted
monitor mode, causing the \fB> \fR prompt to be displayed.  Only three commands
are allowed while in the restricted monitor; the \fB\&'go'\fR command (to
resume a program which was interrupted with the \fBStop-A\fR command), the
\fB\&'n'\fR command (to return to the normal OpenBoot monitor), and boot
commands.  The restricted monitor's boot commands approximate the older
\fBSunMON\fR monitor's boot command syntax. If a \fB\&'security-mode'\fR has
been turned on then the restricted monitor becomes the default monitor
environment.  The restricted monitor may also become the default environment if
the \fB\&'sunmon-compat?' \fR \fBNVRAM\fR parameter is set to true.  Not all
workstations have the \fB\&'sunmon-compat?'\fR parameter.
.SH SUNMON PROM USAGE
.sp
.LP
The following commands are available systems with older  \fBSunMON\fR-based
PROM:
.sp
.ne 2
.na
\fB\fB+\fR|\fB\(mi\fR\fR
.ad
.sp .6
.RS 4n
Increment or decrement the current address and display the contents of the new
location.
.RE

.sp
.ne 2
.na
\fB\fB^C\fR\fI source destination n\fR\fR
.ad
.sp .6
.RS 4n
(caret-C) Copy, byte-by-byte, a block of length \fIn\fR from the \fBsource\fR
address to the \fIdestination\fR address.
.RE
```

```
.sp
.ne 2
.na
\fB\fB^I \fR\fIprogram\fR\fR
.ad
.sp .6
.RS 4n
(caret-I) Display the compilation date and location of \fIprogram\fR.
.RE

.sp
.ne 2
.na
\fB\fB^T \fR\fIvirtual_address\fR\fR
.ad
.sp .6
.RS 4n
(caret-T) Display the physical address to which \fIvirtual_address\fR is
mapped.
.RE

.sp
.ne 2
.na
\fB\fBb\fR [ \fB!\fR ] [ \fIdevice\fR [
\fB(\fR\fIc\fR\fB,\fR\fIu\fR\fB,\fR\fIp\|\fR\fB)\fR\fI\fR ] ] [ \fIpathname\fR
] [ \fIarguments_list\fR ]\fR
.ad
.sp .6
.RS 4n

.RE

.sp
.ne 2
.na
\fB\fBb\fR[\fB?\fR]\fR
.ad
.sp .6
.RS 4n
Reset appropriate parts of the system and bootstrap a program.  A ‘\fB!\fR’
(preceding the \fIdevice\fR argument) prevents the system reset from occurring.
Programs can be loaded from various devices (such as a disk, tape, or
Ethernet). ‘\fBb\fR’ with no arguments causes a default boot, either from a
disk, or from an Ethernet controller. ‘\fBb?\fR’ displays all boot devices and
their \fIdevice\fRs.
.sp
.ne 2
.na
\fB\fIdevice\fR\fR
.ad
.RS 18n
one of
.sp
.ne 2
.na
\fB\fBle\fR\fR
.ad
.RS 6n
Lance Ethernet
.RE

.sp
.ne 2
.na
\fB\fBie\fR\fR
```



```
.ad
.RS 6n
Intel Ethernet
.RE

.sp
.ne 2
.na
\fB\fBsd\fR\fR
.ad
.RS 6n
\fBSCSI\fR disk, CDROM
.RE

.sp
.ne 2
.na
\fB\fBst\fR\fR
.ad
.RS 6n
\fBSCSI\fR 1/4" or 1/2" tape
.RE

.sp
.ne 2
.na
\fB\fBfd\fR\fR
.ad
.RS 6n
Diskette
.RE

.sp
.ne 2
.na
\fB\fBid\fR\fR
.ad
.RS 6n
IPI disk
.RE

.sp
.ne 2
.na
\fB\fBmt\fR\fR
.ad
.RS 6n
Tape Master 9-track 1/2" tape
.RE

.sp
.ne 2
.na
\fB\fBxd\fR\fR
.ad
.RS 6n
Xylogics 7053 disk
.RE

.sp
.ne 2
.na
\fB\fBxt\fR\fR
.ad
.RS 6n
Xylogics 1/2" tape
```

```
761 .RE

763 .sp
764 .ne 2
765 .na
766 \fB\fBxy\fR\fR
767 .ad
768 .RS 6n
769 Xylogics 440/450 disk
770 .RE

772 .RE

774 .sp
775 .ne 2
776 .na
777 \fB\fIc\fR\fR
778 .ad
779 .RS 18n
780 A controller number (\fB0\fR if only one controller),
781 .RE

783 .sp
784 .ne 2
785 .na
786 \fB\fIu\fR\fR
787 .ad
788 .RS 18n
789 A unit number (\fB0\fR if only one driver), and
790 .RE

792 .sp
793 .ne 2
794 .na
795 \fB\fIp\fR\fR
796 .ad
797 .RS 18n
798 A partition.
799 .RE

801 .sp
802 .ne 2
803 .na
804 \fB\fIpathname\fR\fR
805 .ad
806 .RS 18n
807 A pathname for a program such as \fB/stand/diag\fR.
808 .RE

810 .sp
811 .ne 2
812 .na
813 \fB\fIarguments_list\fR\fR
814 .ad
815 .RS 18n
816 A list of up to seven arguments to pass to the program being booted.
817 .RE

819 .RE

821 .sp
822 .ne 2
823 .na
824 \fB\fBc [\fR\fIvirtual_address\fR\fB]\fR\fR
825 .ad
826 .sp .6
```



```
827 .RS 4n
828 Resume execution of a program. When given, \fIvirtual_address\fR is the address
829 at which execution resumes. The default is the current \fBPC.\fR Registers are
830 restored to the values shown by the \fBd\fR, and \fBr\fR commands.
831 .RE

833 .sp
834 .ne 2
835 .na
836 \fB\fBd [\fR\fIwindow_number\fR\fB]\fR\fR
837 .ad
838 .sp .6
839 .RS 4n
840 Display (dump) the state of the processor. The processor state is observable
841 only after:
842 .RS +4
843 .TP
844 .ie t \(bu
845 .el o
846 An unexpected trap was encountered.
847 .RE
848 .RS +4
849 .TP
850 .ie t \(bu
851 .el o
852 A user program dropped into the monitor (by calling \fIabortent\fR).
853 .RE
854 .RS +4
855 .TP
856 .ie t \(bu
857 .el o
858 The user manually entered the monitor by typing \fBL1\(miA\fR or \fBBREAK.\fR
859 .RE
860 The display consists of the following:
861 .RS +4
862 .TP
863 .ie t \(bu
864 .el o
865 The special registers: \fBPSR,\fR \fBPC,\fR nPC, \fBTBR,\fR \fBWIM,\fR and Y
866 .RE
867 .RS +4
868 .TP
869 .ie t \(bu
870 .el o
871 Eight global registers
872 .RE
873 .RS +4
874 .TP
875 .ie t \(bu
876 .el o
877 24 window registers (8 \fIin\fR, 8 \fIlocal\fR, and 8 \fIout\fR), corresponding
878 to one of the 7 available windows.  If a Floating-Point Unit is on board, its
879 status register along with 32 floating-point registers are also shown.
880 .RE
881 .sp
882 .ne 2
883 .na
884 \fB\fIwindow_number\fR\fR
885 .ad
886 .RS 17n
887 Display the indicated \fIwindow_number\fR, which can be any value between
888 \fB0\fR and \fB6\fR, inclusive.  If no window is specified and the \fBPSR's\fR
889 current window pointer contains a valid window number, registers from the
890 window that was active just prior to entry into the monitor are displayed.
891 Otherwise, registers from window 0 are displayed.
892 .RE
```

```
 894 .RE

 896 .sp
 897 .ne 2
 898 .na
 899 \fB\fBe\fR [\fIvirtual_address\fR]\|[\fIaction\fR] .\|.\|.\fR
 900 .ad
 901 .sp .6
 902 .RS 4n
 903 Open the 16-bit word at \fIvirtual_address\fR (default zero).  The address is
 904 interpreted in the address space defined by the \fBs\fR command. See the
 905 \fBa\fR command for a description of \fIaction\fR.
 906 .RE

 908 .sp
 909 .ne 2
 910 .na
 911 \fB\fBf\fR\fI virtual_address1 virtual_address2 pattern \fR [\fIsize\|\fR]\fR
 912 .ad
 913 .sp .6
 914 .RS 4n
 915 Fill the bytes, words, or long words from \fIvirtual_address1\fR (lower) to
 916 \fIvirtual_address2\fR (higher) with the constant, \fIpattern\fR. The
 917 \fBsize\fR argument can take one of the following values:
 918 .sp
 919 .ne 2
 920 .na
 921 \fB\fBb\fR\fR
 922 .ad
 923 .RS 5n
 924 byte format (the default)
 925 .RE

 927 .sp
 928 .ne 2
 929 .na
 930 \fB\fBw\fR\fR
 931 .ad
 932 .RS 5n
 933 word format
 934 .RE

 936 .sp
 937 .ne 2
 938 .na
 939 \fB\fBl\fR\fR
 940 .ad
 941 .RS 5n
 942 long word format
 943 .RE

 945 For example, the following command fills the address block from \fB0x1000\fR to
 946 \fB0x2000\fR with the word pattern, \fB0xABCD\fR:
 947 .sp
 948 \fBf 1000 2000 ABCD W\fR
 949 .RE

 951 .sp
 952 .ne 2
 953 .na
 954 \fB\fBg\fR [\fIvector\|\fR] [\fIargument\|\fR]\fR
 955 .ad
 956 .br
 957 .na
 958 \fB\fBg\fR [\fIvirtual_address\|\fR] [\fIargument\fR\|]\fR
```



```
 959 .ad
 960 .sp .6
 961 .RS 4n
 962 Goto (jump to) a predetermined or default routine (first form), or to a
 963 user-specified routine (second form).  The value of \fIargument\fR is passed to
 964 the routine.  If the \fIvector\fR or \fIvirtual_address\fR argument is omitted,
 965 the value in the \fBPC\fR is used as the address to jump to.
 966 .sp
 967 To set up a predetermined routine to jump to, a user program must, prior to
 968 executing the monitor's \fBg\fR command, set the variable
 969 \fB*romp->v_vector_cmd\fR to be equal to the virtual address of the desired
 970 routine. Predetermined routines need not necessarily return control to the
 971 monitor.
 972 .sp
 973 The default routine, defined by the monitor, prints the user-supplied
 974 \fIvector\fR according to the format supplied in \fIargument\fR. This format
 975 can be one of:
 976 .sp
 977 .ne 2
 978 .na
 979 \fB\fB%x\fR\fR
 980 .ad
 981 .RS 6n
 982 hexadecimal
 983 .RE

 985 .sp
 986 .ne 2
 987 .na
 988 \fB\fB%d\fR\fR
 989 .ad
 990 .RS 6n
 991 decimal
 992 .RE

 994 .RE

 996 .sp
 997 .ne 2
 998 .na
 999 \fB\fBg0\fR\fR
1000 .ad
1001 .sp .6
1002 .RS 4n
1003 Force a panic and produce a crash dump  when the monitor is running as a result
1004 of the system being interrupted,
1005 .RE

1007 .sp
1008 .ne 2
1009 .na
1010 \fB\fBg4\fR\fR
1011 .ad
1012 .sp .6
1013 .RS 4n
1014 (Sun-4 systems only) Force a kernel stack trace  when the monitor is running as
1015 a result of the system being interrupted,
1016 .RE

1018 .sp
1019 .ne 2
1020 .na
1021 \fB\fBh\fR\fR
1022 .ad
1023 .sp .6
1024 .RS 4n
```

Display the help menu for monitor commands and their descriptions. To return to the monitor's basic command level, press \fBESCAPE\fR or \fBq\fR before pressing \fBRETURN.\fR
.RE

.sp
.ne 2
.na
\fB\fBi \fR[\fIcache_data_offset\fR\|] [\fIaction\fR\|]\|.\|.\|.\fR
.ad
.sp .6
.RS 4n
Modify cache data \fBRAM\fR command. Display and/or modify one or more of the cache data addresses. See the \fBa\fR command for a description of \fIaction\fR.
.RE

.sp
.ne 2
.na
\fB\fBj\fR [\fIcache_tag_offset\fR\|] [\fIaction\|\fR]\|.\|.\|.\fR
.ad
.sp .6
.RS 4n
Modify cache tag \fBRAM\fR command. Display and/or modify the contents of one or more of the cache tag addresses. See the \fBa\fR command for a description of \fIaction\fR.
.RE

.sp
.ne 2
.na
\fB\fBk [\fR\fIreset_level\fR\fB]\fR\fR
.ad
.sp .6
.RS 4n
Reset the system, where \fIreset_level\fR is:
.sp
.ne 2
.na
\fB\fB0\fR\fR
.ad
.RS 5n
Reset \fBVMEbus,\fR interrupt registers, video monitor (Sun-4 systems). This is the default.
.RE

.sp
.ne 2
.na
\fB\fB1\fR\fR
.ad
.RS 5n
Software reset.
.RE

.sp
.ne 2
.na
\fB\fB2\fR\fR
.ad
.RS 5n
Power-on reset. Resets and clears the memory. Runs the \fBEPROM-based\fR diagnostic self test, which can take several minutes, depending upon how much memory is being tested.
.RE

.RE

.sp
.ne 2
.na
\fB\fBkb\fR\fR
.ad
.sp .6
.RS 4n
Display the system banner.
.RE

.sp
.ne 2
.na
\fB\fBl\|\fR [\fIvirtual_address\fR\|]\|[\fIaction\fR]\|.\|.\|.\fR
.ad
.sp .6
.RS 4n
Open the long word (32 bit) at memory address \fIvirtual_address\fR (default zero). The address is interpreted in the address space defined by the \fBs\fR command (below). See the \fBa\fR command for a description of \fIaction\fR.
.RE

.sp
.ne 2
.na
\fB\fBm\|\fR [\fIvirtual_address\fR\|]\|[\fIaction\fR\|]\|.\|.\|.\fR
.ad
.sp .6
.RS 4n
Open the segment map entry that maps \fIvirtual_address\fR (default zero). The address is interpreted in the address space defined by the \fBs\fR command. See the \fBa\fR command for a description of \fIaction\fR.
.RE

.sp
.ne 2
.na
\fB\fBne\fR\fR
.ad
.sp .6
.RS 4n

.RE

.sp
.ne 2
.na
\fB\fBni\fR\fR
.ad
.sp .6
.RS 4n
Disable, enable, or invalidate the cache, respectively.
.RE

.sp
.ne 2
.na
\fB\fBo\|\fR [\fIvirtual_address\fR\|]\|[action\fI]\|.\|.\|.\fR\fR
.ad
.sp .6
.RS 4n
Open the byte location specified by \fIvirtual_address\fR (default zero). The address is interpreted in the address space defined by the \fBs\fR command. See

```
1157 the \fBa\fR command for a description of \fIaction\fR.
1158 .RE

1160 .sp
1161 .ne 2
1162 .na
1163 \fB\fBp\|\fR [\fIvirtual_address\fR\|]\|[\fIaction\fR].\|.\|.\fR
1164 .ad
1165 .sp .6
1166 .RS 4n
1167 Open the page map entry that maps \fIvirtual_address\fR (default zero) in the
1168 address space defined by the \fBs\fR command. See the \fBa\fR command for a
1169 description of \fIaction\fR.
1170 .RE

1172 .sp
1173 .ne 2
1174 .na
1175 \fB\fBq\|\fR [\fIeeprom_offset\fR\|]\|[\fIaction\fR\|].\|.\|.\fR
1176 .ad
1177 .sp .6
1178 .RS 4n
1179 Open the \fBEEPROM\fR \fIeeprom_offset\fR (default zero) in the \fBEEPROM\fR
1180 address space. All addresses are referenced from the beginning or base of the
1181 \fBEEPROM\fR in physical address space, and a limit check is performed to
1182 insure that no address beyond the \fBEEPROM\fR physical space is accessed. This
1183 command is used to display or modify configuration parameters, such as: the
1184 amount of memory to test during self test, whether to display a standard or
1185 custom banner, if a serial port (A or B) is to be the system console, etc. See
1186 the \fBa\fR command for a description of \fIaction\fR.
1187 .RE

1189 .sp
1190 .ne 2
1191 .na
1192 \fB\fBr\|\fR [\fIregister_number\|\fR]\fR
1193 .ad
1194 .br
1195 .na
1196 \fB\fBr\|\fR [\fIregister_type\|\fR]\fR
1197 .ad
1198 .br
1199 .na
1200 \fB\fBr\|\fR [\fIw window_number\|\fR]\fR
1201 .ad
1202 .sp .6
1203 .RS 4n
1204 Display and/or modify one or more of the \fBIU\fR or \fBFPU\fR registers. A
1205 hexadecimal \fIregister_number\fR can be one of:
1206 .sp
1207 .ne 2
1208 .na
1209 \fB\fB0x00\fR\(mi\fB0x0f\fR\fR
1210 .ad
1211 .RS 16n
1212 window(0,i0)\(miwindow(0,i7), window(0,i0)\(emwindow(0,i7)
1213 .RE

1215 .sp
1216 .ne 2
1217 .na
1218 \fB\fB0x16\fR\(mi\fB0x1f\fR\fR
1219 .ad
1220 .RS 16n
1221 window(1,i0)\(miwindow(1,i7), window(1,i0)\(emwindow(1,i7)
1222 .RE
```



```
1224 .sp
1225 .ne 2
1226 .na
1227 \fB\fB0x20\fR\(mi\fB0x2f\fR\fR
1228 .ad
1229 .RS 16n
1230 window(2,i0)\(miwindow(2,i7), window(2,i0)\(emwindow(2,i7)
1231 .RE

1233 .sp
1234 .ne 2
1235 .na
1236 \fB\fB0x30\fR\(mi\fB0x3f\fR\fR
1237 .ad
1238 .RS 16n
1239 window(3,i0)\(miwindow(3,i7), window(3,i0)\(emwindow(3,i7)
1240 .RE

1242 .sp
1243 .ne 2
1244 .na
1245 \fB\fB0x40\fR\(mi\fB0x4f\fR\fR
1246 .ad
1247 .RS 16n
1248 window(4,i0)\(miwindow(4,i7), window(4,i0)\(emwindow(4,i7)
1249 .RE

1251 .sp
1252 .ne 2
1253 .na
1254 \fB\fB0x50\fR\(mi\fB0x5f\fR\fR
1255 .ad
1256 .RS 16n
1257 window(5,i0)\(miwindow(5,i7), window(5,i0)\(emwindow(5,i7)
1258 .RE

1260 .sp
1261 .ne 2
1262 .na
1263 \fB\fB0x60\fR\(mi\fB0x6f\fR\fR
1264 .ad
1265 .RS 16n
1266 window(6,i0)\(miwindow(6,i7), window(6,i0)\(emwindow(6,i7)
1267 .RE

1269 .sp
1270 .ne 2
1271 .na
1272 \fB\fB0x70\fR\(mi\fB0x77\fR\fR
1273 .ad
1274 .RS 16n
1275 \fBg0, g1, g2, g3, g4, g5, g6, g7\fR
1276 .RE

1278 .sp
1279 .ne 2
1280 .na
1281 \fB\fB0x78\fR\(mi\fB0x7d\fR\fR
1282 .ad
1283 .RS 16n
1284 \fBPSR,\fR \fBPC,\fR \fBnPC,\fR \fBWIM,\fR \fBTBR,\fR \fBY.\fR
1285 .RE

1287 .sp
1288 .ne 2
```

```
.na
\fB\fB0x7e\fR\(mi\fB0x9e\fR\fR
.ad
.RS 16n
\fBFSR,\fR f0\(mif31
.RE

Register numbers can only be displayed after an unexpected trap, a user program
has entered the monitor using the \fIabortent\fR function, or the user has
entered the monitor by manually typing \fBL1\(miA\fR or \fBBREAK.\fR
.sp
If a \fIregister_type\fR is given, the first register of the indicated type is
displayed. \fIregister_type\fR can be one of:
.sp
.ne 2
.na
\fB\fBf\fR\fR
.ad
.RS 5n
floating-point
.RE

.sp
.ne 2
.na
\fB\fBg\fR\fR
.ad
.RS 5n
global
.RE

.sp
.ne 2
.na
\fB\fBs\fR\fR
.ad
.RS 5n
special
.RE

If \fBw\fR and a \fIwindow_number\fR (\fB0\fR\(em\fB6\fR) are given, the first
\fIin\fR-register within the indicated window is displayed.  If
\fIwindow_number\fR is omitted, the window that was active just prior to
entering the monitor is used.  If the \fBPSR's\fR current window pointer is
invalid, window 0 is used.
.RE

.sp
.ne 2
.na
\fB\fBs [\fR\fIasi\fR\fB])\fR\fR
.ad
.sp .6
.RS 4n
Set or display the Address Space Identifier.  With no argument, \fBs\fR
displays the current Address Space Identifier.  The \fIasi\fR value can be one
of:
.sp
.ne 2
.na
\fB\fB0x2\fR\fR
.ad
.RS 7n
control space
.RE
```



```
.sp
.ne 2
.na
\fB\fB0x3\fR\fR
.ad
.RS 7n
segment table
.RE

.sp
.ne 2
.na
\fB\fB0x4\fR\fR
.ad
.RS 7n
Page table
.RE

.sp
.ne 2
.na
\fB\fB0x8\fR\fR
.ad
.RS 7n
user instruction
.RE

.sp
.ne 2
.na
\fB\fB0x9\fR\fR
.ad
.RS 7n
supervisor instruction
.RE

.sp
.ne 2
.na
\fB\fB0xa\fR\fR
.ad
.RS 7n
user data
.RE

.sp
.ne 2
.na
\fB\fB0xb\fR\fR
.ad
.RS 7n
supervisor data
.RE

.sp
.ne 2
.na
\fB\fB0xc\fR\fR
.ad
.RS 7n
flush segment
.RE

.sp
.ne 2
.na
```

```
\fB\fB0xd\fR\fR
.ad
.RS 7n
flush page
.RE

.sp
.ne 2
.na
\fB\fB0xe\fR\fR
.ad
.RS 7n
flush context
.RE

.sp
.ne 2
.na
\fB\fB0xf\fR\fR
.ad
.RS 7n
cache data
.RE

.RE

.sp
.ne 2
.na
\fB\fBu\fR [ \fBecho\fR ]\fR
.ad
.sp .6
.RS 4n

.RE

.sp
.ne 2
.na
\fB\fBu\fR [ \fIport\fR ] [ \fIoptions\fR ] [ \fIbaud_rate\fR ]\fR
.ad
.sp .6
.RS 4n

.RE

.sp
.ne 2
.na
\fB\fBu\fR [ \fBu\fR ] [ \fIvirtual_address\fR ]\fR
.ad
.sp .6
.RS 4n
With no arguments, display the current I/O device characteristics including:
current input device, current output device, baud rates for serial ports A and
B, an input-to-output echo indicator, and virtual addresses of mapped
\fBUART\fR devices. With arguments, set or configure the current I/O device.
With the \fBu\fR argument (\fBuu\fR.\|.\|.), set the I/O device to be the
\fIvirtual_address\fR of a \fBUART\fR device currently mapped.
.sp
.ne 2
.na
\fB\fBecho\fR\fR
.ad
.RS 13n
 Can be either \fBe\fR to enable input to be echoed to the output device, or
```



```
\fBne\fR, to indicate that input is not echoed.
.RE

.sp
.ne 2
.na
\fB\fIport\fR\fR
.ad
.RS 13n
Assign the indicated \fIport\fR to be the current I/O device. \fIport\fR can be
one of:
.sp
.ne 2
.na
\fB\fBa\fR\fR
.ad
.RS 5n
serial port A
.RE

.sp
.ne 2
.na
\fB\fBb\fR\fR
.ad
.RS 5n
serial port B
.RE

.sp
.ne 2
.na
\fB\fBk\fR\fR
.ad
.RS 5n
the workstation keyboard
.RE

.sp
.ne 2
.na
\fB\fBs\fR\fR
.ad
.RS 5n
the workstation screen
.RE

.RE

.sp
.ne 2
.na
\fB\fIbaud_rate\fR\fR
.ad
.RS 13n
Any legal baud rate.
.RE

.sp
.ne 2
.na
\fB\fIoptions\fR\fR
.ad
.RS 11n
can be any combination of:
.sp
```

```
.ne 2
.na
\fB\fBi\fR\fR
.ad
.RS 6n
input
.RE

.sp
.ne 2
.na
\fB\fBo\fR\fR
.ad
.RS 6n
output
.RE

.sp
.ne 2
.na
\fB\fBu\fR\fR
.ad
.RS 6n
\fBUART\fR
.RE

.sp
.ne 2
.na
\fB\fBe\fR\fR
.ad
.RS 6n
echo input to output
.RE

.sp
.ne 2
.na
\fB\fBne\fR\fR
.ad
.RS 6n
do not echo input
.RE

.sp
.ne 2
.na
\fB\fBr\fR\fR
.ad
.RS 6n
reset indicated serial port (\fBa\fR and \fBb\fR ports only)
.RE

If either \fBa\fR or \fBb\fR is supplied, and no \fIoptions\fR are given, the
serial port is assigned for both input and output.  If \fBk\fR is supplied with
no options, it is assigned for input only.  If \fBs\fR is supplied with no
options, it is assigned for output only.
.RE

.RE

.sp
.ne 2
.na
\fB\fBv\fR\fI virtual_address1 virtual_address2 \fR [\fBsize\fR]\fR
.ad
```



```
.sp .6
.RS 4n
Display the contents of \fIvirtual_address1\fR (lower) \fIvirtual_address2\fR
(higher) in the format specified by \fBsize\fR:
.sp
.ne 2
.na
\fB\fBb\fR\fR
.ad
.RS 5n
byte format (the default)
.RE

.sp
.ne 2
.na
\fB\fBw\fR\fR
.ad
.RS 5n
word format
.RE

.sp
.ne 2
.na
\fB\fBl\fR\fR
.ad
.RS 5n
long word format
.RE

Enter return to pause for viewing; enter another return character to resume the
display.  To terminate the display at any time, press the space bar.
.sp
For example, the following command displays the contents of virtual address
space from address  \fB0x1000\fR to  \fB0x2000\fR in word format:
.sp
\fBv\fR \fB1000\fR \fB2000\fR \fBW\fR
.RE

.sp
.ne 2
.na
\fB\fBw\|\fR [\fIvirtual_address\|\fR]\|[\fIargument\|\fR]\fR
.ad
.sp .6
.RS 4n
Set the execution vector to a predetermined or default routine. Pass
\fIvirtual_address\fR and \fIargument\fR to that routine.
.sp
To set up a predetermined routine to jump to, a user program must, prior to
executing the monitor's \fBw\fR command, set the variable
\fB*romp->v_vector_cmd\fR to be equal to the virtual address of the desired
routine. Predetermined routines need not necessarily return control to the
monitor.
.sp
The default routine, defined by the monitor, prints the user-supplied
\fIvector\fR according to the format supplied in \fIargument\fR. This format
can be one of:
.sp
.ne 2
.na
\fB\fB%x\fR\fR
.ad
.RS 6n
hexadecimal
```

```
.RE

.sp
.ne 2
.na
\fB\fB%d\fR\fR
.ad
.RS 6n
decimal
.RE

.RE

.sp
.ne 2
.na
\fB\fBx\fR\fR
.ad
.sp .6
.RS 4n
Display a menu of extended tests. These diagnostics permit additional testing
of such things as the I/O port connectors, video memory, workstation memory and
keyboard, and boot device paths.
.RE

.sp
.ne 2
.na
\fB\fBy\|c\fR\fI context_number\fR\fR
.ad
.sp .6
.RS 4n

.RE

.sp
.ne 2
.na
\fB\fBy\|p|s\|\fR\fI context_number virtual_address\fR\fR
.ad
.sp .6
.RS 4n
Flush the indicated context, context page, or context segment.
.sp
.ne 2
.na
\fB\fBc\fR\fR
.ad
.RS 5n
flush context \fIcontext_number\fR
.RE

.sp
.ne 2
.na
\fB\fBp\fR\fR
.ad
.RS 5n
flush the page beginning at \fIvirtual_address\fR within context
\fIcontext_number\fR
.RE

.sp
.ne 2
.na
\fB\fBs\fR\fR
```



```
.ad
.RS 5n
flush the segment beginning at \fIvirtual_address\fR within context
\fIcontext_number\fR
.RE

.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
_
Architecture    SPARC
.TE

.SH SEE ALSO
.sp
.LP
\fBtip\fR(1), \fBboot\fR(1M), \fBeeprom\fR(1M), \fBattributes\fR(5)
.sp
.LP
```

```
**********************************************************
    5885 Mon Aug 26 06:56:03 2019
new/usr/src/man/man1m/prtconf.1m
11622 clean up rarer mandoc lint warnings
**********************************************************
'\" te
.\"  Copyright 1989 AT&T Copyright (c) 2003, Sun Microsystems, Inc. All Rights R
.\"  Copyright 2012, Joyent, Inc. All Rights Reserved
.\"  Copyright 2019, Peter tribble.
.\" The contents of this file are subject to the terms of the Common Development
.\" See the License for the specific language governing permissions and limitati
.\" fields enclosed by brackets "[]" replaced with your own identifying informat
.TH PRTCONF 1M "Jan 21, 2019"
.SH NAME
prtconf \- print system configuration
.SH SYNOPSIS
.LP
.nf
\fB/usr/sbin/prtconf\fR [\fB-V\fR] | [\fB-F\fR] | [\fB-m\fR] | [\fB-x\fR] | [\fB
\fI/usr/sbin/prtconf\fR [\fB-V\fR] | [\fB-F\fR] | [\fB-m\fr] | [\fB-x\fR] | [\fB
     [\fIdev_path\fR]
.fi

.SH DESCRIPTION
.LP
The \fBprtconf\fR command prints the system configuration information. The
output includes the total amount of memory, and the configuration of system
peripherals formatted as a device tree.
.sp
.LP
If a device path is specified on the command line for those command options
that can take a device path, \fBprtconf\fR will only display information for
that device node.
.SH OPTIONS
.LP
The following options are supported:
.sp
.ne 2
.na
\fB-a\fR
.ad
.RS 6n
Display all the ancestors device nodes, up to the root node of the device tree,
for the device specified on the command line.
.RE

.sp
.ne 2
.na
\fB-b\fR
.ad
.RS 6n
Display the firmware device tree root properties for the purpose of platform
identification. These properties are "name", "compatible", "banner-name" and
"model".
.RE

.sp
.ne 2
.na
\fB-c\fR
.ad
.RS 6n
Display the device subtree rooted at the device node specified on the command
line, that is, display all the children of the device node specified on the
command line.
```



```
.RE

.sp
.ne 2
.na
\fB-d\fR
.ad
.RS 6n
Display vendor ID and device ID for PCI and PCI Express devices, in addition to
the nodename. If the information is known, the vendor name and device name will
also be shown.
.RE

.sp
.ne 2
.na
\fB-D\fR
.ad
.RS 6n
For each system peripheral in the device tree, displays the name of the device
driver used to manage the peripheral.
.RE

.sp
.ne 2
.na
\fB-F\fR
.ad
.RS 6n
Returns the device path name of the console frame buffer,
if one exists. If there is no frame buffer, \fBprtconf\fR returns a non-zero
exit code. This flag must be used by itself. It returns only the name of the
console, frame buffer device or a non-zero exit code. For example, if the
console frame buffer on a SUNW,Ultra-30 is \fBffb\fR, the command returns:
\fB/SUNW,ffb@1e,0:ffb0\fR. This option could be used to create a symlink for
\fB/dev/fb\fR to the actual console device.
.RE

.sp
.ne 2
.na
\fB-m\fR
.ad
.RS 6n
Displays the amount of system memory in megabytes.
This flag must be used by itself.
.RE


.ne 2
.na
\fB-p\fR
.ad
.RS 6n
Displays information derived from the device tree provided by the firmware
(PROM) on SPARC platforms or the booting system on x86 platforms.The device
tree information displayed using this option is a snapshot of the initial
configuration and may not accurately reflect reconfiguration events that occur
later.
.RE

.sp
.ne 2
.na
\fB-P\fR
.ad
```

```
124 .RS 6n
125 Includes information about pseudo devices. By default, information regarding
126 pseudo devices is omitted.
127 .RE

129 .sp
130 .ne 2
131 .na
132 \fB-v\fR
133 .ad
134 .RS 6n
135 Specifies verbose mode.
136 .RE

138 .sp
139 .ne 2
140 .na
141 \fB-V\fR
142 .ad
143 .RS 6n
144 Displays platform-dependent \fBPROM\fR (on SPARC platforms) or booting system
145 (on x86 platforms) version information. This flag must be used by itself. The
146 output is a string. The format of the string is arbitrary and
147 platform-dependent.
148 .RE

150 .sp
151 .ne 2
152 .na
153 \fB-x\fR
154 .ad
155 .RS 6n
156 A legacy flag that reported if the firmware on this system is 64-bit ready.
157 As illumos only runs on 64-bit platforms, this flag is kept for compatibility
158 only, and zero is always returned.
159 .sp
160 This flag overrides all other flags and must be used by itself.
161 .RE

163 .SH OPERANDS
167 .LP
164 The following operands are supported:
165 .sp
166 .ne 2
167 .na
168 \fIdev_path\fR
169 .ad
170 .RS 12n
171 The path to a target device minor node, device nexus node, or device link for
172 which device node configuration information is displayed
173 .RE

175 .SH EXIT STATUS
180 .LP
176 The following exit values are returned:
177 .sp
178 .ne 2
179 .na
180 \fB0\fR
181 .ad
182 .RS 12n
183 No error occurred.
184 .RE

186 .sp
187 .ne 2
```



```
188 .na
189 \fBnon-zero\fR
190 .ad
191 .RS 12n
192 With the \fB-F\fR option, a non-zero return value means that the
193 output device is not a frame buffer. In all other cases, a
194 non-zero return value means that an error occurred.
195 .RE

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

201 .sp
202 .TS
203 box;
204 c | c
205 l | l .
206 ATTRIBUTE TYPE  ATTRIBUTE VALUE
207 Interface Stability     Unstable
208 .TE

210 .SH SEE ALSO
217 .LP
211 \fBfuser\fR(1M), \fBmodinfo\fR(1M), \fBsysdef\fR(1M), \fBattributes\fR(5),
212 \fBopenprom\fR(7D)
213 .SH NOTES
221 .LP
214 The output of the \fBprtconf\fR command is highly dependent on the version of
215 the \fBPROM\fR installed in the system. The output will be affected in
216 potentially all circumstances.
217 .sp
218 .LP
219 The \fBdriver not attached\fR message means that no driver is currently
220 attached to that instance of the device. In general, drivers are loaded and
221 installed (and attached to hardware instances) on demand, and when needed, and
222 may be uninstalled and unloaded when the device is not in use.
223 .sp
224 .LP
225 On x86 platforms, the use of \fBprtconf\fR \fB-vp\fR provides a subset of
226 information from \fBprtconf\fR \fB-v\fR. The value of integer properties from
227 \fBprtconf\fR \fB-vp\fR might require byte swapping for correct interpretation.
```

```
**********************************************************
   12282 Mon Aug 26 06:56:03 2019
new/usr/src/man/man1m/ypserv.1m
11622 clean up rarer mandoc lint warnings
**********************************************************
   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 YPSERV 1M "Dec 15, 2004"
   7 .SH NAME
   8 ypserv, ypxfrd \- NIS server and binder processes
   9 .SH SYNOPSIS
  10 .LP
  10 .nf
  11 \fB/usr/lib/netsvc/yp/ypserv\fR [\fB-dv\fR] [\fB-i\fR | \fB-I\fR] [\fB-r\fR | \f
  12 .fi

  14 .LP
  15 .nf
  16 \fB/usr/lib/netsvc/yp/ypxfrd\fR
  17 .fi

  19 .SH DESCRIPTION
  21 .sp
  22 .LP
  20 The Network Information Service (\fBNIS\fR) provides a simple network lookup
  21 service consisting of databases and processes. The databases are \fBndbm\fR
  22 files in a directory tree rooted at \fB/var/yp\fR. See \fBndbm\fR(3C). These
  23 files are described in \fBypfiles\fR(4). The processes are
  24 \fB/usr/lib/netsvc/yp/ypserv\fR, the \fBNIS\fR database lookup server, and
  25 \fB/usr/lib/netsvc/yp/ypbind\fR, the \fBNIS\fR binder. The programmatic
  26 interface to the \fBNIS\fR service is described in \fBypclnt\fR(3NSL).
  27 Administrative tools are described in \fByppoll\fR(1M), \fByppush\fR(1M),
  28 \fBypset\fR(1M), \fBypxfr\fR(1M), and \fBypwhich\fR(1). Tools to see the
  29 contents of \fBNIS\fR maps are described in \fBypcat\fR(1), and
  30 \fBypmatch\fR(1). Database generation and maintenance tools are described in
  31 \fBypinit\fR(1M), \fBypmake\fR(1M), and \fBmakedbm\fR(1M).
  32 .sp
  33 .LP
  34 The \fBypserv\fR utility is a daemon process typically activated at system
  35 startup from \fBsvc:/network/nis/server:default\fR. Alternatively, you can, as
  36 the root user, start \fBNIS\fR services using \fBypstart\fR(1M) from the
  37 command-line. \fBypserv\fR runs only on \fBNIS\fR server machines with a
  38 complete \fBNIS\fR database. You can halt all \fBNIS\fR services using the
  39 \fBypstop\fR(1M) command.
  40 .sp
  41 .LP
  42 The \fBypxfrd\fR utility transfers entire \fBNIS\fR maps in an efficient
  43 manner. For systems that use this daemon, map transfers are 10 to 100 times
  44 faster, depending on the map. To use this daemon, be sure \fBypxfrd\fR is
  45 running on the master server. See \fB/usr/lib/netsvc/yp/ypstart\fR. \fBypxfr\fR
  46 attempts to use \fBypxfrd\fR first. If that fails, it prints a warning, then
  47 uses the older transfer method.
  48 .sp
  49 .LP
  50 The \fBypserv\fR daemon's primary function is to look up information in its
  51 local database of \fBNIS\fR maps.
  52 .sp
  53 .LP
  54 The operations performed by \fBypserv\fR are defined for the implementor by the
  55 \fIYP Protocol Specification\fR, and for the programmer by the header file
  56 <\fBrpcsvc/yp_prot.h\fR>.
  57 .sp
  58 .LP
```



```
  59 Communication to and from \fBypserv\fR is by means of \fBRPC\fR calls. Lookup
  60 functions are described in \fBypclnt\fR(3NSL), and are supplied as C-callable
  61 functions in the \fBlibnsl\fR(3LIB) library. There are four lookup functions,
  62 all of which are performed on a specified map within some \fBNIS\fR domain:
  63 \fByp_match\fR(3NSL), \fByp_first\fR(3NSL), \fByp_next\fR(3NSL), and
  64 \fByp_all\fR(3NSL). The \fByp_match\fR operation takes a key, and returns the
  65 associated value. The \fByp_first\fR operation returns the first key-value pair
  66 from the map, and \fByp_next\fR can be used to enumerate the remainder.
  67 \fByp_all\fR ships the entire map to the requester as the response to a single
  68 \fBRPC\fR request.
  69 .sp
  70 .LP
  71 A number of special keys in the \fBDBM\fR files can alter the way in which
  72 \fBypserv\fR operates. The keys of interest are:
  73 .sp
  74 .ne 2
  75 .na
  76 \fB\fBYP_INTERDOMAIN\fR\fR
  77 .ad
  78 .RS 21n
  79 The presence of this key causes \fBypserv\fR to forward to a \fBDNS\fR server
  80 host lookups that cannot be satisfied by the \fBDBM\fR files.
  81 .RE

  83 .sp
  84 .ne 2
  85 .na
  86 \fB\fBYP_SECURE\fR\fR
  87 .ad
  88 .RS 21n
  89 This key causes \fBypserv\fR to answer only questions coming from clients on
  90 reserved ports.
  91 .RE

  93 .sp
  94 .ne 2
  95 .na
  96 \fB\fBYP_MULTI_\fR\fIhostname\fR\fR
  97 .ad
  98 .RS 21n
  99 This is a special key in the form, \fBYP_MULTI_\fR\fIhostname
 100 addr1,...,addrN.\fR A client looking for \fIhostname\fR receives the closest
 101 address.
 102 .RE

 104 .sp
 105 .LP
 106 Two other functions supply information about the map, rather than map entries:
 107 \fByp_order\fR(3NSL), and \fByp_master\fR(3NSL). In fact, both order number and
 108 master name exist in the map as key-value pairs, but the server will not return
 109 either through the normal lookup functions. If you examine the map with
 110 \fBmakedbm\fR(1M), however, they are visible. Other functions are used within
 111 the \fBNIS\fR service subsystem itself, and are not of general interest to
 112 \fBNIS\fR clients. These functions include \fBdo_you_serve_this_domain?\fR,
 113 \fBtransfer_map\fR, and \fBreinitialize_internal_state\fR.
 114 .sp
 115 .LP
 116 On start up, \fBypserv\fR checks for the existence of the NIS to LDAP (N2L)
 117 configuration file \fB/var/yp/NISLDAPmapping\fR. If it is present then a master
 118 server starts in N2L mode. If the file is not present it starts in
 119 "traditional" (non N2L) mode. Slave servers always start in traditional mode.
 120 .sp
 121 .LP
 122 In N2L mode, a new set of map files, with an \fBLDAP_\fR prefix, are generated,
 123 based on the contents of the LDAP DIT. The old map files, NIS source files and
 124 \fBypmake\fR(1M) are not used.
```

```
.sp
.LP
It is possible that \fBypmake\fR(1M) can be accidentally run in N2L mode. If
the occurs, the old style map files are overwritten. That the map files are
overwritten is harmless. However, any resulting \fByppush\fR(1M) operation will
push information based on the DIT rather than the source files. The user may
not expect information based on the DIT. \fBypserv\fR keeps track of the last
modification date of the old style map files. If the map files have been
updated, a warning is logged that suggests that the user call \fByppush\fR
directly instead of \fBypmake\fR.
.sp
.LP
If a server attempts to run in N2L mode and a LDAP server cannot be contacted,
it behaves as follows:
.RS +4
.TP
1.
When \fBypserv\fR is started, a warning will be logged.
.RE
.RS +4
.TP
2.
When a NIS read access is made and the TTL entry has expired, a warning is
logged.Information that is returned from the cache has not been updated.
.RE
.RS +4
.TP
3.
When a NIS write access is made, a warning is logged. The cache will not be
updated, and a NIS failure will be returned.
.RE
.sp
.LP
If \fBypxfrd\fR is running in N2L mode and is asked to transfer a map,
\fBypxfrd\fR first checks whether the map is out of date. If the map is out of
date, \fBypxfrd\fR initiates an update from the DIT. \fBypxfrd\fR cannot wait
for the update to complete. If \fBypxfrd\fR waited, the client end \fBypxfr\fR
operation could time out. To prevent \fBypxfrd\fR from timing out, the existing
map is transferred from the cache. The most up to date map will be transferred
on subsequent \fBypxfrd\fR operations.
.SH OPTIONS
.SS "ypserv"
.sp
.ne 2
.na
\fB\fB-d\fR\fR
.ad
.RS 7n
The \fBNIS\fR service should go to the \fBDNS\fR for more host information.
This requires the existence of a correct \fB/etc/resolv.conf\fR file pointing
to a \fBDNS\fR server. This option turns on \fBDNS\fR forwarding regardless of
whether or not the \fBYP_INTERDOMAIN\fR flag is set in the \fBhosts\fR maps.
See \fBmakedbm\fR(1M). In the absence of an \fB/etc/resolv.conf\fR file,
\fBypserv\fR complains, but ignores the \fB-d\fR option.
.RE

.sp
.ne 2
.na
\fB\fB-i\fR\fR
.ad
.RS 7n
If in N2L mode, initialize the NIS related parts of the \fBDIT\fR based on the
current, non \fBLDAP_\fR prefixed, map files. The \fBLDAP_\fR prefixed maps are
not created or updated. If you require that \fBLDAP_\fR prefixed maps be
updated or created, then use the \fB-ir\fR option.
```



```
.sp
The \fB-i\fR option does not attempt to create any NIS domain or container
objects. If any NIS domain or container objects have not already been created,
then errors will occur, as entries are written to nonexistent containers.
.RE

.sp
.ne 2
.na
\fB\fB-I\fR\fR
.ad
.RS 7n
Identical to \fB-i\fR, except that any missing domain and container objects are
created.
.RE

.sp
.ne 2
.na
\fB\fB-r\fR\fR
.ad
.RS 7n
If in N2L mode, then refresh the \fBLDAP_\fR prefixed map files based on the
contents of the \fBDIT\fR.
.RE

.sp
.ne 2
.na
\fB\fB-ir\fR\fR
.ad
.RS 7n
If both \fB-i\fR and \fB-r\fR are specified in N2L mode, then the \fBDIT\fR
will first be initialized from the current non \fBLDAP_\fR prefixed map files.
A new set of \fBLDAP_\fR prefixed maps will then be generated from the contents
of the \fBDIT\fR. A new set of \fBLDAP_\fR prefixed maps is required when
moving from traditional NIS to N2L mode NIS.
.RE

.sp
.ne 2
.na
\fB\fB-Ir\fR\fR
.ad
.RS 7n
Identical to \fB-ir\fR, except that any missing domain and container objects
are created.
.RE

.sp
.ne 2
.na
\fB\fB-v\fR\fR
.ad
.RS 7n
Operate in the verbose mode, printing diagnostic messages to stderr.
.RE

.sp
.LP
When run with the \fB-i\fR, \fB-r\fR, \fB-I\fR, \fB-ir\fR or \fB-Ir\fR options,
the \fBypserv\fR command runs in the foreground and exits once map
initialization has been completed. Once the \fBypserv\fR command exits, the
user knows the maps are ready and can restart \fBypserv\fR and the other
\fByp\fR daemons by running \fBypstart\fR(1M).
.sp
```

```
256 .LP
257 If there is a requirement to initialize the \fBDIT\fR from the NIS source
258 files, which may have been modified since the maps were last remade, run
259 \fBypmake\fR before running \fBypserv\fR \fB-i\fR or \fBypserv\fR \fB-ir\fR.
260 \fBypmake\fR regenerated old style NIS maps. Then \fBypserv\fR \fB-ir\fR dumps
261 them into the \fBDIT\fR. When the \fB-ir\fR option is used, the \fBLDAP_\fR
262 prefixe maps are also generated or updated. Since these maps will be more
263 recent than the old style maps, \fBypmake\fR will not be reported as erroneous
264 when it is run.
265 .SH FILES
270 .sp
266 .ne 2
267 .na
268 \fB\fB/var/yp/securenets\fR\fR
269 .ad
270 .sp .6
271 .RS 4n
272 Defines the hosts and networks that are granted access to information in the
273 served domain. It is read at startup time by both \fBypserv\fR and
274 \fBypxfrd\fR.
275 .RE

277 .sp
278 .ne 2
279 .na
280 \fB\fB/var/yp/ypserv.log\fR\fR
281 .ad
282 .sp .6
283 .RS 4n
284 If the \fB/var/yp/ypserv.log\fR file exists when \fBypserv\fR starts up, log
285 information is written to it when error conditions arise.
286 .RE

288 .sp
289 .ne 2
290 .na
291 \fB\fB/var/yp/binding/domainname/ypservers\fR\fR
292 .ad
293 .sp .6
294 .RS 4n
295 Lists the \fBNIS\fR server hosts that \fBypbind\fR can bind to.
296 .RE

298 .SH SEE ALSO
304 .sp
305 .LP
299 \fBsvcs\fR(1), \fBypcat\fR(1), \fBypmatch\fR(1), \fBypwhich\fR(1),
300 \fBdomainname\fR(1M), \fBmakedbm\fR(1M), \fBsvcadm\fR(1M), \fBypbind\fR(1M),
301 \fBypinit\fR(1M), \fBypmake\fR(1M), \fByppoll\fR(1M), \fByppush\fR(1M),
302 \fBypset\fR(1M), \fBypstart\fR(1M), \fBypstop\fR(1M), \fBypxfr\fR(1M),
303 \fBndbm\fR(3C), \fBypclnt\fR(3NSL), \fBlibnsl\fR(3LIB),
304 \fBNISLDAPmapping\fR(4), \fBsecurenets\fR(4), \fBypfiles\fR(4),
305 \fBypserv\fR(4), \fBattributes\fR(5), \fBsmf\fR(5)
313 .sp
314 .LP

316 .sp
317 .LP
318 \fI\fR
307 .SH NOTES
320 .sp
321 .LP
308 \fBypserv\fR supports multiple domains. The \fBypserv\fR process determines the
309 domains it serves by looking for directories of the same name in the directory
310 \fB/var/yp\fR. It replies to all broadcasts requesting yp service for that
311 domain.
```



```
312 .sp
313 .LP
314 The Network Information Service (\fBNIS\fR) was formerly known as Sun Yellow
315 Pages (\fBYP\fR). The functionality of the two remains the same; only the name
316 has changed. The name Yellow Pages is a registered trademark in the United
317 Kingdom of British Telecommunications PLC, and must not be used without
318 permission.
319 .sp
320 .LP
321 \fBNIS\fR uses \fBndbm()\fR files to store maps. Therefore, it is subject to
322 the 1024 byte limitations described in the USAGE and NOTES sections of the
323 \fBndbm\fR(3C) man page.
324 .sp
325 .LP
326 The NIS server service is managed by the service management facility,
327 \fBsmf\fR(5), under the service identifier:
328 .sp
329 .in +2
330 .nf
331 svc:/network/nis/server:default
332 .fi
333 .in -2
334 .sp

336 .sp
337 .LP
338 Administrative actions on this service, such as enabling, disabling, or
339 requesting restart, can be performed using \fBsvcadm\fR(1M). The service's
340 status can be queried using the \fBsvcs\fR(1) command.
```

```
**********************************************************
    2553 Mon Aug 26 06:56:03 2019
new/usr/src/man/man3c/epoll_create.3c
11622 clean up rarer mandoc lint warnings
**********************************************************
'\" te
.\"  Copyright (c) 2014, Joyent, Inc. All Rights Reserved.
.\"  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.
.TH EPOLL_CREATE 3C "April 9, 2016"
.SH NAME
epoll_create, epoll_create1 \- create an epoll instance
.SH SYNOPSIS

.LP
.nf
#include <sys/epoll.h>

\fBint\fR \fBepoll_create\fR(\fBint\fR \fIsize\fR);
.fi

.LP
.nf
\fBint\fR \fBepoll_create1\fR(\fBint\fR \fIflags\fR);
.fi

.SH DESCRIPTION
.LP
The \fBepoll_create()\fR and \fBepoll_create1()\fR functions both create an
\fBepoll\fR(5) instance that can be operated upon via \fBepoll_ctl\fR(3C),
\fBepoll_wait\fR(3C) and \fBepoll_pwait\fR(3C).  \fBepoll\fR instances are
represented as file descriptors, and should be closed via \fBclose\fR(2).

The only difference between the two functions is their signature;
\fBepoll_create()\fR takes a size argument that
is vestigal and is only meaningful in as much as it must be greater than
zero, while \fBepoll_create1()\fR takes a flags argument that can have
any of the following values:

.sp
.ne 2
.na
\fBEPOLL_CLOEXEC\fR
.ad
.RS 12n
Instance should be closed upon an
\fBexec\fR(2); see \fBopen\fR(2)'s description of \fBO_CLOEXEC\fR.
.RE

.SH RETURN VALUES
.LP
Upon successful completion, 0 is returned. Otherwise, -1 is returned and errno
is set to indicate the error.
.SH ERRORS
.LP
The \fBepoll_create()\fR and \fBepoll_create1()\fR functions will fail if:
.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
```



```
.ad
.RS 10n
Either the \fIsize\fR is zero (\fBepoll_create()\fR) or the \fIflags\fR
are invalid (\fBepoll_create1()\fR).
.RE

.sp
.ne 2
.na
\fB\fBEMFILE\fR\fR
.ad
.RS 10n
There are currently {\fBOPEN_MAX\fR} file descriptors open in the calling
process.
.RE

.sp
.ne 2
.na
\fB\fBENFILE\fR\fR
.ad
.RS 10n
The maximum allowable number of files is currently open in the system.
.RE

.sp
.SH NOTES
.LP

The \fBepoll\fR(5) facility is implemented for purposes of offering
compatibility for Linux-borne applications; native
applications should continue to prefer using event ports via the
\fBport_create\fR(3C), \fBport_associate\fR(3C) and \fBport_get\fR(3C)
interfaces.  See \fBepoll\fR(5) for compatibility details and restrictions.

.SH SEE ALSO
.LP
\fBepoll_ctl\fR(3C), \fBepoll_wait\fR(3C), \fBepoll\fR(5)
```

```
**********************************************************
    6770 Mon Aug 26 06:56:03 2019
new/usr/src/man/man3c/epoll_ctl.3c
11622 clean up rarer mandoc lint warnings
**********************************************************
'\" te
.\"  Copyright (c) 2014, Joyent, Inc. All Rights Reserved.
.\"  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.
.TH EPOLL_CTL 3C "April 9, 2016"
.SH NAME
epoll_ctl \- control an epoll instance
.SH SYNOPSIS

.LP
.nf
#include <sys/epoll.h>

\fBint\fR \fBepoll_ctl\fR(\fBint\fR \fIepfd\fR, \fBint\fR \fIop\fR, \fBint\fR \f
.fi

.SH DESCRIPTION
.LP
The \fBepoll_ctl()\fR function executes the operation specified by
\fIop\fR (as parameterized by \fIevent\fR) on the \fIepfd\fR epoll instance.
Valid values for \fIop\fR:

.sp
.ne 2
.na
\fBEPOLL_CTL_ADD\fR
.ad
.RS 12n
For the \fBepoll\fR(5) instance specified by \fIepfd\fR,
associate the file descriptor specified by \fIfd\fR with the event specified
by \fIevent\fR.
.RE

.sp
.ne 2
.na
\fBEPOLL_CTL_DEL\fR
.ad
.RS 12n
For the \fBepoll\fR(5) instance specified by \fIepfd\fR,
remove all event associations for the file descriptor specified by \fIfd\fR.
\fIevent\fR is ignored, and may be NULL.
.RE

.sp
.ne 2
.na
\fBEPOLL_CTL_MOD\fR
.ad
.RS 12n
For the \fBepoll\fR(5) instance specified by \fIepfd\fR, modify the event
association for the file descriptor specified by \fIfd\fR to be that
specified by \fIevent\fR.

.RE
```



```
The \fIevent\fR parameter has the following structure:

.in +4
.nf
typedef union epoll_data {
        void    *ptr;
        int      fd;
        uint32_t u32;
        uint64_t u64;
} epoll_data_t;
_____unchanged_portion_omitted_
.fi
.in -4

The \fIdata\fR field specifies the datum to
be associated with the event and
will be returned via \fBepoll_wait\fR(3C).
The \fIevents\fR field denotes both the desired events (when specified via
\fBepoll_ctl()\fR) and the events that have occurred (when returned via
\fBepoll_wait\fR(3C)).
In either case, the
\fIevents\fR field is a bitmask constructed by a logical \fBOR\fR operation
of any combination of the following event flags:

.sp
.ne 2
.na
\fBEPOLLIN\fR
.ad
.RS 14n
Data other than high priority data may be read without blocking. For streams,
this flag is set in the returned \fIevents\fR even if the message is of
zero length.
.RE

.sp
.ne 2
.na
\fBEPOLLPRI\fR
.ad
.RS 14n
Normal data (priority band equals 0) may be read without blocking. For streams,
this flag is set in the returned \fIevents\fR even if the message is of zero
length.
.RE

.sp
.ne 2
.na
\fBEPOLLOUT\fR
.ad
.RS 14n
Normal data (priority band equals 0) may be written without blocking.
.RE

.sp
.ne 2
.na
\fBEPOLLRDNORM\fR
.ad
.RS 14n
Normal data (priority band equals 0) may be read without blocking. For streams,
this flag is set in the returned \fIrevents\fR even if the message is of
zero length.
.RE
```

```
.sp
.ne 2
.na
\fBEPOLLRDBAND\fR
.ad
.RS 14n
Data from a non-zero priority band may be read without blocking. For streams,
this flag is set in the returned \fIrevents\fR even if the message is of
zero length.
.RE

.sp
.ne 2
.na
\fBEPOLLWRNORM\fR
.ad
.RS 14n
The same as \fBEPOLLOUT\fR.
.RE

.sp
.ne 2
.na
\fBEPOLLWRBAND\fR
.ad
.RS 14n
Priority data (priority band > 0) may be written.  This event only examines
bands that have been written to at least once.
.RE

.sp
.ne 2
.na
\fBEPOLLMSG\fR
.ad
.RS 14n
This exists only for backwards binary and source compatibility with Linux;
it has no meaning and is ignored.
.RE

.sp
.ne 2
.na
\fBEPOLLERR\fR
.ad
.RS 14n
An error has occurred on the device or stream.  This flag is only valid in the
returned \fIevents\fR field.
.RE

.sp
.ne 2
.na
\fBEPOLLHUP\fR
.ad
.RS 14n
A hangup has occurred on the stream. This event and \fBEPOLLOUT\fR are mutually
exclusive; a stream can never be writable if a hangup has occurred. However,
this event and  \fBEPOLLIN\fR, \fBEPOLLRDNORM\fR, \fBEPOLLRDBAND\fR,
\fBEPOLLRDHUP\fR or
\fBEPOLLPRI\fR are not mutually exclusive. This flag is only valid in the
\fIevents\fR field returned from \fBepoll_wait\fR(3C); it is not used in the
\fIevents\fR field specified via \fBepoll_ctl()\fR.
.RE
```



```
.sp
.ne 2
.na
\fBEPOLLRDHUP\fR
.ad
.RS 14n
The stream socket peer shutdown the writing half of the connection and no
further data will be readable via the socket.  This event is not mutually
exclusive with \fBEPOLLIN\fR.
.RE

.sp
.ne 2
.na
\fBEPOLLWAKEUP\fR
.ad
.RS 14n
This exists only for backwards binary and source compatibility with Linux;
it has no meaning and is ignored.
.RE

.sp
.ne 2
.na
\fBEPOLLONESHOT\fR
.ad
.RS 14n
Sets the specified event to be in one-shot mode, whereby the event association
with the \fBepoll\fR(5) instance specified by \fIepfd\fR is removed atomically
as the event is returned via \fBepoll_wait\fR(3C).  Use of this mode allows
for resolution of some of the
races inherent in multithreaded use of \fBepoll_wait\fR(3C).
.RE

.sp
.ne 2
.na
\fBEPOLLET\fR
.ad
.RS 14n
Sets the specified event to be edge-triggered mode instead of the default
mode of level-triggered.  In this mode, events will be induced by
transitions on an event source rather than the state of the event source.
While perhaps superficially appealing, this mode introduces several new
potential failure modes for user-level software and should be used
with caution.
.RE

.SH RETURN VALUES
.LP
Upon successful completion, \fBepoll_ctl()\fR returns 0.
If an error occurs, -1 is returned and errno is set to indicate
the error.

.SH ERRORS
.LP
\fBepoll_ctl()\fR will fail if:
.sp
.ne 2
.na
\fB\fBEBADF\fR\fR
.ad
.RS 10n
\fIepfd\fR is not a valid file descriptor.
.RE
```

```
.sp
.ne 2
.na
\fB\fBEFAULT\fR\fR
.ad
.RS 10n
The memory associated with \fIevent\fR was not mapped.
.RE

.sp
.ne 2
.na
\fB\fBEEXIST\fR\fR
.ad
.RS 10n
The operation specified was \fBEPOLL_CTL_ADD\fR and the specified file
descriptor is already associated with an event for the specified
\fBepoll\fR(5) instance.
.RE

.sp
.ne 2
.na
\fB\fBENOENT\fR\fR
.ad
.RS 10n
The operation specified was \fBEPOLL_CTL_MOD\fR or \fBEPOLL_CTL_DEL\fR and
the specified file descriptor is not associated with an event for the
specified \fBepoll\fR(5) instance.
.RE

.sp
.SH NOTES
.LP

The \fBepoll\fR(5) facility is implemented for purposes of offering
compatibility for Linux-borne applications; native
applications should continue to prefer using event ports via the
\fBport_create\fR(3C), \fBport_associate\fR(3C) and \fBport_get\fR(3C)
interfaces.  See \fBepoll\fR(5) for compatibility details and restrictions.

.SH SEE ALSO
.LP
\fBepoll_create\fR(3C), \fBepoll_wait\fR(3C),
\fBport_create\fR(3C), \fBport_associate\fR(3C), \fBport_get\fR(3C),
\fBepoll\fR(5)
```

```
**********************************************************
    3137 Mon Aug 26 06:56:03 2019
new/usr/src/man/man3c/epoll_wait.3c
11622 clean up rarer mandoc lint warnings
**********************************************************
'\" te
.\"  Copyright (c) 2014, Joyent, Inc. All Rights Reserved.
.\"  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.
.TH EPOLL_WAIT 3C "Apr 17, 2014"
.SH NAME
epoll_wait, epoll_pwait \- wait for epoll events
.SH SYNOPSIS

.LP
.nf
#include <sys/epoll.h>

\fBint\fR \fBepoll_wait\fR(\fBint\fR \fIepfd\fR, \fBstruct epoll_event *\fR\fIev
     \fBint\fR \fImaxevents\fR, \fBint\fR \fItimeout\fR);
.fi

.LP
.nf
\fBint\fR \fBepoll_pwait\fR(\fBint\fR \fIepfd\fR, \fBstruct epoll_event *\fR\fIe
     \fBint\fR \fImaxevents\fR, \fBint\fR \fItimeout\fR,
     \fBconst sigset_t *\fR\fIsigmask\fR);
.fi

.SH DESCRIPTION
.LP
The \fBepoll_wait()\fR function waits for events on the \fBepoll\fR(5)
instance specified by \fIepfd\fR.  The \fIevents\fR parameter must point to
an array of \fImaxevents\fR \fIepoll_event\fR structures to be
filled in with pending events.  The \fItimeout\fR argument specifies the
number of milliseconds to wait for an event if none is pending.  A
\fItimeout\fR of -1 denotes an infinite timeout.

The \fBepoll_pwait()\fR is similar to \fBepoll_wait()\fR, but takes an
additional \fIsigmask\fR argument that specifies the desired signal mask
when \fBepoll_pwait()\fR is blocked.  It is equivalent to atomically
setting the signal mask, calling \fBepoll_wait()\fR, and restoring the
signal mask upon return, and is therefore similar to the relationship
between \fBselect\fR(3C) and \fBpselect\fR(3C).

.SH RETURN VALUES
.LP
Upon successful completion, \fBepoll_wait()\fR and \fBepoll_pwait()\fR return
the number of events, or 0 if none was pending and \fItimeout\fR milliseconds
elapsed.  If an error occurs, -1 is returned and errno is set to indicate
the error.

.SH ERRORS
.LP
The \fBepoll_wait()\fR and \fBepoll_pwait()\fR functions will fail if:
.sp
.ne 2
.na
\fB\fBEBADF\fR\fR
.ad
```



```
.RS 10n
\fIepfd\fR is not a valid file descriptor.
.RE

.sp
.ne 2
.na
\fB\fBEFAULT\fR\fR
.ad
.RS 10n
The memory associated with \fIevents\fR was not mapped or was not writable.
.RE

.sp
.ne 2
.na
\fB\fBEINTR\fR\fR
.ad
.RS 10n
A signal was received during the \fBepoll_wait()\fR or \fBepoll_pwait()\fR.
.RE

.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 10n
Either \fIepfd\fR is not a valid \fBepoll\fR(5) instance or \fImaxevents\fR
is not greater than zero.
.RE

.sp
.SH NOTES
.LP
The \fBepoll\fR(5) facility is implemented for purposes of offering
compatibility for Linux-borne applications; native
applications should continue to prefer using event ports via the
\fBport_create\fR(3C), \fBport_associate\fR(3C) and \fBport_get\fR(3C)
interfaces.  See \fBepoll\fR(5) for compatibility details and restrictions.

.SH SEE ALSO
.LP
\fBepoll_create\fR(3C), \fBepoll_ctl\fR(3C),
\fBport_create\fR(3C), \fBport_associate\fR(3C), \fBport_get\fR(3C),
\fBpselect\fR(3C), \fBepoll\fR(5)
```

```
**********************************************************
    8030 Mon Aug 26 06:56:03 2019
new/usr/src/man/man3c/getlogin.3c
11622 clean up rarer mandoc lint warnings
**********************************************************
.\"
.\" 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) 2004 Sun Microsystems, Inc.  All Rights Reserved.
.\" Copyright (c) 2013 Gary Mills
.\"
.TH GETLOGIN 3C "Mar 15, 2014"
.SH NAME
getlogin, getlogin_r \- get login name
.SH SYNOPSIS
.LP
.nf
#include <unistd.h>

\fBchar *\fR\fBgetlogin\fR(\fBvoid\fR);
.fi

.LP
.nf
\fBchar *\fR\fBgetlogin_r\fR(\fBchar *\fR\fIname\fR, \fBint\fR \fInamelen\fR);
```



```
.fi

.SS "Standard conforming"
.LP
.nf
cc [ \fIflag \fR... ] \fIfile\fR... \fB-D_POSIX_PTHREAD_SEMANTICS\fR [ \fIlibrar

\fBint\fR \fBgetlogin_r\fR(\fBchar *\fR\fIname\fR, \fBsize_t\fR \fInamesize\fR);
.fi

.SH DESCRIPTION
.sp
.LP
The \fBgetlogin()\fR function returns a pointer to the login name as found in
\fB/var/adm/utmpx\fR. It can be used in conjunction with \fBgetpwnam\fR(3C) to
locate the correct password file entry when the same user \fBID\fR is shared by
several login names.
.sp
.LP
The login name plus the terminating null byte can be up to 33 characters
in length.
Newly-compiled programs should use the \fBLOGIN_NAME_MAX\fR symbol,
defined in <\fBlimits.h\fR>, to size the buffer.
Older programs that call \fBgetlogin()\fR  expect only the legacy
9-character length.
These automatically link to a version of the \fBgetlogin()\fR functions that
truncates longer login names.
It’s also possible to compile new programs that link to truncating versions
of these functions by defining \fB__USE_LEGACY_LOGNAME__\fR in the
compile environment.
.sp
.LP
Some older programs will correctly handle long login names returned
by the \fBgetlogin()\fR function.
For this case, the user compatibility library
\fB/usr/lib/getloginx.so.1\fR redirects to a version of the \fBgetlogin()\fR
function that returns the long name.
This library should be added to such an application
at runtime using \fBLD_PRELOAD\fR.
.sp
.LP
If \fBgetlogin()\fR is called within a process that is not attached to a
terminal, it returns a null pointer. The correct procedure for determining the
login name is to call \fBcuserid\fR(3C), or to call \fBgetlogin()\fR and if it
fails to call \fBgetpwuid\fR(3C).
.sp
.LP

The \fBgetlogin_r()\fR function has the same functionality as \fBgetlogin()\fR
except that the caller must supply a buffer \fIname\fR with length \fInamelen\fR
to store the result.  The \fIname\fR buffer should be at least
\fBLOGIN_NAME_MAX\fR bytes in size (defined in <\fBlimits.h\fR>). The POSIX
version (see \fBstandards\fR(5)) of \fBgetlogin_r()\fR takes a \fInamesize\fR
parameter of type \fBsize_t\fR. If the size of the supplied buffer is less than
the size of \fBLOGIN_NAME_MAX\fR and the name, including the null
terminator, does not fit inside the buffer, than an error will be generated.
Otherwise, the buffer \fIname\fR will be updated with the login name.

.SH RETURN VALUES
.sp
.LP
Upon successful completion, \fBgetlogin()\fR returns a pointer to the login
name or a null pointer if the user’s login name cannot be found.  Otherwise it
returns a null pointer and sets \fBerrno\fR to indicate the error.
.sp
.LP
```

```
121 The standard-conforming \fBgetlogin_r()\fR returns \fB0\fR if successful, or
122 the error number upon failure.
123 .SH ERRORS
131 .sp
132 .LP
124 The \fBgetlogin_r()\fR function will fail if:
125 .sp
126 .ne 2
127 .na
128 \fB\fBERANGE\fR\fR
129 .ad
130 .RS 10n
131 The size of the buffer is smaller than the result to be returned.
132 .RE

134 .sp
135 .ne 2
136 .na
137 \fB\fBEINVAL\fR\fR
138 .ad
139 .RS 10n
140 And entry for the current user was not found in the \fB/var/adm/utmpx\fR file.
141 .RE

143 .sp
144 .LP
145 The \fBgetlogin()\fR and \fBgetlogin_r()\fR functions may fail if:
146 .sp
147 .ne 2
148 .na
149 \fB\fBEMFILE\fR\fR
150 .ad
151 .RS 10n
152 There are {\fBOPEN_MAX\fR} file descriptors currently open in the calling
153 process.
154 .RE

156 .sp
157 .ne 2
158 .na
159 \fB\fBENFILE\fR\fR
160 .ad
161 .RS 10n
162 The maximum allowable number of files is currently open in the system.
163 .RE

165 .sp
166 .ne 2
167 .na
168 \fB\fBENXIO\fR\fR
169 .ad
170 .RS 10n
171 The calling process has no controlling terminal.
172 .RE

174 .sp
175 .LP
176 The \fBgetlogin_r()\fR function may fail if:
177 .sp
178 .ne 2
179 .na
180 \fB\fBERANGE\fR\fR
181 .ad
182 .RS 10n
183 The size of the buffer is smaller than the result to be returned.
184 .RE
```



```
186 .SH USAGE
196 .sp
197 .LP
187 The return value of \fBgetlogin()\fR points to thread-specific data whose
188 content is overwritten on each call by the same thread.
189 .sp
190 .LP
191 Three names associated with the current process can be determined:
192 \fBgetpwuid(\fR\fBgeteuid()\fR\fB)\fR returns the name associated with the
193 effective user ID of the process; \fBgetlogin()\fR returns the name associated
194 with the current login activity; and \fBgetpwuid(\fR\fBgetuid()\fR\fB)\fR
195 returns the name associated with the real user ID of the process.
196 .SH FILES
208 .sp
197 .ne 2
198 .na
199 \fB\fB/var/adm/utmpx\fR\fR
200 .ad
201 .RS 18n
202 user access and administration information
203 .RE

205 .sp
206 .ne 2
207 .na
208 \fB\fB/usr/lib/getloginx.so.1\fR\fR
209 .ad
210 .RS 18n
211 A compatibility library that returns long login names to older applications.
212 .RE

214 .sp
215 .ne 2
216 .na
217 \fB\fB/usr/lib/64/getloginx.so.1\fR\fR
218 .ad
219 .RS 18n
220 A 64-bit compatibility library to return long login names.
221 .RE

223 .SH ATTRIBUTES
236 .sp
237 .LP
224 See \fBattributes\fR(5) for descriptions of the following attributes:
225 .sp

227 .sp
228 .TS
229 box;
230 c | c
231 l | l .
232 ATTRIBUTE TYPE  ATTRIBUTE VALUE
233 _
234 Interface Stability     Standard
235 _
236 MT-Level        See below.
237 .TE

239 .SH SEE ALSO
254 .sp
255 .LP
240 \fBgeteuid\fR(2), \fBgetuid\fR(2), \fBcuserid\fR(3C), \fBgetgrnam\fR(3C),
241 \fBgetpwnam\fR(3C), \fBgetpwuid\fR(3C), \fButmpx\fR(4), \fBattributes\fR(5),
242 \fBstandards\fR(5)
243 .SH NOTES
```

```
260 .sp
261 .LP
244 When compiling multithreaded programs, see \fBIntro\fR(3).
245 .sp
246 .LP
247 The \fBgetlogin()\fR function is safe to use in multithreaded applications, but
248 is discouraged. The \fBgetlogin_r()\fR function should be used instead.
249 .sp
250 .LP
251 Solaris 2.4 and earlier releases provided a \fBgetlogin_r()\fR as specified in
252 POSIX.1c Draft 6. The final POSIX.1c standard changed the interface as
253 described above. Support for the Draft 6 interface is provided for
254 compatibility only and may not be supported in future releases. New
255 applications and libraries should use the standard-conforming interface.
```

```
**********************************************************
    5341 Mon Aug 26 06:56:03 2019
new/usr/src/man/man3c/nanosleep.3c
11622 clean up rarer mandoc lint warnings
**********************************************************
.\"
.\" 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 2016 Joyent, Inc.
.\"
.TH NANOSLEEP 3C "Mar 27, 2016"
.SH NAME
nanosleep, thrd_sleep \- high resolution sleep
.SH SYNOPSIS
.LP
.nf
#include <time.h>

\fBint\fR \fBnanosleep\fR(\fBconst struct timespec *\fR\fIrqtp\fR,
     \fBstruct timespec *\fR\fIrmtp\fR);
.fi

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



```
\fBint\fR \fBthrd_sleep\fR(\fBconst struct timespec *\fR\fIrqtp\fR,
     \fBstruct timespec *\fR\fIrmtp\fR);

.SH DESCRIPTION
.LP
The \fBnanosleep()\fR and \fBthrd_sleep()\fR functions cause the current thread
to be suspended from execution until either the time interval specified by the
\fIrqtp\fR argument has elapsed or a signal is delivered to the calling thread
and its action is to invoke a signal-catching function or to terminate the
process. The suspension time may be longer than requested because the argument
value is rounded up to an integer multiple of the sleep resolution or because of
the scheduling of other activity by the system. But, except for the case of
being interrupted by a signal, the suspension time will not be less than the
time specified by \fIrqtp\fR, as measured by the system clock,
\fBCLOCK_REALTIME\fR.
.sp
.LP
The use of the \fBnanosleep()\fR and \fBthrd_sleep()\fR functions has no effect
on the action or blockage of any signal.
.SH RETURN VALUES
.LP
If the \fBnanosleep()\fR or \fBthrd_sleep()\fR function returns because the
requested time has elapsed, its return value is \fB0\fR.
.sp
.LP
If the \fBnanosleep()\fR function returns because it has been interrupted by a
signal, the function returns a value of \(mi1 and sets \fBerrno\fR to indicate
the interruption. If the \fIrmtp\fR argument is non-\fINULL\fR, the
\fBtimespec\fR structure referenced by it is updated to contain the amount of
time remaining in the interval (the requested time minus the time actually
slept). If the \fIrmtp\fR argument is \fINULL\fR, the remaining time is not
returned.
.sp
.LP
If \fBnanosleep()\fR fails, it returns \fB\(mi1\fR and sets \fBerrno\fR to
indicate the error.
.sp
.LP

The
.B thrd_sleep()
function may fail for identical reasons as the
.B nanosleep()
function and returns \(mi1; however, the C11 standard does not define that
.B errno
should be set, therefore callers of \fBthrd_sleep()\fR cannot rely on
.B errno
being set or staying the same across a call to
.B thrd_sleep() .

.SH ERRORS
.LP
The \fBnanosleep()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBEINTR\fR\fR
.ad
.RS 10n
The \fBnanosleep()\fR function was interrupted by a signal.
.RE

.sp
.ne 2
.na
```

```
123 \fB\fBEINVAL\fR\fR
124 .ad
125 .RS 10n
126 The \fIrqtp\fR argument specified a nanosecond value less than zero or greater
127 than or equal to 1000 million.
128 .RE

130 .sp
131 .ne 2
132 .na
133 \fB\fBENOSYS\fR\fR
134 .ad
135 .RS 10n
136 The \fBnanosleep()\fR function is not supported by this implementation.
137 .RE

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

143 .sp
144 .TS
145 box;
146 c | c
147 l | l .
148 ATTRIBUTE TYPE  ATTRIBUTE VALUE
149 _
150 Interface Stability     Committed
151 _
152 MT-Level        MT-Safe
153 _
154 Standard        See \fBstandards\fR(5).
155 .TE

157 .SH SEE ALSO
164 .LP
158 \fBsleep\fR(3C), \fBtime.h\fR(3HEAD), \fBattributes\fR(5), \fBstandards\fR(5)
```

```
**********************************************************
    7648 Mon Aug 26 06:56:04 2019
new/usr/src/man/man3c/pfmt.3c
11622 clean up rarer mandoc lint warnings
**********************************************************
'\" te
.\"  Copyright 1989 AT&T  Copyright (c) 1997, Sun Microsystems, Inc.  All Rights
.\" 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 PFMT 3C "Dec 29, 1996"
.SH NAME
pfmt \- display error message in standard format
.SH SYNOPSIS
.LP
.nf
#include <pfmt.h>

\fBint\fR \fBpfmt\fR(\fBFILE *\fR\fIstream\fR, \fBlong\fR \fIflags\fR, \fBchar *
.fi

.SH DESCRIPTION
.sp
.LP
The \fBpfmt()\fR retrieves a format string from a locale-specific message
database (unless \fBMM_NOGET\fR is specified) and uses it for \fBprintf\fR(3C)
style formatting of \fIargs\fR. The output is displayed on \fIstream\fR.
.sp
.LP
The \fBpfmt()\fR function encapsulates the output in the standard error message
format (unless \fBMM_NOSTD\fR is specified, in which case the output is similar
to \fBprintf()\fR).
.sp
.LP
If the \fBprintf()\fR format string is to be retrieved from a message database,
the \fBformat\fR argument must have the following structure:
.sp
.LP
\fI<catalog>\fR\fB:\fR\fI<msgnum>\fR\fB:\fR\fI<defmsg>\fR\&.
.sp
.LP
If \fBMM_NOGET\fR is specified, only the \fIdefmsg\fR field must be specified.
.sp
.LP
The \fIcatalog\fR field is used to indicate the message database that contains
the localized version of the format string. This field must be limited to 14
characters selected from the set of all characters values, excluding \fB\e0\fR
(null) and the ASCII codes for \fB/\fR (slash) and \fB:\fR (colon).
.sp
.LP
The \fImsgnum\fR field is a positive number that indicates the index of the
string into the message database.
.sp
.LP
If the catalog does not exist in the locale (specified by the last call to
\fBsetlocale\fR(3C) using the \fBLC_ALL\fR or \fBLC_MESSAGES\fR categories), or
if the message number is out of bound, \fBpfmt()\fR will attempt to retrieve
the message from the C locale. If this second retrieval fails, \fBpfmt()\fR
uses the \fIdefmsg\fR field of the \fBformat\fR argument.
.sp
.LP
If \fIcatalog\fR is omitted, \fBpfmt()\fR will attempt to retrieve the string
from the default catalog specified by the last call to \fBsetcat\fR(3C). In
this case, the \fBformat\fR argument has the following structure:
.sp
.LP
```



```
\fB:\fR\fI<msgnum>\fR\fB:\fR\fI<defmsg>\fR\&.
.sp
.LP
The \fBpfmt()\fR will output \fBMessage not found!!\en\fR as format string if
\fIcatalog\fR is not a valid catalog name, if no catalog is specified (either
explicitely or with \fBsetcat()\fR), if \fImsgnum\fR is not a valid number, or
if no message could be retrieved from the message databases and \fIdefmsg\fR
was omitted.
.sp
.LP
The \fIflags\fR argument determine the type of output (such as whether the
\fBformat\fR should be interpreted as is or encapsulated in the standard
message format), and the access to message catalogs to retrieve a localized
version of \fBformat\fR.
.sp
.LP
The \fIflags\fR argument is composed of several groups, and can take the
following values (one from each group):
.sp
.LP
\fIOutput format control\fR
.sp
.ne 2
.na
\fB\fBMM_NOSTD\fR\fR
.ad
.RS 12n
Do not use the standard message format, interpret \fBformat\fR as
\fBprintf()\fR \fBformat\fR. Only \fIcatalog access control flags\fR should be
specified if \fBMM_NOSTD\fR is used; all other flags will be ignored.
.RE

.sp
.ne 2
.na
\fB\fBMM_STD\fR\fR
.ad
.RS 12n
Output using the standard message format (default value 0).
.RE

.sp
.LP
\fICatalog access control\fR
.sp
.ne 2
.na
\fB\fBMM_NOGET\fR\fR
.ad
.RS 12n
Do not retrieve a localized version of \fBformat\fR. In this case, only the
\fIdefmsg\fR field of the \fBformat\fR is specified.
.RE

.sp
.ne 2
.na
\fB\fBMM_GET\fR\fR
.ad
.RS 12n
Retrieve a localized version of \fBformat\fR from the \fIcatalog\fR, using
\fImsgid\fR as the index and \fIdefmsg\fR as the default message (default value
0).
.RE

.sp
```

```
.LP
\fISeverity (standard message format only)\fR
.sp
.ne 2
.na
\fB\fBMM_HALT\fR\fR
.ad
.RS 14n
Generate a localized version of \fBHALT,\fR but do not halt the machine.
.RE

.sp
.ne 2
.na
\fB\fBMM_ERROR\fR\fR
.ad
.RS 14n
Generate a localized version of \fBERROR\fR (default value 0).
.RE

.sp
.ne 2
.na
\fB\fBMM_WARNING\fR\fR
.ad
.RS 14n
Generate a localized version of \fBWARNING.\fR
.RE

.sp
.ne 2
.na
\fB\fBMM_INFO\fR\fR
.ad
.RS 14n
Generate a localized version of \fBINFO.\fR
.RE

.sp
.LP
Additional severities can be defined. Add-on severities can be defined with
number-string pairs with numeric values from the range [5-255], using
\fBaddsev\fR(3C). The specified severity will be generated from the bitwise
\fBOR\fR operation of the numeric value and other \fIflags\fR If the severity
is not defined, \fBpfmt()\fR uses the string \fBSEV=\fR\fIN\fR, where \fIN\fR
is replaced by the integer severity value passed in \fIflags\fR.
.sp
.LP
Multiple severities passed in \fIflags\fR will not be detected as an error. Any
combination of severities will be summed and the numeric value will cause the
display of either a severity string (if defined) or the string
\fBSEV=\fR\fIN\fR (if undefined).
.sp
.LP
\fIAction\fR
.sp
.ne 2
.na
\fB\fBMM_ACTION\fR\fR
.ad
.RS 13n
Specify an action message. Any severity value is superseded and replaced by a
localized version of \fBTO FIX\fR.
.RE

.SH STANDARD ERROR MESSAGE FORMAT
```



```
.sp
.LP
The \fBpfmt()\fR function displays error messages in the following format:
.sp
.in +2
.nf
\fIlabel\fR: \fIseverity\fR: \fItext\fR
.fi
.in -2

.sp
.LP
If no \fIlabel\fR was defined by a call to \fBsetlabel\fR(3C), the message is
displayed in the format:
.sp
.in +2
.nf
\fIseverity\fR: \fItext\fR
.fi
.in -2

.sp
.LP
If \fBpfmt()\fR is called twice to display an error message and a helpful
\fIaction\fR or recovery message, the output can look like:
.sp
.in +2
.nf
\fIlabel\fR: \fIseverity\fR: \fItext\fR\fIlabel\fR: TO FIX: \fItext\fR
.fi
.in -2

.br
.in +2

.in -2
.br
.in +2

.in -2
.SH RETURN VALUES
.sp
.LP
Upon success, \fBpfmt()\fR returns the number of bytes transmitted. Upon
failure, it returns a negative value:
.sp
.ne 2
.na
\fB\fB\(mi1\fR\fR
.ad
.RS 9n
Write error to \fIstream\fR.
.RE

.SH EXAMPLES
.LP
\fBExample 1 \fRExample of \fBpfmt()\fR function.
.sp
.LP
Example 1:

.sp
.in +2
.nf
setlabel("UX:test");
pfmt(stderr, MM_ERROR, "test:2:Cannot open file: %s\en",
```

```
 244      strerror(errno));

 246 displays the message:

 248 UX:test: ERROR: Cannot open file: No such file or directory
 249 .fi
 250 .in -2

 252 .sp
 253 .LP
 254 Example 2:

 256 .sp
 257 .in +2
 258 .nf
 259 setlabel("UX:test");
 260 setcat("test");
 261 pfmt(stderr, MM_ERROR, ":10:Syntax error\en");
 262 pfmt(stderr, MM_ACTION, "55:Usage ...\en");
 263 .fi
 264 .in -2

 266 .sp
 267 .LP
 268 displays the message

 270 .sp
 271 .in +2
 272 .nf
 273 UX:test: ERROR: Syntax error
 274 UX:test: TO FIX: Usage ...
 275 .fi
 276 .in -2

 278 .SH USAGE
 295 .sp
 296 .LP
 279 Since it uses \fBgettxt\fR(3C), \fBpfmt()\fR should not be used.
 280 .SH ATTRIBUTES
 299 .sp
 300 .LP
 281 See \fBattributes\fR(5) for descriptions of the following attributes:
 282 .sp

 284 .sp
 285 .TS
 286 box;
 287 c | c
 288 l | l .
 289 ATTRIBUTE TYPE  ATTRIBUTE VALUE
 290 _
 291 MT-Level        MT-safe
 292 .TE

 294 .SH SEE ALSO
 315 .sp
 316 .LP
 295 \fBaddsev\fR(3C), \fBgettxt\fR(3C), \fBlfmt\fR(3C), \fBprintf\fR(3C),
 296 \fBsetcat\fR(3C), \fBsetlabel\fR(3C), \fBsetlocale\fR(3C), \fBattributes\fR(5),
 297 \fBenviron\fR(5)
```

```
**********************************************************
   34519 Mon Aug 26 06:56:04 2019
new/usr/src/man/man3lib/libc.3lib
11622 clean up rarer mandoc lint warnings
**********************************************************
```

```
'\" te
.\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
.\" Copyright (c) 2009, Sun Microsystems, Inc.  All rights reserved.
.\" Copyright 2016 Joyent, Inc.
.\" 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
.\" Copyright 2011 by Delphix.  All rights reserved.
.TH LIBC 3LIB "Dec 10, 2015"
.SH NAME
libc \- C library
.SH DESCRIPTION
.LP
Functions in this library provide various facilities defined by System V, ANSI
C, POSIX, and so on.  See \fBstandards\fR(5). In addition, those facilities
previously defined in the internationalization and the wide-character libraries
are now defined in this library, as are the facilities previously defined in
the multithreading libraries, \fBlibthread\fR and \fBlibpthread\fR.
.SH INTERFACES
.LP
The shared object \fBlibc.so.1\fR provides the public interfaces defined below.
See \fBIntro\fR(3) for additional information on shared object interfaces.
.sp


.sp
.TS
l l
l l .
\fB___loc1\fR  \fB___errno\fR
\fB__builtin_alloca\fR  \fB__ctype\fR
\fB__fbufsize\fR        \fB__flbf\fR
\fB__flt_rounds\fR      \fB__fpending\fR
\fB__fpurge\fR  \fB__freadable\fR
\fB__freading\fR        \fB__fsetlocking\fR
\fB__fwritable\fR       \fB__fwriting\fR
\fB__huge_val\fR        \fB__iob\fR
\fB__loc1\fR    \fB__major\fR
\fB__makedev\fR \fB__mb_cur_max_l\fR
\fB__minor\fR
\fB__nsw_extended_action\fR     \fB__nsw_freeconfig\fR
\fB__nsw_getconfig\fR   \fB__posix_asctime_r\fR
\fB__posix_ctime_r\fR   \fB__posix_getgrgid_r\fR
\fB__posix_getgrnam_r\fR        \fB__posix_getlogin_r\fR
\fB__posix_getpwnam_r\fR        \fB__posix_getpwuid_r\fR
\fB__posix_sigwait\fR   \fB__posix_ttyname_r\fR
\fB__priocntl\fR        \fB__priocntlset\fR
\fB__pthread_cleanup_pop\fR     \fB__pthread_cleanup_push\fR
\fB__sysconf_xpg5\fR    \fB__xpg4\fR
\fB__xpg4_putmsg\fR     \fB__xpg4_putpmsg\fR
\fB_Exit\fR     \fB_altzone\fR
\fB_assert\fR   \fB_cleanup\fR
\fB_ctype\fR    \fB_daylight\fR
\fB_environ\fR  \fB_exit\fR
\fB_exithandle\fR       \fB_filbuf\fR
\fB_flsbuf\fR   \fB_flushlbf\fR
\fB_getdate_err\fR      \fB_getdate_err_addr\fR
\fB_iob\fR      \fB_isnan\fR
\fB_isnand\fR   \fB_lwp_cond_broadcast\fR
\fB_lwp_cond_reltimedwait\fR    \fB_lwp_cond_signal\fR
\fB_lwp_cond_timedwait\fR       \fB_lwp_cond_wait\fR
\fB_lwp_continue\fR     \fB_lwp_info\fR
```



```
\fB_lwp_kill\fR \fB_lwp_mutex_lock\fR
\fB_lwp_mutex_trylock\fR        \fB_lwp_mutex_unlock\fR
\fB_lwp_self\fR \fB_lwp_sema_init\fR
\fB_lwp_sema_post\fR    \fB_lwp_sema_trywait\fR
\fB_lwp_sema_wait\fR    \fB_lwp_suspend\fR
\fB_lwp_suspend2\fR     \fB_modf\fR
\fB_nextafter\fR        \fB_nsc_trydoorcall\fR
\fB_nss_XbyY_buf_alloc\fR       \fB_nss_XbyY_buf_free\fR
\fB_nss_netdb_aliases\fR        \fB_numeric\fR
\fB_scalb\fR    \fB_sibuf\fR
\fB_sobuf\fR    \fB_stack_grow\fR
\fB_sys_buslist\fR      \fB_sys_cldlist\fR
\fB_sys_fpelist\fR      \fB_sys_illlist\fR
\fB_sys_segvlist\fR     \fB_sys_siginfolistp\fR
\fB_sys_siglist\fR      \fB_sys_siglistn\fR
\fB_sys_siglistp\fR     \fB_sys_traplist\fR
\fB_timezone\fR \fB_tolower\fR
\fB_toupper\fR  \fB_tzname\fR
\fB_xftw\fR     \fB\fR
\fBa64l\fR      \fBabort\fR
\fBabs\fR       \fBaccess\fR
\fBacct\fR      \fBacl\fR
\fBaddrtosymstr\fR      \fBaddsev\fR
\fBaddseverity\fR       \fBadjtime\fR
\fBaio_cancel\fR        \fBaio_error\fR
\fBaio_fsync\fR \fBaio_read\fR
\fBaio_return\fR        \fBaio_suspend\fR
\fBaio_waitn\fR \fBaio_write\fR
\fBaiocancel\fR \fBaioread\fR
\fBaiowait\fR   \fBaiowrite\fR
\fBalarm\fR     \fBalphasort\fR
\fBaltzone\fR   \fBascftime\fR
\fBasctime\fR   \fBasctime_r\fR
\fBasprintf\fR
\fBatexit\fR    \fBatof\fR
\fBatoi\fR      \fBatol\fR
\fBatoll\fR     \fBatomic_add_16\fR
\fBatomic_add_16_nv\fR  \fBatomic_add_32\fR
\fBatomic_add_32_nv\fR  \fBatomic_add_64\fR
\fBatomic_add_64_nv\fR  \fBatomic_add_8\fR
\fBatomic_add_8_nv\fR   \fBatomic_add_char\fR
\fBatomic_add_char_nv\fR        \fBatomic_add_int\fR
\fBatomic_add_int_nv\fR \fBatomic_add_long\fR
\fBatomic_add_long_nv\fR        \fBatomic_add_ptr\fR
\fBatomic_add_ptr_nv\fR \fBatomic_add_short\fR
\fBatomic_add_short_nv\fR       \fBatomic_and_16\fR
\fBatomic_and_16_nv\fR  \fBatomic_and_32\fR
\fBatomic_and_32_nv\fR  \fBatomic_and_64\fR
\fBatomic_and_64_nv\fR  \fBatomic_and_8\fR
\fBatomic_and_8_nv\fR   \fBatomic_and_uchar\fR
\fBatomic_and_uchar_nv\fR       \fBatomic_and_uint\fR
\fBatomic_and_uint_nv\fR        \fBatomic_and_ulong\fR
\fBatomic_and_ulong_nv\fR       \fBatomic_and_ushort\fR
\fBatomic_and_ushort_nv\fR      \fBatomic_cas_16\fR
\fBatomic_cas_32\fR     \fBatomic_cas_64\fR
\fBatomic_cas_8\fR      \fBatomic_cas_ptr\fR
\fBatomic_cas_uchar\fR  \fBatomic_cas_uint\fR
\fBatomic_cas_ulong\fR  \fBatomic_cas_ushort\fR
\fBatomic_clear_long_excl\fR    \fBatomic_dec_16\fR
\fBatomic_dec_16_nv\fR  \fBatomic_dec_32\fR
\fBatomic_dec_32_nv\fR  \fBatomic_dec_64\fR
\fBatomic_dec_64_nv\fR  \fBatomic_dec_8\fR
\fBatomic_dec_8_nv\fR   \fBatomic_dec_ptr\fR
\fBatomic_dec_ptr_nv\fR \fBatomic_dec_uchar\fR
\fBatomic_dec_uchar_nv\fR       \fBatomic_dec_uint\fR
\fBatomic_dec_uint_nv\fR        \fBatomic_dec_ulong\fR
```

```
\fBatomic_dec_ulong_nv\fR       \fBatomic_dec_ushort\fR
\fBatomic_dec_ushort_nv\fR      \fBatomic_inc_16\fR
\fBatomic_inc_16_nv\fR  \fBatomic_inc_32\fR
\fBatomic_inc_32_nv\fR  \fBatomic_inc_64\fR
\fBatomic_inc_64_nv\fR  \fBatomic_inc_8\fR
\fBatomic_inc_8_nv\fR   \fBatomic_inc_ptr\fR
\fBatomic_inc_ptr_nv\fR \fBatomic_inc_uchar\fR
\fBatomic_inc_uchar_nv\fR       \fBatomic_inc_uint\fR
\fBatomic_inc_uint_nv\fR        \fBatomic_inc_ulong\fR
\fBatomic_inc_ulong_nv\fR       \fBatomic_inc_ushort\fR
\fBatomic_inc_ushort_nv\fR      \fBatomic_or_16\fR
\fBatomic_or_16_nv\fR   \fBatomic_or_32\fR
\fBatomic_or_32_nv\fR   \fBatomic_or_64\fR
\fBatomic_or_64_nv\fR   \fBatomic_or_8\fR
\fBatomic_or_8_nv\fR    \fBatomic_or_uchar\fR
\fBatomic_or_uchar_nv\fR        \fBatomic_or_uint\fR
\fBatomic_or_uint_nv\fR \fBatomic_or_ulong\fR
\fBatomic_or_ulong_nv\fR        \fBatomic_or_ushort\fR
\fBatomic_or_ushort_nv\fR       \fBatomic_set_long_excl\fR
\fBatomic_swap_16\fR    \fBatomic_swap_32\fR
\fBatomic_swap_64\fR    \fBatomic_swap_8\fR
\fBatomic_swap_ptr\fR   \fBatomic_swap_uchar\fR
\fBatomic_swap_uint\fR  \fBatomic_swap_ulong\fR
\fBatomic_swap_ushort\fR        \fBattropen\fR
\fBbacktrace\fR \fBbacktrace_symbols\fR
\fBbacktrace_symbols_fd\fR      \fBbasename\fR
\fBbcmp\fR      \fBbcopy\fR
\fBbindtextdomain\fR    \fBbind_textdomain_codeset\fR
\fBbrk\fR       \fBbsd_signal\fR
\fBbsearch\fR
\fBbtowc\fR     \fBbtowc_l\fR
\fBbzero\fR     \fBcalloc\fR
\fBcanonicalize_file_name\fR
\fBcatclose\fR  \fBcatgets\fR
\fBcatopen\fR   \fBcfgetispeed\fR
\fBcfgetospeed\fR       \fBcfsetispeed\fR
\fBcfsetospeed\fR       \fBcftime\fR
\fBchdir\fR     \fBchmod\fR
\fBchown\fR     \fBchroot\fR
\fBclearerr\fR  \fBclearenv\fR
\fBclock\fR
\fBclock_getres\fR      \fBclock_gettime\fR
\fBclock_nanosleep\fR   \fBclock_settime\fR
\fBclose\fR     \fBclosedir\fR
\fBclosefrom\fR \fBcloselog\fR
\fBcond_broadcast\fR    \fBcond_destroy\fR
\fBcond_init\fR \fBcond_reltimedwait\fR
\fBcond_signal\fR       \fBcond_timedwait\fR
\fBcond_wait\fR \fBconfstr\fR
\fBcreat\fR     \fBcrypt\fR
\fBcrypt_genhash_impl\fR        \fBcrypt_gensalt\fR
\fBcrypt_gensalt_impl\fR        \fBcsetcol\fR
\fBcsetlen\fR   \fBctermid\fR
\fBctermid_r\fR \fBctime\fR
\fBctime_r\fR   \fBcuserid\fR
\fBdaemon\fR
\fBdaylight\fR  \fBdbm_clearerr\fR
\fBdbm_close\fR \fBdbm_delete\fR
\fBdbm_error\fR \fBdbm_fetch\fR
\fBdbm_firstkey\fR      \fBdbm_nextkey\fR
\fBdbm_open\fR  \fBdbm_store\fR
\fBdcgettext\fR \fBdcngettext\fR
\fBdecimal_to_double\fR \fBdecimal_to_extended\fR
\fBdecimal_to_quadruple\fR      \fBdecimal_to_single\fR
\fBdgettext\fR  \fBdifftime\fR
\fBdirectio\fR  \fBdirfd\fR
```



```
\fBdirname\fR   \fBdiv\fR
\fBdladdr\fR    \fBdladdr1\fR
\fBdlclose\fR   \fBdldump\fR
\fBdlerror\fR   \fBdlinfo\fR
\fBdlmopen\fR   \fBdlopen\fR
\fBdlsym\fR     \fBdngettext\fR
\fBdoor_bind\fR \fBdoor_call\fR
\fBdoor_create\fR       \fBdoor_cred\fR
\fBdoor_getparam\fR     \fBdoor_info\fR
\fBdoor_return\fR       \fBdoor_revoke\fR
\fBdoor_server_create\fR        \fBdoor_setparam\fR
\fBdoor_ucred\fR        \fBdoor_unbind\fR
\fBdouble_to_decimal\fR \fBdrand48\fR
\fBdup\fR       \fBdup2\fR
\fBduplocale\fR
\fBeconvert\fR  \fBecvt\fR
\fBenable_extended_FILE_stdio\fR
\fBencrypt\fR   \fBendgrent\fR
\fBendnetgrent\fR       \fBendpwent\fR
\fBendspent\fR  \fBendusershell\fR
\fBendutent\fR  \fBendutxent\fR
\fBenviron\fR   \fBerand48\fR
\fBerr\fR       \fBerrno\fR
\fBerrx\fR      \fBeuccol\fR
\fBeuclen\fR    \fBeucscol\fR
\fBexecl\fR     \fBexecle\fR
\fBexeclp\fR    \fBexecv\fR
\fBexecve\fR    \fBexecvp\fR
\fBexit\fR      \fBextended_to_decimal\fR
\fBfaccessat\fR
\fBfacl\fR      \fBfattach\fR
\fBfchdir\fR    \fBfchmod\fR
\fBfchmodat\fR
\fBfchown\fR    \fBfchownat\fR
\fBfchroot\fR   \fBfclose\fR
\fBfcloseall\fR \fBfcntl\fR
\fBfconvert\fR  \fBfcvt\fR
\fBfdatasync\fR \fBfdetach\fR
\fBfdopen\fR    \fBfdopendir\fR
\fBfdwalk\fR    \fBfeof\fR
\fBferror\fR    \fBfflush\fR
\fBffs\fR       \fBffsl\fR
\fBffsll\fR     \fBfls\fR
\fBflsl\fR      \fBflsll\fR
\fBfgetattr\fR  \fBfgetc\fR
\fBfgetgrent\fR \fBfgetgrent_r\fR
\fBfgetpos\fR   \fBfgetpwent\fR
\fBfgetpwent_r\fR       \fBfgets\fR
\fBfgetspent\fR \fBfgetspent_r\fR
\fBfgetwc\fR    \fBfgetwc_l\fR
\fBfgetws\fR
\fBfile_to_decimal\fR   \fBfileno\fR
\fBfinite\fR    \fBflockfile\fR
\fBfmtmsg\fR    \fBfnmatch\fR
\fBfopen\fR     \fBfork\fR
\fBfork1\fR     \fBforkall\fR
\fBforkallx\fR  \fBforkx\fR
\fBfpathconf\fR \fBfpclass\fR
\fBfpgetmask\fR \fBfpgetround\fR
\fBfpgetsticky\fR       \fBfprintf\fR
\fBfpsetmask\fR \fBfpsetround\fR
\fBfpsetsticky\fR       \fBfputc\fR
\fBfputs\fR     \fBfputwc\fR
\fBfputws\fR    \fBfread\fR
\fBfree\fR      \fBfreelocale\fR
\fBfreopen\fR
```

```
\fBfrexp\fR     \fBfscanf\fR
\fBfseek\fR     \fBfseeko\fR
\fBfsetattr\fR
\fBfsetpos\fR   \fBfstat\fR
\fBfstatat\fR   \fBfstatfs\fR
\fBfstatvfs\fR  \fBfsync\fR
\fBftell\fR     \fBftello\fR
\fBftime\fR     \fBftok\fR
\fBftruncate\fR \fBftrylockfile\fR
\fBftw\fR       \fBfunc_to_decimal\fR
\fBfunlockfile\fR       \fBfutimens\fR
\fBfutimesat\fR
\fBfwide\fR     \fBfwprintf\fR
\fBfwrite\fR    \fBfwscanf\fR
\fBgconvert\fR  \fBgcvt\fR
\fBgetacct\fR   \fBgetattrat\fR
\fBgetc\fR
\fBgetc_unlocked\fR     \fBgetchar\fR
\fBgetchar_unlocked\fR  \fBgetcontext\fR
\fBgetcpuid\fR  \fBgetcwd\fR
\fBgetdate\fR   \fBgetdate_err\fR
\fBgetdents\fR  \fBgetdtablesize\fR
\fBgetegid\fR   \fBgetenv\fR
\fBgeteuid\fR   \fBgetexecname\fR
\fBgetextmntent\fR      \fBgetgid\fR
\fBgetgrent\fR  \fBgetgrent_r\fR
\fBgetgrgid\fR  \fBgetgrgid_r\fR
\fBgetgrnam\fR  \fBgetgrnam_r\fR
\fBgetgroups\fR \fBgethomelgroup\fR
\fBgethostid\fR \fBgethostname\fR
\fBgethrtime\fR \fBgethrvtime\fR
\fBgetisax\fR   \fBgetitimer\fR
\fBgetloadavg\fR        \fBgetlogin\fR
\fBgetlogin_r\fR        \fBgetmntany\fR
\fBgetmntent\fR \fBgetmsg\fR
\fBget_nprocs\fR        \fBget_nprocs_conf\fR
\fBgetnetgrent\fR       \fBgetnetgrent_r\fR
\fBgetopt\fR    \fBgetopt_clip\fR
\fBgetopt_long\fR       \fBgetopt_long_only\fR
\fBgetpagesize\fR       \fBgetpagesizes\fR
\fBgetpass\fR   \fBgetpassphrase\fR
\fBgetpeerucred\fR      \fBgetpflags\fR
\fBgetpgid\fR   \fBgetpgrp\fR
\fBgetpid\fR    \fBgetpmsg\fR
\fBgetppid\fR   \fBgetppriv\fR
\fBgetpriority\fR       \fBgetprogname\fR
\fIfBgetpriority\fR     \fIfbgetprogname\fR
\fBgetprojid\fR
\fBgetpw\fR     \fBgetpwent\fR
\fBgetpwent_r\fR        \fBgetpwnam\fR
\fBgetpwnam_r\fR        \fBgetpwuid\fR
\fBgetpwuid_r\fR        \fBgetrctl\fR
\fBgetrlimit\fR \fBgetrusage\fR
\fBgets\fR      \fBgetsid\fR
\fBgetspent\fR  \fBgetspent_r\fR
\fBgetspnam\fR  \fBgetspnam_r\fR
\fBgetsubopt\fR \fBgettaskid\fR
\fBgettext\fR   \fBgettimeofday\fR
\fBgettxt\fR    \fBgetuid\fR
\fBgetusershell\fR      \fBgetustack\fR
\fBgetutent\fR  \fBgetutid\fR
\fBgetutline\fR \fBgetutmp\fR
\fBgetutmpx\fR  \fBgetutxent\fR
\fBgetutxid\fR  \fBgetutxline\fR
\fBgetvfsany\fR \fBgetvfsent\fR
\fBgetvfsfile\fR        \fBgetvfsspec\fR
```



```
\fBgetw\fR
\fBgetwc\fR     \fBgetwc_l\fR
\fBgetwchar\fR  \fBgetwchar_l\fR
\fBgetwd\fR
\fBgetwidth\fR  \fBgetws\fR
\fBgetzoneid\fR \fBgetzoneidbyname\fR
\fBgetzonenamebyid\fR   \fBglob\fR
\fBglobfree\fR  \fBgmtime\fR
\fBgmtime_r\fR  \fBgrantpt\fR
\fBgsignal\fR   \fBhasmntopt\fR
\fBhcreate\fR   \fBhdestroy\fR
\fBhsearch\fR   \fBiconv\fR
\fBiconv_close\fR       \fBiconv_open\fR
\fBimaxabs\fR   \fBimaxdiv\fR
\fBindex\fR     \fBinitgroups\fR
\fBinitstate\fR \fBinnetgr\fR
\fBinsque\fR    \fBioctl\fR
\fBis_system_labeled\fR
\fBisaexec\fR
\fBisalnum\fR   \fBisalnum_l\fR
\fBisalpha\fR   \fBisalpha_l\fR
\fBisascii\fR   \fBisastream\fR
\fBisatty\fR
\fBisblank\fR   \fBisblank_l\fR
\fBiscntrl\fR   \fBiscntrl_l\fR
\fBisdigit\fR   \fBisdigit_l\fR
\fBisenglish\fR
\fBisgraph\fR   \fBisgraph_l\fR
\fBisideogram\fR
\fBislower\fR   \fBislower_l\fR
\fBisnan\fR
\fBisnand\fR    \fBisnanf\fR
\fBisnumber\fR  \fBisphonogram\fR
\fBisprint\fR   \fBisprint_l\fR
\fBispunct\fR   \fBispunct_l\fR
\fBissetugid\fR
\fBisspace\fR   \fBisspace_l\fR
\fBisspecial\fR
\fBisupper\fR   \fBisupper_l\fR
\fBiswalnum\fR  \fBiswalnum_l\fR
\fBiswalpha\fR  \fBiswalpha_l\fR
\fBiswblank\fR  \fBiswblank_l\fR
\fBiswcntrl\fR  \fBiswcntrl_l\fR
\fBiswctype\fR  \fBiswctype_l\fR
\fBiswdigit\fR  \fBiswdigit_l\fR
\fBiswideogram\fR       \fBiswideogram_l\fR
\fBiswgraph\fR  \fBiswgraph_l\fR
\fBiswhexnumber\fR      \fBiswhexnumber_l\fR
\fBiswlower\fR  \fBiswlower_l\fR
\fBiswnumber\fR \fBiswnumber_l\fR
\fBiswphonogram\fR      \fBiswphonogram_l\fR
\fBiswprint\fR  \fBiswprint_l\fR
\fBiswpunct\fR  \fBiswpunct_l\fR
\fBiswspace\fR  \fBiswspace_l\fR
\fBiswspecial\fR        \fBiswspecial_l\fR
\fBiswupper\fR  \fBiswupper_l\fR
\fBiswxdigit\fR \fBiswxdigit_l\fR
\fBisxdigit\fR  \fBisxdigit_l\fR
\fBjrand48\fR   \fBkill\fR
\fBkillpg\fR    \fBl64a\fR
\fBlabs\fR      \fBladd\fR
\fBlchown\fR    \fBlckpwdf\fR
\fBlcong48\fR   \fBldexp\fR
\fBldivide\fR   \fBlexp10\fR
\fBlfind\fR     \fBlfmt\fR
\fBlink\fR      \fBlinkat\fR
```

```
\fBlio_listio\fR        \fBllabs\fR
\fBlldiv\fR     \fBllog10\fR
\fBllseek\fR    \fBlltostr\fR
\fBlocaleconv\fR        \fBlocaltime\fR
\fBlocaltime_r\fR       \fBlockf\fR
\fBlogb\fR      \fBlone\fR
\fBlongjmp\fR   \fBlrand48\fR
\fBlsearch\fR   \fBlseek\fR
\fBlshiftl\fR   \fBlstat\fR
\fBlsub\fR      \fBlten\fR
\fBlzero\fR     \fBmadvise\fR
\fBmakecontext\fR       \fBmakeutx\fR
\fBmalloc\fR
\fBmblen\fR     \fBmblen_l\fR
\fBmbrlen\fR    \fBmbrlen_l\fR
\fBmbrtowc\fR   \fBmbrtowc_l\fR
\fBmbsinit\fR   \fBmbsinit_l\fR
\fBmbsnrtowcs\fR        \fBmbsnrtowcs_l\fR
\fBmbsrtowcs\fR \fBmbsrtowcs_l\fR
\fBmbstowcs\fR  \fBmbstowcs_l\fR
\fBmbtowc\fR    \fBmbtowc_l\fR
\fBmemalign\fR  \fBmembar_consumer\fR
\fBmembar_enter\fR      \fBmembar_exit\fR
\fBmembar_producer\fR   \fBmemccpy\fR
\fBmemchr\fR    \fBmemcmp\fR
\fBmemcntl\fR   \fBmemcpy\fR
\fBmeminfo\fR   \fBmemmem\fR
\fBmemmove\fR
\fBmemset\fR    \fBmincore\fR
\fBmkdir\fR     \fBmkdirat\fR
\fBmkfifo\fR    \fBmkfifoat\fR
\fBmknod\fR     \fBmknodat\fR
\fBmkstemp\fR
\fBmktemp\fR    \fBmktime\fR
\fBmlock\fR     \fBmlockall\fR
\fBmmap\fR      \fBmmapobj\fR
\fBmodctl\fR
\fBmodf\fR      \fBmodff\fR
\fBmodutx\fR    \fBmonitor\fR
\fBmount\fR     \fBmprotect\fR
\fBmq_close\fR  \fBmq_getattr\fR
\fBmq_notify\fR \fBmq_open\fR
\fBmq_receive\fR        \fBmq_reltimedreceive_np\fR
\fBmq_reltimedsend_np\fR        \fBmq_send\fR
\fBmq_setattr\fR        \fBmq_timedreceive\fR
\fBmq_timedsend\fR      \fBmq_unlink\fR
\fBmrand48\fR   \fBmsgctl\fR
\fBmsgget\fR    \fBmsgids\fR
\fBmsgrcv\fR    \fBmsgsnap\fR
\fBmsgsnd\fR    \fBmsync\fR
\fBmunlock\fR   \fBmunlockall\fR
\fBmunmap\fR    \fBmutex_consistent\fR
\fBmutex_destroy\fR     \fBmutex_init\fR
\fBmutex_lock\fR        \fBmutex_trylock\fR
\fBmutex_unlock\fR      \fBnanosleep\fR
\fBnextafter\fR \fBnewlocale\fR
\fBnfs_getfh\fR \fBnftw\fR
\fBngettext\fR  \fBnice\fR
\fBnl_langinfo\fR       \fBnl_langinfo_l\fR
\fBnrand48\fR
\fBnss_default_finders\fR       \fBnss_delete\fR
\fBnss_endent\fR        \fBnss_getent\fR
\fBnss_search\fR        \fBnss_setent\fR
\fBntp_adjtime\fR       \fBntp_gettime\fR
\fBopen\fR      \fBopenat\fR
\fBopendir\fR   \fBopenlog\fR
```



```
\fBoptarg\fR    \fBopterr\fR
\fBoptind\fR    \fBoptopt\fR
\fBp_online\fR  \fBpathconf\fR
\fBpause\fR     \fBpclose\fR
\fBpcsample\fR  \fBperror\fR
\fBpfmt\fR      \fBpipe\fR
\fBplock\fR     \fBpoll\fR
\fBpopen\fR     \fBport_alert\fR
\fBport_associate\fR    \fBport_create\fR
\fBport_dissociate\fR   \fBport_get\fR
\fBport_getn\fR \fBport_send\fR
\fBport_sendn\fR        \fBposix_fadvise\fR
\fBposix_fallocate\fR   \fBposix_madvise\fR
\fBposix_memalign\fR    \fBposix_openpt\fR
\fBposix_spawn\fR       \fBposix_spawn_file_actions_addclose\fR
\fBposix_spawn_file_actions_addclosefrom_np\fR  \fBposix_spawn_file_actions_addd
\fBposix_spawn_file_actions_addopen\fR
\fBposix_spawn_file_actions_destroy\fR
\fBposix_spawn_file_actions_init\fR
\fBposix_spawn_pipe_np\fR
\fBposix_spawnattr_destroy\fR
\fBposix_spawnattr_getflags\fR  \fBposix_spawnattr_getpgroup\fR
\fBposix_spawnattr_getschedparam\fR     \fBposix_spawnattr_getschedpolicy\fR
\fBposix_spawnattr_getsigdefault\fR     \fBposix_spawnattr_getsigignore_np\fR
\fBposix_spawnattr_getsigmask\fR        \fBposix_spawnattr_init\fR
\fBposix_spawnattr_setflags\fR  \fBposix_spawnattr_setpgroup\fR
\fBposix_spawnattr_setschedparam\fR     \fBposix_spawnattr_setschedpolicy\fR
\fBposix_spawnattr_setsigdefault\fR     \fBposix_spawnattr_setsigignore_np\fR
\fBposix_spawnattr_setsigmask\fR        \fBposix_spawnp\fR
\fBpread\fR     \fBpreadv\fR
\fBprintf\fR
\fBprintstack\fR        \fBpriocntl\fR
\fBpriocntlset\fR       \fBpriv_addset\fR
\fBpriv_allocset\fR     \fBpriv_copyset\fR
\fBpriv_delset\fR       \fBpriv_emptyset\fR
\fBpriv_fillset\fR      \fBpriv_freeset\fR
\fBpriv_getbyname\fR    \fBpriv_getbynum\fR
\fBpriv_getsetbyname\fR \fBpriv_getsetbynum\fR
\fBpriv_gettext\fR      \fBpriv_ineffect\fR
\fBpriv_intersect\fR    \fBpriv_inverse\fR
\fBpriv_isemptyset\fR   \fBpriv_isequalset\fR
\fBpriv_isfullset\fR    \fBpriv_ismember\fR
\fBpriv_issubset\fR     \fBpriv_set\fR
\fBpriv_set_to_str\fR   \fBpriv_str_to_set\fR
\fBpriv_union\fR        \fBprocessor_bind\fR
\fBprocessor_info\fR    \fBprofil\fR
\fBpselect\fR   \fBpset_assign\fR
\fBpset_bind\fR \fBpset_create\fR
\fBpset_destroy\fR      \fBpset_getattr\fR
\fBpset_getloadavg\fR   \fBpset_info\fR
\fBpset_list\fR \fBpset_setattr\fR
\fBpsiginfo\fR  \fBpsignal\fR
\fBpthread_atfork\fR    \fBpthread_attr_destroy\fR
\fBpthread_attr_get_np\fR
\fBpthread_attr_getdetachstate\fR       \fBpthread_attr_getguardsize\fR
\fBpthread_attr_getinheritsched\fR      \fBpthread_attr_getschedparam\fR
\fBpthread_attr_getschedpolicy\fR       \fBpthread_attr_getscope\fR
\fBpthread_attr_getstack\fR     \fBpthread_attr_getstackaddr\fR
\fBpthread_attr_getstacksize\fR \fBpthread_attr_init\fR
\fBpthread_attr_setdetachstate\fR       \fBpthread_attr_setguardsize\fR
\fBpthread_attr_setinheritsched\fR      \fBpthread_attr_setschedparam\fR
\fBpthread_attr_setschedpolicy\fR       \fBpthread_attr_setscope\fR
\fBpthread_attr_setstack\fR     \fBpthread_attr_setstackaddr\fR
\fBpthread_attr_setstacksize\fR \fBpthread_barrier_destroy\fR
\fBpthread_barrier_init\fR      \fBpthread_barrier_wait\fR
\fBpthread_barrierattr_destroy\fR       \fBpthread_barrierattr_getpshared\fR
```

```
\fBpthread_barrierattr_init\fR  \fBpthread_barrierattr_setpshared\fR
\fBpthread_cancel\fR    \fBpthread_cond_broadcast\fR
\fBpthread_cond_destroy\fR      \fBpthread_cond_init\fR
\fBpthread_cond_reltimedwait_np\fR      \fBpthread_cond_signal\fR
\fBpthread_cond_timedwait\fR    \fBpthread_cond_wait\fR
\fBpthread_condattr_destroy\fR  \fBpthread_condattr_getclock\fR
\fBpthread_condattr_getpshared\fR       \fBpthread_condattr_init\fR
\fBpthread_condattr_setclock\fR \fBpthread_condattr_setpshared\fR
\fBpthread_create\fR    \fBpthread_detach\fR
\fBpthread_equal\fR     \fBpthread_exit\fR
\fBpthread_getconcurrency\fR    \fBpthread_getschedparam\fR
\fBpthread_getspecific\fR       \fBpthread_join\fR
\fBpthread_key_create\fR        \fBpthread_key_create_once_np\fR
\fBpthread_key_delete\fR
\fBpthread_kill\fR      \fBpthread_mutex_consistent\fR
\fBpthread_mutex_destroy\fR     \fBpthread_mutex_getprioceiling\fR
\fBpthread_mutex_init\fR        \fBpthread_mutex_lock\fR
\fBpthread_mutex_reltimedlock_np\fR     \fBpthread_mutex_setprioceiling\fR
\fBpthread_mutex_timedlock\fR   \fBpthread_mutex_trylock\fR
\fBpthread_mutex_unlock\fR      \fBpthread_mutexattr_destroy\fR
\fBpthread_mutexattr_getprioceiling\fR  \fBpthread_mutexattr_getprotocol\fR
\fBpthread_mutexattr_getpshared\fR      \fBpthread_mutexattr_getrobust\fR
\fBpthread_mutexattr_gettype\fR \fBpthread_mutexattr_init\fR
\fBpthread_mutexattr_setprioceiling\fR  \fBpthread_mutexattr_setprotocol\fR
\fBpthread_mutexattr_setpshared\fR      \fBpthread_mutexattr_setrobust\fR
\fBpthread_mutexattr_settype\fR \fBpthread_once\fR
\fBpthread_rwlock_destroy\fR    \fBpthread_rwlock_init\fR
\fBpthread_rwlock_rdlock\fR     \fBpthread_rwlock_reltimedrdlock_np\fR
\fBpthread_rwlock_reltimedwrlock_np\fR  \fBpthread_rwlock_timedrdlock\fR
\fBpthread_rwlock_timedwrlock\fR        \fBpthread_rwlock_tryrdlock\fR
\fBpthread_rwlock_trywrlock\fR  \fBpthread_rwlock_unlock\fR
\fBpthread_rwlock_wrlock\fR     \fBpthread_rwlockattr_destroy\fR
\fBpthread_rwlockattr_getpshared\fR     \fBpthread_rwlockattr_init\fR
\fBpthread_rwlockattr_setpshared\fR     \fBpthread_self\fR
\fBpthread_setcancelstate\fR    \fBpthread_setcanceltype\fR
\fBpthread_setconcurrency\fR    \fBpthread_setspecific\fR
\fBpthread_sigmask\fR   \fBpthread_setschedparam\fR
\fBpthread_setschedprio\fR      \fBpthread_spin_destroy\fR
\fBpthread_spin_init\fR \fBpthread_spin_lock\fR
\fBpthread_spin_trylock\fR      \fBpthread_spin_unlock\fR
\fBpthread_testcancel\fR        \fBptsname\fR
\fBputacct\fR   \fBputc\fR
\fBputc_unlocked\fR     \fBputchar\fR
\fBputchar_unlocked\fR  \fBputenv\fR
\fBputmsg\fR    \fBputpmsg\fR
\fBputpwent\fR  \fBputs\fR
\fBputspent\fR  \fBpututline\fR
\fBpututxline\fR        \fBputw\fR
\fBputwc\fR     \fBputwchar\fR
\fBputws\fR     \fBpwrite\fR
\fBpwritev\fR
\fBqeconvert\fR \fBqecvt\fR
\fBqfconvert\fR \fBqfcvt\fR
\fBqgconvert\fR \fBqgcvt\fR
\fBqsort\fR     \fBquadruple_to_decimal\fR
\fBraise\fR     \fBrand\fR
\fBrand_r\fR    \fBrandom\fR
\fBrctl_walk\fR \fBrctlblk_get_enforced_value\fR
\fBrctlblk_get_firing_time\fR   \fBrctlblk_get_global_action\fR
\fBrctlblk_get_global_flags\fR  \fBrctlblk_get_local_action\fR
\fBrctlblk_get_local_flags\fR   \fBrctlblk_get_privilege\fR
\fBrctlblk_get_recipient_pid\fR \fBrctlblk_get_value\fR
\fBrctlblk_set_local_action\fR  \fBrctlblk_set_local_flags\fR
\fBrctlblk_set_privilege\fR     \fBrctlblk_set_recipient_pid\fR
\fBrctlblk_set_value\fR \fBrctlblk_size\fR
\fBre_comp\fR   \fBre_exec\fR
```



```
\fBread\fR      \fBreaddir\fR
\fBreaddir_r\fR \fBreadlink\fR
\fBreadlinkat\fR
\fBreadv\fR     \fBrealloc\fR
\fBrealpath\fR  \fBreboot\fR
\fBregcmp\fR    \fBregcomp\fR
\fBregerror\fR  \fBregex\fR
\fBregexec\fR   \fBregfree\fR
\fBremove\fR    \fBremque\fR
\fBrename\fR    \fBrenameat\fR
\fBresetmnttab\fR       \fBresolvepath\fR
\fBrewind\fR    \fBrewinddir\fR
\fBrindex\fR    \fBrmdir\fR
\fBrw_rdlock\fR \fBrw_read_held\fR
\fBrw_tryrdlock\fR      \fBrw_trywrlock\fR
\fBrw_unlock\fR \fBrw_write_held\fR
\fBrw_wrlock\fR \fBrwlock_destroy\fR
\fBrwlock_init\fR       \fBsbrk\fR
\fBscalb\fR     \fBscandir\fR
\fBscanf\fR     \fBsched_get_priority_max\fR
\fBsched_get_priority_min\fR    \fBsched_getparam\fR
\fBsched_getscheduler\fR        \fBsched_rr_get_interval\fR
\fBsched_setparam\fR    \fBsched_setscheduler\fR
\fBsched_yield\fR       \fBschedctl_exit\fR
\fBschedctl_init\fR     \fBschedctl_lookup\fR
\fBschedctl_start\fR    \fBschedctl_stop\fR
\fBseconvert\fR \fBseed48\fR
\fBseekdir\fR   \fBselect\fR
\fBsem_close\fR \fBsem_destroy\fR
\fBsem_getvalue\fR      \fBsem_init\fR
\fBsem_open\fR  \fBsem_post\fR
\fBsem_reltimedwait_np\fR       \fBsem_timedwait\fR
\fBsem_trywait\fR       \fBsem_unlink\fR
\fBsem_wait\fR
\fBsema_destroy\fR      \fBsema_held\fR
\fBsema_init\fR \fBsema_post\fR
\fBsema_trywait\fR      \fBsema_wait\fR
\fBsemctl\fR    \fBsemget\fR
\fBsemids\fR    \fBsemop\fR
\fBsemtimedop\fR        \fBsetattrat\fR
\fBsetbuf\fR
\fBsetbuffer\fR \fBsetcat\fR
\fBsetcontext\fR        \fBsetegid\fR
\fBsetenv\fR    \fBseteuid\fR
\fBsetgid\fR    \fBsetgrent\fR
\fBsetgroups\fR \fBsethostname\fR
\fBsetitimer\fR \fBsetjmp\fR
\fBsetkey\fR    \fBsetlabel\fR
\fBsetlinebuf\fR        \fBsetlocale\fR
\fBsetlogmask\fR        \fBsetnetgrent\fR
\fBsetpflags\fR \fBsetpgid\fR
\fBsetpgrp\fR   \fBsetppriv\fR
\fBsetpriority\fR       \fBsetprogname\fR
\fBsetpwent\fR
\fBsetrctl\fR   \fBsetregid\fR
\fBsetreuid\fR  \fBsetrlimit\fR
\fBsetsid\fR    \fBsetspent\fR
\fBsetstate\fR  \fBsettaskid\fR
\fBsettimeofday\fR      \fBsetuid\fR
\fBsetusershell\fR      \fBsetustack\fR
\fBsetutent\fR  \fBsetutxent\fR
\fBsetvbuf\fR   \fBsfconvert\fR
\fBsgconvert\fR \fBshm_open\fR
\fBshm_unlink\fR        \fBshmat\fR
\fBshmctl\fR    \fBshmdt\fR
\fBshmget\fR    \fBshmids\fR
```

```
\fBsig2str\fR   \fBsigaction\fR
\fBsigaddset\fR \fBsigaltstack\fR
\fBsigdelset\fR \fBsigemptyset\fR
\fBsigfillset\fR        \fBsigfpe\fR
\fBsighold\fR   \fBsigignore\fR
\fBsiginterrupt\fR      \fBsigismember\fR
\fBsiglongjmp\fR        \fBsignal\fR
\fBsigpause\fR  \fBsigpending\fR
\fBsigprocmask\fR       \fBsigqueue\fR
\fBsigrelse\fR
\fBsigsend\fR   \fBsigsendset\fR
\fBsigset\fR    \fBsigsetjmp\fR
\fBsigstack\fR  \fBsigsuspend\fR
\fBsigtimedwait\fR      \fBsigwait\fR
\fBsigwaitinfo\fR       \fBsingle_to_decimal\fR
\fBsleep\fR     \fBsmt_pause\fR
\fBsnprintf\fR
\fBsprintf\fR   \fBsrand\fR
\fBsrand48\fR   \fBsrandom\fR
\fBsscanf\fR    \fBssignal\fR
\fBstack_getbounds\fR   \fBstack_inbounds\fR
\fBstack_setbounds\fR   \fBstack_violation\fR
\fBstat\fR      \fBstatfs\fR
\fBstatvfs\fR   \fBstime\fR
\fBstr2sig\fR
\fBstrcasecmp\fR        \fBstrcasecmp_l\fR
\fBstrcat\fR    \fBstrchr\fR
\fBstrcmp\fR    \fBstrcoll\fR
\fBstrcpy\fR    \fBstrcspn\fR
\fBstrdup\fR    \fBstrerror\fR
\fBstrerror_l\fR        \fBstrerror_r\fR
\fBstrfmon\fR   \fBstrfmon_l\fR
\fBstrftime\fR  \fBstrftime_l\fR
\fBstring_to_decimal\fR
\fBstrlcat\fR   \fBstrlcpy\fR
\fBstrlen\fR    \fBstrncasecmp\fR
\fBstrncasecmp_l\fR     \fBstrncat\fR
\fBstrncmp\fR   \fBstrncpy\fR
\fBstrnstr\fR   \fBstrpbrk\fR
\fBstrptime\fR  \fBstrptime_l\fR
\fBstrrchr\fR   \fBstrsep\fR
\fBstrsignal\fR \fBstrspn\fR
\fBstrstr\fR    \fBstrtod\fR
\fBstrtof\fR    \fBstrtoimax\fR
\fBstrtok\fR    \fBstrtok_r\fR
\fBstrtol\fR    \fBstrtold\fR
\fBstrtoll\fR   \fBstrtoul\fR
\fBstrtoull\fR  \fBstrtoumax\fR
\fBstrtows\fR   \fBstrxfrm\fR
\fBswab\fR      \fBswapcontext\fR
\fBswapctl\fR   \fBswprintf\fR
\fBswscanf\fR   \fBsymlink\fR
\fBsymlinkat\fR
\fBsync\fR      \fBsync_instruction_memory\fR
\fBsysconf\fR   \fBsysfs\fR
\fBsysinfo\fR   \fBsyslog\fR
\fBsystem\fR    \fBtcdrain\fR
\fBtcflow\fR    \fBtcflush\fR
\fBtcgetattr\fR \fBtcgetpgrp\fR
\fBtcgetsid\fR  \fBtcsendbreak\fR
\fBtcsetattr\fR \fBtcsetpgrp\fR
\fBtdelete\fR   \fBtell\fR
\fBtelldir\fR   \fBtempnam\fR
\fBtextdomain\fR        \fBtfind\fR
\fBthr_continue\fR      \fBthr_create\fR
\fBthr_exit\fR  \fBthr_getconcurrency\fR
```



```
\fBthr_getprio\fR       \fBthr_getspecific\fR
\fBthr_join\fR  \fBthr_keycreate\fR
\fBthr_keycreate_once\fR
\fBthr_kill\fR  \fBthr_main\fR
\fBthr_min_stack\fR     \fBthr_self\fR
\fBthr_setconcurrency\fR        \fBthr_setprio\fR
\fBthr_setspecific\fR   \fBthr_sigsetmask\fR
\fBthr_stksegment\fR    \fBthr_suspend\fR
\fBthr_yield\fR \fBtime\fR
\fBtimer_create\fR      \fBtimer_delete\fR
\fBtimer_getoverrun\fR  \fBtimer_gettime\fR
\fBtimer_settime\fR
\fBtimes\fR     \fBtimezone\fR
\fBtmpfile\fR   \fBtmpnam\fR
\fBtmpnam_r\fR  \fBtoascii\fR
\fBtolower\fR   \fBtolower_l\fR
\fBtoupper\fR   \fBtoupper_l\fR
\fBtowctrans\fR \fBtowctrans_l\fR
\fBtowlower\fR  \fBtowlower_l\fR
\fBtowupper\fR  \fBtowupper_l\fR
\fBtruncate\fR
\fBtsearch\fR   \fBttyname\fR
\fBttyname_r\fR \fBttyslot\fR
\fBtwalk\fR     \fBtzname\fR
\fBtzset\fR     \fBu8_strcmp\fR
\fBu8_textprep_str\fR   \fBu8_validate\fR
\fBuadmin\fR
\fBualarm\fR    \fBuconv_u16tou32\fR
\fBuconv_u16tou8\fR     \fBuconv_u32tou16\fR
\fBuconv_u32tou8\fR     \fBuconv_u8tou16\fR
\fBuconv_u8tou32\fR     \fBucred_free\fR
\fBucred_get\fR \fBucred_getegid\fR
\fBucred_geteuid\fR     \fBucred_getgroups\fR
\fBucred_getpflags\fR   \fBucred_getpid\fR
\fBucred_getprivset\fR  \fBucred_getprojid\fR
\fBucred_getrgid\fR     \fBucred_getruid\fR
\fBucred_getsgid\fR     \fBucred_getsuid\fR
\fBucred_getzoneid\fR   \fBucred_size\fR
\fBulckpwdf\fR  \fBulimit\fR
\fBulltostr\fR  \fBumask\fR
\fBumount\fR    \fBumount2\fR
\fBuname\fR     \fBungetc\fR
\fBungetwc\fR   \fBunlink\fR
\fBunlinkat\fR  \fBunlockpt\fR
\fBunordered\fR \fBunsetenv\fR
\fBupdwtmp\fR   \fBupdwtmpx\fR
\fBuselocale\fR
\fBusleep\fR    \fBustat\fR
\fButime\fR     \fButimensat\fR
\fButimes\fR    \fButmpname\fR
\fButmpxname\fR \fBuucopy\fR
\fBvalloc\fR    \fBvasprintf\fR
\fBverr\fR      \fBverrx\fR
\fBvfork\fR     \fBvforkx\fR
\fBvfprintf\fR  \fBvfscanf\fR
\fBvfwprintf\fR \fBvfwscanf\fR
\fBvhangup\fR   \fBvlfmt\fR
\fBvpfmt\fR     \fBvprintf\fR
\fBvscanf\fR    \fBvsnprintf\fR
\fBvsprintf\fR  \fBvsscanf\fR
\fBvswprintf\fR \fBvswscanf\fR
\fBvsyslog\fR   \fBvwarn\fR
\fBvwarnx\fR    \fBvwprintf\fR
\fBvwscanf\fR   \fBwait\fR
\fBwait3\fR     \fBwait4\fR
\fBwaitid\fR    \fBwaitpid\fR
```

```
\fBwalkcontext\fR       \fBwarn\fR
\fBwarnx\fR     \fBwatoll\fR
\fBwcpcpy\fR    \fBwcpncpy\fR
\fBwcrtomb\fR   \fBwcrtomb_l\fR
\fBwcscasecmp\fR        \fBwcscasecmp_l\fR
\fBwcscat\fR
\fBwcschr\fR    \fBwcscmp\fR
\fBwcscoll\fR   \fBwcscoll_l\fR
\fBwcscpy\fR    \fBwcscspn\fR
\fBwcsdup\fR    \fBwcsftime\fR
\fBwcslen\fR    \fBwcsncat\fR
\fBwcsncasecmp\fR       \fBwcsncasecmp_l\fR
\fBwcsncmp\fR   \fBwcsncpy\fR
\fBwcsnrtombs\fR        \fBwcsnrtombs_l\fR
\fBwcspbrk\fR   \fBwcsrchr\fR
\fBwcsrtombs\fR \fBwcsrtombs_l\fR
\fBwcsspn\fR
\fBwcsstr\fR    \fBwcstod\fR
\fBwcstof\fR    \fBwcstoimax\fR
\fBwcstok\fR    \fBwcstol\fR
\fBwcstold\fR   \fBwcstoll\fR
\fBwcstombs\fR  \fBwctombs_l\fR
\fBwcstoul\fR
\fBwcstoull\fR  \fBwcstoumax\fR
\fBwcswcs\fR
\fBwcswidth\fR  \fBwcswidth_l\fR
\fBwcsxfrm\fR   \fBwcsxfrm_l\fR
\fBwctob\fR     \fBwctob_l\fR
\fBwctomb\fR    \fBwctomb_l\fR
\fBwctrans\fR   \fBwctrans_l\fR
\fBwctype\fR    \fBwctype_l\fR
\fBwcwidth\fR   \fBwcwidth_l\fR
\fBwmemchr\fR   \fBwmemcmp\fR
\fBwmemcpy\fR   \fBwmemmove\fR
\fBwmemset\fR   \fBwordexp\fR
\fBwordfree\fR  \fBwprintf\fR
\fBwracct\fR    \fBwrite\fR
\fBwritev\fR    \fBwscanf\fR
\fBwscasecmp\fR \fBwscat\fR
\fBwschr\fR     \fBwscmp\fR
\fBwscol\fR     \fBwscoll\fR
\fBwscpy\fR     \fBwscspn\fR
\fBwsdup\fR     \fBwslen\fR
\fBwsncasecmp\fR        \fBwsncat\fR
\fBwsncmp\fR    \fBwsncpy\fR
\fBwspbrk\fR    \fBwsprintf\fR
\fBwsrchr\fR    \fBwsscanf\fR
\fBwsspn\fR     \fBwstod\fR
\fBwstok\fR     \fBwstol\fR
\fBwstoll\fR    \fBwstostr\fR
\fBwsxfrm\fR    \fByield\fR
.TE

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

.sp
.TS
l l
l l .
\fB__div64\fR   \fB__mul64\fR
\fB__posix_readdir_r\fR \fB__rem64\fR
\fB__udiv64\fR  \fB__urem64\fR
\fB_bufendtab\fR        \fB_lastbuf\fR
```



```
\fB_s_fcntl\fR  \fB_sys_nsig\fR
\fB_xftw64\fR   \fBaio_cancel64\fR
\fBaio_error64\fR       \fBaio_fsync64\fR
\fBaio_read64\fR        \fBaio_return64\fR
\fBaio_suspend64\fR     \fBaio_waitn64\fR
\fBaio_write64\fR       \fBcreat64\fR
\fBfgetpos64\fR \fBfopen64\fR
\fBfreopen64\fR \fBfseeko64\fR
\fBfsetpos64\fR \fBfstat64\fR
\fBfstatvfs64\fR        \fBftello64\fR
\fBftruncate64\fR       \fBftw64\fR
\fBgetdents64\fR        \fBgetrlimit64\fR
\fBlio_listio64\fR      \fBlockf64\fR
\fBlseek64\fR   \fBlstat64\fR
\fBmkstemp64\fR \fBmmap64\fR
\fBnftw64\fR    \fBopen64\fR
\fBpread64\fR   \fBpreadv64\fR
\fBptrace\fR    \fBpwrite64\fR
\fBpwritev64\fR \fBreaddir64\fR
\fBreaddir64_r\fR       \fBs_fcntl\fR
\fBs_ioctl\fR   \fBselect_large_fdset\fR
\fBsetrlimit64\fR       \fBstat64\fR
\fBstatvfs64\fR \fBsys_errlist\fR
\fBsys_nerr\fR  \fBtell64\fR
\fBtmpfile64\fR \fBtruncate64\fR
.TE

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

.sp
.TS
l l
l l .
\fB\&.div\fR    \fB\&.mul\fR
\fB\&.rem\fR    \fB\&.stret1\fR
\fB\&.stret2\fR \fB\&.stret4\fR
\fB\&.stret8\fR \fB\&.udiv\fR
\fB\&.umul\fR   \fB\&.urem\fR
\fB_Q_add\fR    \fB_Q_cmp\fR
\fB_Q_cmpe\fR   \fB_Q_div\fR
\fB_Q_dtoq\fR   \fB_Q_feq\fR
\fB_Q_fge\fR    \fB_Q_fgt\fR
\fB_Q_fle\fR    \fB_Q_flt\fR
\fB_Q_fne\fR    \fB_Q_itoq\fR
\fB_Q_lltoq\fR  \fB_Q_mul\fR
\fB_Q_neg\fR    \fB_Q_qtod\fR
\fB_Q_qtoi\fR   \fB_Q_qtoll\fR
\fB_Q_qtos\fR   \fB_Q_qtou\fR
\fB_Q_qtoull\fR \fB_Q_sqrt\fR
\fB_Q_stoq\fR   \fB_Q_sub\fR
\fB_Q_ulltoq\fR \fB_Q_utoq\fR
\fB__dtoll\fR   \fB__dtou\fR
\fB__dtoull\fR  \fB__ftoll\fR
\fB__ftou\fR    \fB__ftoull\fR
\fB__umul64\fR
.TE

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

```
 917 .sp
 918 .TS
 919 l l
 920 l l .
 921 \fB__fpstart\fR \fB_fp_hw\fR
 922 \fB_fpstart\fR  \fB_fxstat\fR
 923 \fB_lxstat\fR   \fB_nuname\fR
 924 \fB_thr_errno_addr\fR   \fB_xmknod\fR
 925 \fB_xstat\fR    \fBnuname\fR
 926 .TE

 928 .sp
 929 .LP
 930 The following interfaces are unique to the 64-bit SPARC version of this
 931 library:
 932 .sp

 934 .sp
 935 .TS
 936 l l
 937 l l .
 938 \fB_Qp_add\fR   \fB_Qp_cmp\fR
 939 \fB_Qp_cmpe\fR  \fB_Qp_div\fR
 940 \fB_Qp_dtoq\fR  \fB_Qp_feq\fR
 941 \fB_Qp_fge\fR   \fB_Qp_fgt\fR
 942 \fB_Qp_fle\fR   \fB_Qp_flt\fR
 943 \fB_Qp_fne\fR   \fB_Qp_itoq\fR
 944 \fB_Qp_mul\fR   \fB_Qp_neg\fR
 945 \fB_Qp_qtod\fR  \fB_Qp_qtoi\fR
 946 \fB_Qp_qtos\fR  \fB_Qp_qtoui\fR
 947 \fB_Qp_qtoux\fR \fB_Qp_qtox\fR
 948 \fB_Qp_sqrt\fR  \fB_Qp_stoq\fR
 949 \fB_Qp_sub\fR   \fB_Qp_uitoq\fR
 950 \fB_Qp_uxtoq\fR \fB_Qp_xtoq\fR
 951 \fB__align_cpy_1\fR     \fB__align_cpy_16\fR
 952 \fB__align_cpy_2\fR     \fB__align_cpy_4\fR
 953 \fB__align_cpy_8\fR     \fB__dtoul\fR
 954 \fB__ftoul\fR   \fB__sparc_utrap_install\fR
 955 .TE

 957 .SH FILES
 958 .ne 2
 959 .na
 960 \fB\fB/lib/libc.so.1\fR\fR
 961 .ad
 962 .RS 27n
 963 shared object
 964 .RE

 966 .sp
 967 .ne 2
 968 .na
 969 \fB\fB/lib/64/libc.so.1\fR\fR
 970 .ad
 971 .RS 27n
 972 64-bit shared object
 973 .RE

 975 .sp
 976 .ne 2
 977 .na
 978 \fB\fB/lib/c_synonyms.so.1\fR\fR
 979 .ad
 980 .RS 27n
 981 A compatibility library to provide access to obsolete \fBlibc\fR synonym
 982 symbols
```



```
 983 .RE

 985 .sp
 986 .ne 2
 987 .na
 988 \fB\fB/lib/64/c_synonyms.so.1\fR\fR
 989 .ad
 990 .RS 27n
 991 A 64-bit compatibility library to provide access to obsolete \fBlibc\fR synonym
 992 symbols
 993 .RE

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

 999 .sp
1000 .TS
1001 box;
1002 c | c
1003 l | l .
1004 ATTRIBUTE TYPE  ATTRIBUTE VALUE
1005 MT-Level        Safe
1006 .TE

1008 .SH SEE ALSO
1012 .LP
1009 \fBpvs\fR(1), \fBIntro\fR(2), \fBIntro\fR(3), \fBattributes\fR(5),
1010 \fBlf64\fR(5), \fBstandards\fR(5)
1011 .SH NOTES
1016 .LP
1012 The synonyms compatibility library, \fBc_synonyms.so.1\fR, provides a mechanism
1013 to support old applications and libraries that were mistakenly built using
1014 now-obsolete synonym symbols from \fBlibc\fR.
1015 .LP
1016 Before the advent of direct binding (-B direct) \fBlibc\fR provided many
1017 functions with two names.  For example, \fBgetpwent()\fR and \fB_getpwent()\fR.
1018 These two names referred to exactly the same function in \fBlibc\fR. The
1019 leading-underscore symbol was intended to be used by system libraries in order
1020 to avoid conflicting with an application that might define its own version of
1021 \fBgetpwent()\fR with completely different semantics. Standard-conforming
1022 applications may not define and use function names with leading underscores.
1023 .LP
1024 illumos system libraries are now built with direct binding. This means that a
1025 system library that calls \fBgetpwent()\fR will bind directly to the instance
1026 of \fBgetpwent()\fR in \fBlibc\fR, even if the application to which it is
1027 linked defines a different \fBgetpwent()\fR for its own use. The application
1028 binds to its instance of \fBgetpwent()\fR and there is no resulting conflict.
1029 The direct binding mechanism is equally available to libraries not delivered
1030 with illumos.
1031 .LP
1032 As a result of this evolution, most of the leading-underscore synonym symbols
1033 have been removed from \fBlibc\fR.  This means that applications that call
1034 these now-obsolete function names will cease to work.  They will typically draw
1035 the error:
1036 .sp
1037 .in +2
1038 .nf
1039 $ ./application
1040 ld.so.1: fatal: relocation error: symbol _getpwent:
1041 referenced symbol not found
1042 Killed
1043 .fi
1044 .in -2
1045 .sp
```

```
.sp
.LP
All of the old leading-underscore symbols have been copied to the synonyms
compatibility library. This library simply redirects the calls to the
non-underscore instances of the corresponding functions in \fBlibc\fR. Use it
as a pre-loaded object:
.sp
.in +2
.nf
$ LD_PRELOAD=c_synonyms.so.1 ./application
.fi
.in -2
.sp

.LP
The synonyms compatibility library is intended neither to enable the generation
of applications that call the obsolete leading-underscore synonym functions,
nor to endorse this particular programming practice.
```

```
**********************************************************
    9993 Mon Aug 26 06:56:04 2019
new/usr/src/man/man3nsl/netdir.3nsl
11622 clean up rarer mandoc lint warnings
**********************************************************
'\" te
.\" Copyright 2015 Nexenta Systems, Inc.  All rights reserved.
.\"  Copyright 1989 AT&T
.\" 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 NETDIR 3NSL "Feb 14, 2015"
.SH NAME
netdir, netdir_getbyname, netdir_getbyaddr, netdir_free, netdir_options,
taddr2uaddr, uaddr2taddr, netdir_perror, netdir_sperror, netdir_mergeaddr \-
generic transport name-to-address translation
.SH SYNOPSIS
.LP
.nf
\fBcc\fR [ \fIflag\fR... ] \fIfile\fR... \fB-lnsl\fR [ \fIlibrary\fR...]
#include <netdir.h>

\fBint\fR \fBnetdir_getbyname\fR(\fBstruct netconfig  *\fR\fIconfig\fR,
     \fBstruct nd_hostserv *\fR\fIservice\fR, \fBstruct nd_addrlist **\fR\fIaddr
.fi

.LP
.nf
\fBint\fR \fBnetdir_getbyaddr\fR(\fBstruct netconfig  *\fR\fIconfig\fR,
     \fBstruct nd_hostservlist **\fR\fIservice\fR, \fBstruct netbuf  *\fR\fIneta
.fi

.LP
.nf
\fBvoid\fR \fBnetdir_free\fR(\fBvoid *\fR\fIptr\fR, \fBint\fR \fIstruct_type\fR)
.fi

.LP
.nf
\fBint\fR \fBnetdir_options\fR(\fBstruct netconfig  *\fR\fIconfig\fR, \fBint\fR
     \fBchar *\fR\fIpointer_to_args\fR);
.fi

.LP
.nf
\fBchar *\fR\fBtaddr2uaddr\fR(\fBstruct netconfig *\fR\fIconfig\fR, \fBstruct ne
.fi

.LP
.nf
\fBstruct netbuf *\fR\fBuaddr2taddr\fR(\fBstruct netconfig *\fR\fIconfig\fR, \fB
.fi

.LP
.nf
\fBvoid\fR \fBnetdir_perror\fR(\fBchar *\fR\fIs\fR);
.fi

.LP
.nf
\fBchar *\fR\fBnetdir_sperror\fR(\fBvoid\fR);
.fi

.SH DESCRIPTION
.LP
```



```
The \fBnetdir\fR functions provide a generic interface for name-to-address
mapping that will work with all transport protocols. This interface provides a
generic way for programs to convert transport specific addresses into common
structures and back again. The \fBnetconfig\fR structure, described on the
\fBnetconfig\fR(4) manual page, identifies the transport.
.sp
.LP
The \fBnetdir_getbyname()\fR function maps the machine name and service name in
the \fBnd_hostserv\fR structure to a collection of addresses of the type
understood by the transport identified in the \fBnetconfig\fR structure. This
function returns all addresses that are valid for that transport in the
\fBnd_addrlist\fR structure. The \fBnd_hostserv\fR structure contains the
following members:
.sp
.in +2
.nf
char *h_host;      /* host name */
char *h_serv;      /* service name */
.fi
.in -2

.sp
.LP
The \fBnd_addrlist\fR structure contains the following members:
.sp
.in +2
.nf
int  n_cnt;       /* number of addresses */
struct netbuf *n_addrs;
.fi
.in -2

.sp
.LP
The \fBnetdir_getbyname()\fR function accepts some special-case host names. The
host names are defined in <\fBnetdir.h\fR>. The currently defined host names
are:
.sp
.ne 2
.na
\fB\fBHOST_SELF\fR\fR
.ad
.RS 21n
Represents the address to which local programs will bind their endpoints.
\fBHOST_SELF\fR differs from the host name provided by \fBgethostname\fR(3C),
which represents the address to which \fIremote\fR programs will bind their
endpoints.
.RE

.sp
.ne 2
.na
\fB\fBHOST_ANY\fR\fR
.ad
.RS 21n
Represents any host accessible by this transport provider. \fBHOST_ANY\fR
allows applications to specify a required service without specifying a
particular host name.
.RE

.sp
.ne 2
.na
\fB\fBHOST_SELF_CONNECT\fR\fR
.ad
.RS 21n
```

```
126 Represents the host address that can be used to connect to the local host.
127 .RE

129 .sp
130 .ne 2
131 .na
132 \fB\fBHOST_BROADCAST\fR\fR
133 .ad
134 .RS 21n
135 Represents the address for all hosts accessible by this transport provider.
136 Network requests to this address are received by all machines.
137 .RE

139 .sp
140 .LP
141 All fields of the \fBnd_hostserv\fR structure must be initialized.
142 .sp
143 .LP
144 To find the address of a given host and service on all available transports,
145 call the \fBnetdir_getbyname()\fR function with each \fBstruct netconfig\fR
146 structure returned by \fBgetnetconfig\fR(3NSL).
147 .sp
148 .LP
149 The \fBnetdir_getbyaddr()\fR function maps addresses to service names. The
150 function returns \fIservice\fR, a list of host and service pairs that yield
151 these addresses. If more than one tuple of host and service name is returned,
152 the first tuple contains the preferred host and service names:
153 .sp
154 .in +2
155 .nf
156 struct nd_hostservlist {
157    int  *h_cnt;                 /* number of hostservs found */
158    struct hostserv *h_hostservs;
159 }
160 .fi
161 .in -2

163 .sp
164 .LP
165 The \fBnetdir_free()\fR structure is used to free the structures allocated by
166 the name to address translation functions. The \fIptr\fR parameter points to
167 the structure that has to be freed. The parameter \fBstruct_type\fR identifies
168 the structure:
169 .sp
170 .in +2
171 .nf
172 struct netbuf            ND_ADDR
173 struct nd_addrlist       ND_ADDRLIST
174 struct hostserv          ND_HOSTSERV
175 struct nd_hostservlist   ND_HOSTSERVLIST
176 .fi
177 .in -2

179 .sp
180 .LP
181 The \fBnetdir_options()\fR function is used to do all transport-specific setups
182 and option management. \fIfildes\fR is the associated file descriptor.
183 \fIoption\fR, \fIfildes\fR, and \fIpointer_to_args\fR are passed to the
184 \fBnetdir_options()\fR function for the transport specified in \fIconfig\fR.
185 Currently four values are defined for \fIoption\fR:
188 .br

187 .in +2
188 \fBND_SET_BROADCAST\fR
189 .in -2
190 .br
```



```
191 .in +2
192 \fBND_SET_RESERVEDPORT\fR
193 .in -2
194 .br
195 .in +2
196 \fBND_CHECK_RESERVEDPORT\fR
197 .in -2
198 .br
199 .in +2
200 \fBND_MERGEADDR\fR
201 .sp
202 .LP
203 The specific actions of each option follow.
204 .sp
205 .ne 2
206 .na
207 \fB\fBND_SET_BROADCAST\fR\fR
208 .ad
209 .RS 25n
210 Sets the transport provider up to allow broadcast if the transport supports
211 broadcast. \fIfildes\fR is a file descriptor into the transport, that is, the
212 result of a \fBt_open\fR of \fB/dev/udp\fR. \fIpointer_to_args\fR is not used.
213 If this completes, broadcast operations can be performed on file descriptor
214 \fIfildes\fR.
215 .RE

217 .sp
218 .ne 2
219 .na
220 \fB\fBND_SET_RESERVEDPORT\fR\fR
221 .ad
222 .RS 25n
223 Allows the application to bind to a reserved port if that concept exists for
224 the transport provider. \fIfildes\fR is an unbound file descriptor into the
225 transport. If \fIpointer_to_args\fR is \fINULL,\fR \fIfildes\fR is bound to a
226 reserved port. If \fIpointer_to_args\fR is a pointer to a \fBnetbuf\fR
227 structure, an attempt is made to bind to any reserved port on the specified
228 address.
229 .RE

231 .sp
232 .ne 2
233 .na
234 \fB\fBND_CHECK_RESERVEDPORT\fR\fR
235 .ad
236 .RS 25n
237 Used to verify that the address corresponds to a reserved port if that concept
238 exists for the transport provider. \fIfildes\fR is not used.
239 \fIpointer_to_args\fR is a pointer to a \fBnetbuf\fR structure that contains
240 the address. This option returns \fB0\fR only if the address specified in
241 \fIpointer_to_args\fR is reserved.
242 .RE

244 .sp
245 .ne 2
246 .na
247 \fB\fBND_MERGEADDR\fR\fR
248 .ad
249 .RS 25n
250 Used to take a ‘‘local address’’ such as a \fB0.0.0.0\fR TCP address and return
251 a ‘‘real address’’ to which client machines can connect. \fIfildes\fR is not
252 used. \fIpointer_to_args\fR is a pointer to a \fBstruct nd_mergearg\fR which
253 has the following members:
254 .sp
255 .in +2
256 .nf
```

```
\fBchar\fR \fIs_uaddr;\fR  /* server's universal address */
\fBchar\fR \fIc_uaddr;\fR  /* client's universal address */
\fBchar\fR \fIm_uaddr;\fR  /* the result */
.fi
.in -2

If \fBs_uaddr\fR is an address such as \fB0.0.0.0.1.12\fR, and the call is
successful \fBm_uaddr\fR is set to an address such as \fB192.11.109.89.1.12\fR.
For most transports, \fBm_uaddr\fR is identical to \fBs_uaddr\fR.
.RE
.sp
.LP
If a transport provider does not support an option, \fBnetdir_options\fR
returns \fB-1\fR and the error message can be printed through
\fBnetdir_perror()\fR or \fBnetdir_sperror()\fR.

.in -2
.sp
.LP
The \fBtaddr2uaddr()\fR and \fBuaddr2taddr()\fR functions support translation
between universal addresses and \fBTLI\fR type \fBnetbufs\fR. The
\fBtaddr2uaddr()\fR function takes a \fBstruct\fR \fBnetbuf\fR data structure
and returns a pointer to a string that contains the universal address. It
returns \fINULL\fR if the conversion is not possible. This is not a fatal
condition as some transports do not support a universal address form.
.sp
.LP
The \fBfree()\fR function is used to free the universal address returned by the
\fBtaddr2uaddr()\fR function.
.sp
.LP
The \fBuaddr2taddr()\fR function is the reverse of the \fBtaddr2uaddr()\fR
function. It returns the \fBstruct netbuf\fR data structure for the given
universal address and \fINULL\fR on failure.

.sp
.LP
The \fBnetdir_perror()\fR function prints an error message in standard output
that states the cause of a name-to-address mapping failure. The error message
is preceded by the string given as an argument.
.sp
.LP
The \fBnetdir_sperror()\fR function returns a string with an error message that
states the cause of a name-to-address mapping failure.

.SH RETURN VALUES
.LP
The \fBnetdir_sperror()\fR function returns a pointer to a buffer which
contains the error message string. The buffer is overwritten on each call. In
multithreaded applications, this buffer is implemented as thread-specific
data.
.sp
.LP
The \fBnetdir_getbyname()\fR, \fBnetdir_getbyaddr()\fR, and \fBnetdir_options()\
functions return \fB0\fR on success and a non-zero value on failure.
.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
```



```
_
MT-Level        Safe
.TE

.SH SEE ALSO
.LP
\fBgethostname\fR(3C), \fBgetnetconfig\fR(3NSL), \fBgetnetpath\fR(3NSL),
\fBnetconfig\fR(4), \fBattributes\fR(5)
```

```
**********************************************************
   23415 Mon Aug 26 06:56:04 2019
new/usr/src/man/man4/ipf.4
11622 clean up rarer mandoc lint warnings
**********************************************************
'\" te
.\" To view license terms, attribution, and copyright for IP Filter, the
.\" default path is /usr/lib/ipf/IPFILTER.LICENCE. If the illumos operating
.\" default path is /usr/lib/ipf/IPFILTER.LICENCE. If the Illumos operating
.\" environment has been installed anywhere other than the default, modify the
.\" given path to access the file at the installed location.
.\" Portions Copyright (c) 2015, Joyent, Inc.
.TH IPF 4 "Mar 18, 2015"
.SH NAME
ipf, ipf.conf, ipf6.conf \- IP packet filter rule syntax
.SH DESCRIPTION
.PP
A rule file for \fBipf\fP may have any name or even be stdin.  As
\fBipfstat\fP produces parsable rules as output when displaying the internal
kernel filter lists, it is quite plausible to use its output to feed back
into \fBipf\fP.  Thus, to remove all filters on input packets, the following
could be done:
.nf

# ipfstat \-i | ipf \-rf \-\fP
\fC# ipfstat \-i | ipf \-rf \-\fP
.fi
.SH GRAMMAR
.PP
The format used by \fBipf\fP for construction of filtering rules can be
described using the following grammar in BNF:
\fC
.nf
filter-rule = [ insert ] action in-out [ options ] [ tos ] [ ttl ]
              [ proto ] ip [ group ].

insert  = "@" decnumber .
action  = block | "pass" | log | "count" | skip | auth | call .
in-out  = "in" | "out" .
options = [ log ] [ tag ] [ "quick" ] [ "on" interface-name [ dup ]
           [ froute ] [ replyto ] ] .
tos     = "tos" decnumber | "tos" hexnumber .
ttl     = "ttl" decnumber .
proto   = "proto" protocol .
ip      = srcdst [ flags ] [ with withopt ] [ icmp ] [ keep ] .
group   = [ "head" decnumber ] [ "group" decnumber ] .

block   = "block" [ return-icmp[return-code] | "return-rst" ] .
log     = "log" [ "body" ] [ "first" ] [ "or-block" ] [ "level" loglevel ] .
tag     = "tag" tagid .
skip    = "skip" decnumber .
auth    = "auth" | "preauth" .
call    = "call" [ "now" ] function-name .
dup     = "dup-to" interface-name [ ":" ipaddr ] .
froute  = "fastroute" | "to" interface-name [ ":" ipaddr ] .
replyto = "reply-to" interface-name [ ":" ipaddr ] .
protocol = "tcp/udp" | "udp" | "tcp" | "icmp" | decnumber .
srcdst  = "all" | fromto .
fromto  = "from" [ "!" ] object "to" [ "!" ] object .

return-icmp = "return-icmp" | "return-icmp-as-dest" .
return-code = "(" icmp-code ")" .
object  = addr [ port-comp | port-range ] .
addr    = "any" | nummask | host-name [ "mask" ipaddr | "mask" hexnumber ] .
addr    = "any" | "<thishost>" | nummask |
          host-name [ "mask" ipaddr | "mask" hexnumber ] .
```



```
port-comp = "port" compare port-num .
port-range = "port" port-num range port-num .
flags   = "flags" flag { flag } [ "/" flag { flag } ] .
with    = "with" | "and" .
icmp    = "icmp-type" icmp-type [ "code" decnumber ] .
return-code = "(" icmp-code ")" .
keep    = "keep" "state" [ "(" state-options ")" ] | "keep" "frags" .
loglevel = facility"."priority | priority .

nummask = host-name [ "/" decnumber ] .
host-name = ipaddr | hostname | "any" .
ipaddr  = host-num "." host-num "." host-num "." host-num .
host-num = digit [ digit [ digit ] ] .
port-num = service-name | decnumber .
state-options = state-opts [ "," state-options ] .

state-opts = "age" decnumber [ "/" decnumber ] | "strict" |
             "no-icmp-err" | "limit" decnumber | "newisn" | "sync" .
withopt = [ "not" | "no" ] opttype [ withopt ] .
opttype = "ipopts" | "short" | "frag" | "opt" optname .
optname = ipopts [ "," optname ] .
ipopts  = optlist | "sec-class" [ secname ] .
secname = seclvl [ "," secname ] .
seclvl  = "unclass" | "confid" | "reserv-1" | "reserv-2" | "reserv-3" |
          "reserv-4" | "secret" | "topsecret" .
icmp-type = "unreach" | "echo" | "echorep" | "squench" | "redir" |
            "timex" | "paramprob" | "timest" | "timestrep" | "inforeq" |
            "inforep" | "maskreq" | "maskrep"  | decnumber .
icmp-code = decumber | "net-unr" | "host-unr" | "proto-unr" | "port-unr" |
            "needfrag" | "srcfail" | "net-unk" | "host-unk" | "isolate" |
            "net-prohib" | "host-prohib" | "net-tos" | "host-tos" |
            "filter-prohib" | "host-preced" | "cutoff-preced" .
optlist = "nop" | "rr" | "zsu" | "mtup" | "mtur" | "encode" | "ts" |
          "tr" | "sec" | "lsrr" | "e-sec" | "cipso" | "satid" | "ssrr" |
          "addext" | "visa" | "imitd" | "eip" | "finn" .
facility = "kern" | "user" | "mail" | "daemon" | "auth" | "syslog" |
           "lpr" | "news" | "uucp" | "cron" | "ftp" | "authpriv" |
           "audit" | "logalert" | "local0" | "local1" | "local2" |
           "local3" | "local4" | "local5" | "local6" | "local7" .
priority = "emerg" | "alert" | "crit" | "err" | "warn" | "notice" |
           "info" | "debug" .

hexnumber = "0" "x" hexstring .
hexstring = hexdigit [ hexstring ] .
decnumber = digit [ decnumber ] .

compare = "=" | "!=" | "<" | ">" | "<=" | ">=" | "eq" | "ne" | "lt" |
          "gt" | "le" | "ge" .
range   = "<>" | "><" .
hexdigit = digit | "a" | "b" | "c" | "d" | "e" | "f" .
digit   = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9" .
flag    = "F" | "S" | "R" | "P" | "A" | "U" .
.fi
.PP
This syntax is somewhat simplified for readability, some combinations
that match this grammar are disallowed by the software because they do
not make sense (such as tcp \fBflags\fP for non-TCP packets).
.SH FILTER RULES
.PP
The "briefest" valid rules are (currently) no-ops and are of the form:
.nf
      block in all
      pass in all
      log out all
      count in all
.fi
```

```
122 .PP
123 Filter rules are checked in order, with the last matching rule
124 determining the fate of the packet (but see the \fBquick\fP option,
125 below).
126 .PP
127 Filters are installed by default at the end of the kernel's filter
128 lists, prepending the rule with \fB@n\fP will cause it to be inserted
129 as the n'th entry in the current list. This is especially useful when
130 modifying and testing active filter rulesets. See \fBipf\fP(1M) for more
131 information.
132 .SH ACTIONS
137 .PP
133 The action indicates what to do with the packet if it matches the rest
134 of the filter rule. Each rule MUST have an action. The following
135 actions are recognised:
136 .TP
137 .B block
138 indicates that the packet should be flagged to be dropped. In response
139 to blocking a packet, the filter may be instructed to send a reply
140 packet, either an ICMP packet (\fBreturn-icmp\fP), an ICMP packet
141 masquerading as being from the original packet's destination
142 (\fBreturn-icmp-as-dest\fP), or a TCP "reset" (\fBreturn-rst\fP).  An
143 ICMP packet may be generated in response to any IP packet, and its
144 type may optionally be specified, but a TCP reset may only be used
145 with a rule which is being applied to TCP packets.  When using
146 \fBreturn-icmp\fP or \fBreturn-icmp-as-dest\fP, it is possible to specify
147 the actual unreachable 'type'.  That is, whether it is a network
148 unreachable, port unreachable or even administratively
149 prohibited. This is done by enclosing the ICMP code associated with
150 it in parenthesis directly following \fBreturn-icmp\fP or
151 \fBreturn-icmp-as-dest\fP as follows:
152 .nf
153         block return-icmp(11) ...
154 .fi
155 .PP
156 Would return a Type-Of-Service (TOS) ICMP unreachable error.
157 .TP
158 .B pass
159 will flag the packet to be let through the filter.
160 .TP
161 .B log
162 causes the packet to be logged (as described in the LOGGING section
163 below) and has no effect on whether the packet will be allowed through
164 the filter.
165 .TP
166 .B count
167 causes the packet to be included in the accounting statistics kept by
168 the filter, and has no effect on whether the packet will be allowed through
169 the filter. These statistics are viewable with ipfstat(1M).
170 .TP
171 .B call
172 this action is used to invoke the named function in the kernel, which
173 must conform to a specific calling interface. Customised actions and
174 semantics can thus be implemented to supplement those available. This
175 feature is for use by knowledgeable hackers, and is not currently
176 documented.
177 .TP
178 .B "skip <n>"
179 causes the filter to skip over the next \fIn\fP filter rules.  If a rule is
180 inserted or deleted inside the region being skipped over, then the value of
181 \fIn\fP is adjusted appropriately.
182 .TP
183 .B auth
184 this allows authentication to be performed by a user-space program running
185 and waiting for packet information to validate.  The packet is held for a
186 period of time in an internal buffer whilst it waits for the program to return
```



```
187 to the kernel the \fIreal\fP flags for whether it should be allowed through
188 or not.  Such a program might look at the source address and request some sort
189 of authentication from the user (such as a password) before allowing the
190 packet through or telling the kernel to drop it if from an unrecognised source.
191 .TP
192 .B preauth
193 tells the filter that for packets of this class, it should look in the
194 pre-authenticated list for further clarification.  If no further matching
195 rule is found, the packet will be dropped (the FR_PREAUTH is not the same
196 as FR_PASS).  If a further matching rule is found, the result from that is
197 used in its instead.  This might be used in a situation where a person
198 \fIlogs in\fP to the firewall and it sets up some temporary rules defining
199 the access for that person.
200 .PP
201 The next word must be either \fBin\fP or \fBout\fP.  Each packet
202 moving through the kernel is either inbound (just been received on an
203 interface, and moving towards the kernel's protocol processing) or
204 outbound (transmitted or forwarded by the stack, and on its way to an
205 interface). There is a requirement that each filter rule explicitly
206 state which side of the I/O it is to be used on.
207 .SH OPTIONS
213 .PP
208 The list of options is brief, and all are indeed optional. Where
209 options are used, they must be present in the order shown here. These
210 are the currently supported options:
211 .TP
212 .B log
213 indicates that, should this be the last matching rule, the packet
214 header will be written to the \fBipl\fP log (as described in the
215 LOGGING section below).
216 .TP
217 .B tag tagid
218 indicates that, if this rule causes the packet to be logged or entered
219 in the state table, the tagid will be logged as part of the log entry.
220 This can be used to quickly match "similar" rules in scripts that post
221 process the log files for e.g. generation of security reports or accounting
222 purposes. The tagid is a 32 bit unsigned integer.
223 .TP
224 .B quick
225 allows "short-cut" rules in order to speed up the filter or override
226 later rules.  If a packet matches a filter rule which is marked as
227 \fBquick\fP, this rule will be the last rule checked, allowing a
228 "short-circuit" path to avoid processing later rules for this
229 packet. The current status of the packet (after any effects of the
230 current rule) will determine whether it is passed or blocked.
231 .IP
232 If this option is missing, the rule is taken to be a "fall-through"
233 rule, meaning that the result of the match (block/pass) is saved and
234 that processing will continue to see if there are any more matches.
235 .TP
236 .B on
237 allows an interface name to be incorporated into the matching
238 procedure. Interface names are as printed by "netstat \-i". If this
239 option is used, the rule will only match if the packet is going
240 through that interface in the specified direction (in/out). If this
241 option is absent, the rule is taken to be applied to a packet
242 regardless of the interface it is present on (i.e. on all interfaces).
243 Filter rulesets are common to all interfaces, rather than having a
244 filter list for each interface.
245 .IP
246 This option is especially useful for simple IP-spoofing protection:
247 packets should only be allowed to pass inbound on the interface from
248 which the specified source address would be expected, others may be
249 logged and/or dropped.
250 .TP
251 .B dup-to
```

causes the packet to be copied, and the duplicate packet to be sent
outbound on the specified interface, optionally with the destination
IP address changed to that specified. This is useful for off-host
logging, using a network sniffer.
.TP
.B to
causes the packet to be moved to the outbound queue on the
specified interface. This can be used to circumvent kernel routing
decisions, and even to bypass the rest of the kernel processing of the
packet (if applied to an inbound rule). It is thus possible to
construct a firewall that behaves transparently, like a filtering hub
or switch, rather than a router. The \fBfastroute\fP keyword is a
synonym for this option.
.SH MATCHING PARAMETERS
.PP
The keywords described in this section are used to describe attributes
of the packet to be used when determining whether rules match or don't
match. The following general-purpose attributes are provided for
matching, and must be used in this order:
.TP
.B tos
packets with different Type-Of-Service values can be filtered.
Individual service levels or combinations can be filtered upon. The
value for the TOS mask can either be represented as a hex number or a
decimal integer value.
.TP
.B ttl
packets may also be selected by their Time-To-Live value. The value given in
the filter rule must exactly match that in the packet for a match to occur.
This value can only be given as a decimal integer value.
.TP
.B proto
allows a specific protocol to be matched against. All protocol names
found in \fB/etc/protocols\fP are recognised and may be used.
However, the protocol may also be given as a DECIMAL number, allowing
for rules to match your own protocols, or new ones which would
out-date any attempted listing.
.IP
The special protocol keyword \fBtcp/udp\fP may be used to match either
a TCP or a UDP packet, and has been added as a convenience to save
duplication of otherwise-identical rules.
.\" XXX grammar should reflect this (/etc/protocols)
.PP
The \fBfrom\fP and \fBto\fP keywords are used to match against IP
addresses (and optionally port numbers). Rules must specify BOTH
source and destination parameters.
.PP
IP addresses may be specified in one of two ways: as a numerical
address\fB/\fPmask, or as a hostname \fBmask\fP netmask. The hostname
may either be a valid hostname, from either the hosts file or DNS
(depending on your configuration and library) or of the dotted numeric
form. There is no special designation for networks but network names
are recognised. Note that having your filter rules depend on DNS
results can introduce an avenue of attack, and is discouraged.
.PP
There is a special case for the hostname \fBany\fP which is taken to
be 0.0.0.0/0 (see below for mask syntax) and matches all IP addresses.
Only the presence of "any" has an implied mask, in all other
situations, a hostname MUST be accompanied by a mask. It is possible
to give "any" a hostmask, but in the context of this language, it is
non-sensical.
.PP
The numerical format "x\fB/\fPy" indicates that a mask of y
consecutive 1 bits set is generated, starting with the MSB, so a y value
of 16 would give 0xffff0000. The symbolic "x \fBmask\fP y" indicates
that the mask y is in dotted IP notation or a hexadecimal number of

the form 0x12345678. Note that all the bits of the IP address
indicated by the bitmask must match the address on the packet exactly;
there isn't currently a way to invert the sense of the match, or to
match ranges of IP addresses which do not express themselves easily as
bitmasks (anthropomorphization; it's not just for breakfast anymore).
.PP
If a \fBport\fP match is included, for either or both of source and
destination, then it is only applied to
.\" XXX - "may only be" ? how does this apply to other protocols? will it not ma
TCP and UDP packets. If there is no \fBproto\fP match parameter,
packets from both protocols are compared. This is equivalent to "proto
tcp/udp". When composing \fBport\fP comparisons, either the service
name or an integer port number may be used. Port comparisons may be
done in a number of forms, with a number of comparison operators, or
port ranges may be specified. When the port appears as part of the
\fBfrom\fP object, it matches the source port number, when it appears
as part of the \fBto\fP object, it matches the destination port number.
See the examples for more information.
.PP
The \fBall\fP keyword is essentially a synonym for "from any to any"
with no other match parameters.
.PP
Following the source and destination matching parameters, the
following additional parameters may be used:
.TP
.B with
is used to match irregular attributes that some packets may have
associated with them. To match the presence of IP options in general,
use \fBwith ipopts\fP. To match packets that are too short to contain
a complete header, use \fBwith short\fP. To match fragmented packets,
use \fBwith frag\fP. For more specific filtering on IP options,
individual options can be listed.
.IP
Before any parameter used after the \fBwith\fP keyword, the word
\fBnot\fP or \fBno\fP may be inserted to cause the filter rule to only
match if the option(s) is not present.
.IP
Multiple consecutive \fBwith\fP clauses are allowed. Alternatively,
the keyword \fBand\fP may be used in place of \fBwith\fP, this is
provided purely to make the rules more readable ("with ... and ...").
When multiple clauses are listed, all those must match to cause a
match of the rule.
.\" XXX describe the options more specifically in a separate section
.TP
.B flags
is only effective for TCP filtering. Each of the letters possible
represents one of the possible flags that can be set in the TCP
header. The association is as follows:
.LP
.nf
        F - FIN
        S - SYN
        R - RST
        P - PUSH
        A - ACK
        U - URG
.fi
.IP
The various flag symbols may be used in combination, so that "SA"
would represent a SYN-ACK combination present in a packet. There is
nothing preventing the specification of combinations, such as "SFR",
that would not normally be generated by law-abiding TCP
implementations. However, to guard against weird aberrations, it is
necessary to state which flags you are filtering against. To allow
this, it is possible to set a mask indicating which TCP flags you wish
to compare (i.e., those you deem significant). This is done by

```
383 appending "/<flags>" to the set of TCP flags you wish to match
384 against, e.g.:
385 .LP
386 .nf
387         ... flags S
388                         # becomes "flags S/AUPRFS" and will match
389                         # packets with ONLY the SYN flag set.

391         ... flags SA
392                         # becomes "flags SA/AUPRFS" and will match any
393                         # packet with only the SYN and ACK flags set.

395         ... flags S/SA
396                         # will match any packet with just the SYN flag set
397                         # out of the SYN-ACK pair; the common "establish"
398                         # keyword action.  "S/SA" will NOT match a packet
399                         # with BOTH SYN and ACK set, but WILL match "SFP".
400 .fi
401 .TP
402 .B icmp-type
403 is only effective when used with \fBproto icmp\fP and must NOT be used
404 in conjunction with \fBflags\fP.  There are a number of types, which can be
405 referred to by an abbreviation recognised by this language, or the numbers
406 with which they are associated can be used.  The most important from
407 a security point of view is the ICMP redirect.
408 .SH KEEP HISTORY
416 .PP
409 The second last parameter which can be set for a filter rule is whether or not
410 to record historical information for that packet, and what sort to keep. The
411 following information can be kept:
412 .TP
413 .B state
414 keeps information about the flow of a communication session. State can
415 be kept for TCP, UDP, and ICMP packets.
416 .TP
417 .B frags
418 keeps information on fragmented packets, to be applied to later
419 fragments.
420 .PP
421 allowing packets which match these to flow straight through, rather
422 than going through the access control list.
423 .SH GROUPS
424 The last pair of parameters control filter rule "grouping".  By default, all
425 filter rules are placed in group 0 if no other group is specified.  To add a
426 rule to a non-default group, the group must first be started by creating a
427 group \fIhead\fP.  If a packet matches a rule which is the \fIhead\fP of a
428 group, the filter processing then switches to the group, using that rule as
429 the default for the group.  If \fBquick\fP is used with a \fBhead\fP rule, rule
430 processing isn't stopped until it has returned from processing the group.
431 .PP
432 A rule may be both the head for a new group and a member of a non-default
433 group (\fBhead\fP and \fBgroup\fP may be used together in a rule).
434 .TP
435 .B "head <n>"
436 indicates that a new group (number n) should be created.
437 .TP
438 .B "group <n>"
439 indicates that the rule should be put in group (number n) rather than group 0.
440 .SH LOGGING
449 .PP
441 When a packet is logged, with either the \fBlog\fP action or option,
442 the headers of the packet are written to the \fBipl\fP packet logging
443 pseudo-device. Immediately following the \fBlog\fP keyword, the
444 following qualifiers may be used (in order):
445 .TP
446 .B body
```



```
447 indicates that the first 128 bytes of the packet contents will be
448 logged after the headers.
449 .TP
450 .B first
451 If log is being used in conjunction with a "keep" option, it is recommended
452 that this option is also applied so that only the triggering packet is logged
453 and not every packet which thereafter matches state information.
454 .TP
455 .B or-block
456 indicates that, if for some reason the filter is unable to log the
457 packet (such as the log reader being too slow) then the rule should be
458 interpreted as if the action was \fBblock\fP for this packet.
459 .TP
460 .B "level <loglevel>"
461 indicates what logging facility and priority, or just priority with
462 the default facility being used, will be used to log information about
463 this packet using ipmon's -s option.
464 .PP
465 See ipl(4) for the format of records written
466 to this device. The ipmon(1M) program can be used to read and format
467 this log.
468 .SH EXAMPLES
478 .PP
469 The \fBquick\fP option is good for rules such as:
480 \fC
470 .nf
471 block in quick from any to any with ipopts
472 .fi
473 .PP
474 which will match any packet with a non-standard header length (IP
475 options present) and abort further processing of later rules,
476 recording a match and also that the packet should be blocked.
477 .PP
478 The "fall-through" rule parsing allows for effects such as this:
479 .LP
480 .nf
481         block in from any to any port < 6000
482         pass in from any to any port >= 6000
483         block in from any to any port > 6003
484 .fi
485 .PP
486 which sets up the range 6000-6003 as being permitted and all others being
487 denied.  Note that the effect of the first rule is overridden by subsequent
488 rules.  Another (easier) way to do the same is:
489 .LP
490 .nf
491         block in from any to any port 6000 <> 6003
492         pass in from any to any port 5999 >< 6004
493 .fi
494 .PP
495 Note that both the "block" and "pass" are needed here to effect a
496 result as a failed match on the "block" action does not imply a pass,
497 only that the rule hasn't taken effect.  To then allow ports < 1024, a
498 rule such as:
499 .LP
500 .nf
501         pass in quick from any to any port < 1024
502 .fi
503 .PP
504 would be needed before the first block.  To create a new group for
505 processing all inbound packets on le0/le1/lo0, with the default being to block
506 all inbound packets, we would do something like:
507 .LP
508 .nf
509        block in all
510        block in quick on le0 all head 100
```

```
511         block in quick on le1 all head 200
512         block in quick on lo0 all head 300
513 .fi
514 .PP

515 and to then allow ICMP packets in on le0, only, we would do:
516 .LP
517 .nf
518         pass in proto icmp all group 100
519 .fi
520 .PP
521 Note that because only inbound packets on le0 are used processed by group 100,
522 there is no need to respecify the interface name.  Likewise, we could further
523 breakup processing of TCP, etc, as follows:
524 .LP
525 .nf
526         block in proto tcp all head 110 group 100
527         pass in from any to any port = 23 group 110
528 .fi
529 .PP
530 and so on.  The last line, if written without the groups would be:
531 .LP
532 .nf
533         pass in on le0 proto tcp from any to any port = telnet
534 .fi
535 .PP
536 Note, that if we wanted to say "port = telnet", "proto tcp" would
537 need to be specified as the parser interprets each rule on its own and
538 qualifies all service/port names with the protocol specified.
539 .SH FILES
540 /dev/ipauth
541 .br
542 /dev/ipl
543 .br
544 /dev/ipstate
545 .br
546 /etc/hosts
547 .br
548 /etc/services
549 .SH SEE ALSO
550 \fBipnat\fR(4), \fBipf\fR(1M), \fBipfstat\fR(1M), \fBipfilter\fR(5)
```

```
**********************************************************
    3905 Mon Aug 26 06:56:04 2019
new/usr/src/man/man5/epoll.5
11622 clean up rarer mandoc lint warnings
**********************************************************
'\" te
.\"  Copyright (c) 2014, Joyent, Inc. All Rights Reserved.
.\"  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.
.TH EPOLL 5 "Apr 17, 2014"
.SH NAME
epoll \- Linux-compatible I/O event notification facility
.SH SYNOPSIS

.LP
.nf
#include <sys/epoll.h>
.fi

.SH DESCRIPTION
.LP

\fBepoll\fR is a facility for efficient event-oriented I/O that has a
similar model to \fBpoll\fR(2), but does not necessitate rescanning a
set of file descriptors to wait for an event.  \fBepoll\fR is of Linux
origins, and this facility is designed to be binary-compatible with
the Linux facility, including the following interfaces:

.RS +4
.TP
.ie t \(bu
.el o
\fBepoll_create\fR(3C) creates an \fBepoll\fR instance, returning a file
descriptor. It contains a size arugment which is meaningful only in as
much as it cannot be 0.
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fBepoll_create1\fR(3C) also creates an \fBepoll\fR instance, but eliminates
the meaningless size argument -- replacing it instead with a flags
argument.
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fBepoll_ctl\fR(3C) allows file descriptors to be added
(via \fBEPOLL_CTL_ADD\fR), deleted (via \fBEPOLL_CTL_DEL\fR) or
modified (via \fBEPOLL_CTL_MOD\fR) with respect to the \fBepoll\fR'd set
of file descriptors.
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fBepoll_wait\fR(3C) fetches pending events for file descriptors added
via \fBepoll_ctl\fR(3C), blocking the caller if no such events are pending.
.RE
```



```
.RS +4
.TP
.ie t \(bu
.el o
\fBepoll_pwait\fR(3C) opeates in a similar manner to \fBepoll_wait\fR(3C), but
allows the caller to specify a signal mask to be set atomically with respect
to waiting for events.
.RE

.sp
.SH NOTES
.LP

The \fBepoll\fR facility is implemented
for purposes of offering compatibility to and portability of Linux-borne
applications; native applications should continue to prefer using event ports
via the \fBport_create\fR(3C),
\fBport_associate\fR(3C) and \fBport_getn\fR(3C) interfaces.
In particular, use of \fBepoll\fR in a multithreaded environment is fraught
with peril; even when using \fBEPOLLONESHOT\fR for one-shot events,
there are race conditions with respect to \fBclose\fR(2) that are unresolvable.
(For more details, see the aborted effort in Linux to resolve this via the
proposed
\fBEPOLL_CTL_DISABLE\fR operation.)
The event port facility -- like the BSD kqueue facility that inspired it --
is designed to deal with such issues via explicit event source dissociation.

While a best effort has been made to mimic the Linux semantics, there
are some semantics that are too peculiar or ill-conceived to merit
accommodation.  In particular, the Linux \fBepoll\fR facility will -- by
design -- continue to generate events for closed file descriptors where/when
the underlying file description remains open.  For example, if one were
to \fBfork\fR(2) and subsequently close an actively \fBepoll\fR'd file
descriptor in the parent,
any events generated in the child on the implicitly duplicated file descriptor
will continue to be delivered to the parent -- despite the fact that the
parent itself no longer has any notion of the file description!
This \fBepoll\fR facility refuses to honor
these semantics; closing the \fBEPOLL_CTL_ADD\fR'd file descriptor
will always result in no further
events being generated for that event description.

.SH SEE ALSO
.LP
\fBepoll_create\fR(3C), \fBepoll_create1\fR(3C), \fBepoll_ctl\fR(3C),
\fBepoll_wait\fR(3C), \fBepoll_pwait\fR(3C),
\fBport_create\fR(3C), \fBport_associate\fR(3C), \fBport_dissociate\fR(3C),
\fBport_get\fR(3C),
\fBpselect\fR(3C)
```

```
**********************************************************
   16214 Mon Aug 26 06:56:04 2019
new/usr/src/man/man5/threads.5
11622 clean up rarer mandoc lint warnings
**********************************************************
'\" 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
```

```
252 \fBpthread_testcancel()\fR      \fB-\fR \fB-\fR
253 \fBpthread_cleanup_pop()\fR     \fB-\fR \fB-\fR
254 \fBpthread_cleanup_push()\fR    \fB-\fR \fB-\fR
255 .TE

257 .SS "Functions Related to Mutexes"

259 .TS
260 l l l
261 l l l .
262 \fBPOSIX\fR     \fBillumos\fR   \fBc11\fR
263 \fBpthread_mutex_init()\fR      \fBmutex_init()\fR      \fBmtx_init()\fR
264 \fBpthread_mutexattr_init()\fR  \fB-\fR \fB-\fR
265 \fBpthread_mutexattr_setpshared()\fR    \fB-\fR \fB-\fR
266 \fBpthread_mutexattr_getpshared()\fR    \fB-\fR \fB-\fR
267 \fBpthread_mutexattr_setprotocol()\fR   \fB-\fR \fB-\fR
268 \fBpthread_mutexattr_getprotocol()\fR   \fB-\fR \fB-\fR
269 \fBpthread_mutexattr_setprioceiling()\fR        \fB-\fR \fB-\fR
270 \fBpthread_mutexattr_getprioceiling()\fR        \fB-\fR \fB-\fR
271 \fBpthread_mutexattr_settype()\fR       \fB-\fR \fB-\fR
272 \fBpthread_mutexattr_gettype()\fR       \fB-\fR \fB-\fR
273 \fBpthread_mutexattr_setrobust()\fR     \fB-\fR \fB-\fR
274 \fBpthread_mutexattr_getrobust()\fR     \fB-\fR \fB-\fR
275 \fBpthread_mutexattr_destroy()\fR       \fB-\fR \fBmtx_destroy()\fR
276 \fBpthread_mutex_setprioceiling()\fR    \fB-\fR \fB-\fR
277 \fBpthread_mutex_getprioceiling()\fR    \fB-\fR \fB-\fR
278 \fBpthread_mutex_lock()\fR      \fBmutex_lock()\fR      \fBmtx_lock()\fR
279 \fBpthread_mutex_timedlock()\fR \fB-\fR \fBmtx_timedlock()\fR
280 \fBpthread_mutex_trylock()\fR   \fBmutex_trylock()\fR   \fBmtx_trylock()\fR
281 \fBpthread_mutex_unlock()\fR    \fBmutex_unlock()\fR    \fBmtx_unlock()\fR
282 \fBpthread_mutex_destroy()\fR   \fBmutex_destroy()\fR   \fBmtx_destroy()\fR
283 .TE

285 .SS "Functions Related to Condition Variables"

287 .TS
288 l l l
289 l l l .
290 \fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
291 \fBpthread_cond_init()\fR       \fBcond_init()\fR       \fBcnd_init()\fR
292 \fBpthread_condattr_init()\fR   \fB-\fR \fB-\fR
293 \fBpthread_condattr_setpshared()\fR     \fB-\fR \fB-\fR
294 \fBpthread_condattr_getpshared()\fR     \fB-\fR \fB-\fR
295 \fBpthread_condattr_destroy()\fR        \fB-\fR \fB-\fR
296 \fBpthread_cond_wait()\fR       \fBcond_wait()\fR       \fBcnd_wait()\fR
297 \fBpthread_cond_timedwait()\fR  \fBcond_timedwait()\fR  \fBcond_timedwait()\fR
298 \fBpthread_cond_signal()\fR     \fBcond_signal()\fR     \fBcnd_signal()\fR
299 \fBpthread_cond_broadcast()\fR  \fBcond_broadcast()\fR  \fBcnd_broadcast()\fR
300 \fBpthread_cond_destroy()\fR    \fBcond_destroy()\fR    \fBcnd_destroy()\fR
301 .TE

303 .SS "Functions Related to Reader/Writer Locking"

305 .TS
306 l l l
307 l l l .
308 \fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
309 \fBpthread_rwlock_init()\fR     \fBrwlock_init()\fR     \fB-\fR
310 \fBpthread_rwlock_rdlock()\fR   \fBrw_rdlock()\fR       \fB-\fR
311 \fBpthread_rwlock_tryrdlock()\fR        \fBrw_tryrdlock()\fR    \fB-\fR
312 \fBpthread_rwlock_wrlock()\fR   \fBrw_wrlock()\fR       \fB-\fR
313 \fBpthread_rwlock_trywrlock()\fR        \fBrw_trywrlock()\fR    \fB-\fR
314 \fBpthread_rwlock_unlock()\fR   \fBrw_unlock()\fR       \fB-\fR
315 \fBpthread_rwlock_destroy()\fR  \fBrwlock_destroy()\fR  \fB-\fR
316 \fBpthread_rwlockattr_init()\fR \fB-\fR \fB-\fR
317 \fBpthread_rwlockattr_destroy()\fR      \fB-\fR \fB-\fR
```



```
318 \fBpthread_rwlockattr_getpshared()\fR   \fB-\fR \fB-\fR
319 \fBpthread_rwlockattr_setpshared()\fR   \fB-\fR \fB-\fR
320 .TE

322 .SS "Functions Related to Semaphores"

324 .TS
325 l l l
326 l l l .
327 \fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
328 \fBsem_init()\fR        \fBsema_init()\fR       \fB-\fR
329 \fBsem_open()\fR        \fB-\fR \fB-\fR
330 \fBsem_close()\fR       \fB-\fR \fB-\fR
331 \fBsem_wait()\fR        \fBsema_wait()\fR       \fB-\fR
339 \fIsem_wait()\fR        \fIsema_wait()\ \fI-\fR
332 \fBsem_trywait()\fR     \fBsema_trywait()\fR    \fB-\fR
333 \fBsem_post()\fR        \fBsema_post()\fR       \fB-\fR
334 \fBsem_getvalue()\fR    \fB-\fR \fB-\fR
335 \fBsem_unlink()\fR      \fB-\fR \fB-\fR
336 \fBsem_destroy()\fR     \fBsema_destroy()\fR    \fB-\fR
337 .TE

339 .SS "Functions Related to fork(\|) Clean Up"

341 .TS
342 l l l
343 l l l .
344 \fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
345 \fBpthread_atfork()\fR  \fB-\fR \fB-\fR
346 .TE

348 .SS "Functions Related to Limits"

350 .TS
351 l l l
352 l l l .
353 \fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
354 \fBpthread_once()\fR    \fB-\fR \fBcall_once()\fR
355 .TE

357 .SS "Functions Related to Debugging"

359 .TS
360 l l l
361 l l l .
362 \fBPOSIX\fR     \fBillumos\fR   \fBC11\fR
363 \fB-\fR \fBthr_stksegment()\fR  \fB-\fR
364 .TE

366 .SH LOCKING
367 .SS "Synchronization"
376 .LP
368 Multithreaded behavior is asynchronous, and therefore,  optimized for
369 concurrent and parallel processing. As threads, always from within the same
370 process and  sometimes from multiple processes, share global data with each
371 other, they are not guaranteed exclusive access to the shared data at any point
372 in time. Securing mutually exclusive access to shared data requires
373 synchronization among the threads. Both POSIX and illumos implement four
374 synchronization mechanisms: mutexes, condition variables, reader/writer locking
375 (\fIoptimized frequent-read occasional-write mutex\fR), and semaphores, where as
376 C11 threads only implement two mechanisms: mutexes and condition variables.
377 .sp
378 .LP
379 Synchronizing multiple threads diminishes their concurrency. The coarser the
380 grain of synchronization, that is, the larger the block of code that is locked,
381 the lesser the concurrency.
```

```
.SS "MT \fBfork()\fR"
.LP
If a threads program calls \fBfork\fR(2), it implicitly calls \fBfork1\fR(2),
which replicates only the calling thread. Should there be any outstanding
mutexes throughout the process, the application should call
\fBpthread_atfork\fR(3C) to wait for and acquire those mutexes prior to calling
\fBfork()\fR.
.SH SCHEDULING
.SS "POSIX Threads"
.LP
illumos supports the following three POSIX scheduling policies:
.sp
.ne 2
.na
\fB\fBSCHED_OTHER\fR\fR
.ad
.RS 15n
Traditional Timesharing scheduling policy. It is based on the timesharing (TS)
scheduling class.
.RE

.sp
.ne 2
.na
\fB\fBSCHED_FIFO\fR\fR
.ad
.RS 15n
First-In-First-Out scheduling policy. Threads scheduled to this policy, if not
preempted by a higher priority, will proceed until completion. Such threads are
in real-time (RT) scheduling class. The calling process must have a effective
user \fBID\fR of \fB0\fR.
.RE

.sp
.ne 2
.na
\fB\fBSCHED_RR\fR\fR
.ad
.RS 15n
Round-Robin scheduling policy. Threads scheduled to this policy, if not
preempted by a higher priority, will execute for a time period determined by
the system. Such threads are in real-time (RT) scheduling class and the calling
process must have a effective user \fBID\fR of \fB0\fR.
.RE

.sp
.LP
In addition to the POSIX-specified scheduling policies above, illumos also
supports these scheduling policies:
.sp
.ne 2
.na
\fB\fBSCHED_IA\fR\fR
.ad
.RS 13n
Threads are scheduled according to the Inter-Active Class (IA) policy as
described in \fBpriocntl\fR(2).
.RE

.sp
.ne 2
.na
\fB\fBSCHED_FSS\fR\fR
.ad
.RS 13n
Threads are scheduled according to the Fair-Share Class (FSS) policy as
```



```
described in \fBpriocntl\fR(2).
.RE

.sp
.ne 2
.na
\fB\fBSCHED_FX\fR\fR
.ad
.RS 13n
Threads are scheduled according to the Fixed-Priority Class (FX) policy as
described in \fBpriocntl\fR(2).
.RE

.SS "illumos Threads"
.LP
Only scheduling policy supported is \fBSCHED_OTHER\fR, which is timesharing,
based on the \fBTS\fR scheduling class.
.SH ERRORS
.LP
In a multithreaded application, \fBEINTR\fR can be returned from blocking
system calls when another thread calls \fBforkall\fR(2).
.SH USAGE
.SS "\fB-mt\fR compiler option"
.LP
The \fB-mt\fR compiler option compiles and links for multithreaded code. It
compiles source files with \(mi\fBD_REENTRANT\fR and augments the set of
support libraries properly.
.sp
.LP
Users of other compilers such as gcc and clang should manually set
\(mi\fBD_REENTRANT\fR on the compilation line. There are no other libraries or
flags necessary.
.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
_
MT-Level        MT-Safe, Fork 1-Safe
.TE

.SH SEE ALSO
.LP
\fBcrle\fR(1), \fBfork\fR(2), \fBpriocntl\fR(2), \fBlibpthread\fR(3LIB),
\fBlibrt\fR(3LIB), \fBlibthread\fR(3LIB), \fBpthread_atfork\fR(3C),
\fBpthread_create\fR(3C), \fBattributes\fR(5), \fBstandards\fR(5)
.sp
.LP
\fILinker and Libraries Guide\fR
```

```
**********************************************************
    3825 Mon Aug 26 06:56:04 2019
new/usr/src/man/man7d/ipmi.7d
11622 clean up rarer mandoc lint warnings
**********************************************************
'\"
.\" CDDL HEADER START
.\"
.\" 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]
.\"
.\" CDDL HEADER END
.\"
.\" Copyright (c) 2012, Joyent, Inc. All Rights Reserved
.\" Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
.\"
.TH IPMI 7D "Oct 31, 2013"
.SH NAME
ipmi \- OpenIPMI compatible IPMI interface driver
.SH SYNOPSIS
.LP
.nf
\fB/dev/ipmi0\fR
.fi

.SH DESCRIPTION
.sp
.LP
The \fBipmi\fR device is a character special file that provides access to the
Intelligent Platform Management Interface for the system.  For more
information on \fBIPMI\fR and to obtain a copy of the \fBIPMI\fR
specification and implementation guidelines, refer to
http://www.intel.com/design/servers/ipmi/.
The driver is adapted from the FreeBSD driver which is in turn adapted from
the Linux driver, however, not all features described in the standard are
supported. The current implementation depends on the \fBsmbios\fR(7d) to
discover the existence of an IPMI device.
.sp
.LP

.SH IOCTLS
.sp
.LP
Sending and receiving messages through the IPMI drivers requires the use of
\fBioctl\fR(2).

The ioctl command codes below are defined in \fBsys/ipmi.h\fR.
The third argument to ioctl should be a pointer to the type indicated.
Currently the following ioctls are supported:
.RS +4
.TP
.ie t \(bu
.el o
IPMICTL_RECEIVE_MSG "struct ipmi_recv"
```



```
.br
Receive a message.
.br
Possible error values:
.RS +8
EAGAIN No messages are in the process queue.
.br
EFAULT An address supplied was invalid.
.br
EMSGSIZE The address could not fit in the message buffer and
will remain in the buffer.
.RE
.RE

.RS +4
.TP
.ie t \(bu
.el o
IPMICTL_RECEIVE_MSG_TRUNC "struct ipmi_recv"
.br
Like IPMICTL_RECEIVE_MSG but if the message cannot fit into the buffer, it
will truncate the contents instead of leaving the data in the buffer.
.RE

.RS +4
.TP
.ie t \(bu
.el o
IPMICTL_SEND_COMMAND "struct ipmi_req"
.br
Send a message to the interface.
.br
Possible error values:
.RS +8
EFAULT An address supplied was invalid
.br
ENOMEM Buffers could not be allowed for the command, out of memory.
.RE
.RE

.RS +4
.TP
.ie t \(bu
.el o
IPMICTL_SET_MY_ADDRESS_CMD "unsigned int"
.br
Set the slave address for source messages.
.RE

.RS +4
.TP
.ie t \(bu
.el o
IPMICTL_GET_MY_ADDRESS_CMD "unsigned int"
.br
Get the slave address for source messages.
.RE

.RS +4
.TP
.ie t \(bu
.el o
IPMICTL_SET_MY_LUN_CMD "unsigned int"
.br
Set the slave LUN for source messages.
.RE
```

```
122 .RS +4
123 .TP
124 .ie t \(bu
125 .el o
126 IPMICTL_GET_MY_LUN_CMD "unsigned int"
127 .br
128 Get the slave LUN for source messages.
129 .RE

131 Stub Only Ioctl

133 .RS +4
134 .TP
135 .ie t \(bu
136 .el o
137 IPMICTL_SET_GETS_EVENTS_CMD int
138 .br
139 Set whether this interface receives events.
140 .RE

142 Unimplemented Ioctls

144 .RS +4
145 .TP
146 .ie t \(bu
147 .el o
148 IPMICTL_REGISTER_FOR_CMD
149 .br
150 Register to receive a specific command
151 .RE

153 .RS +4
154 .TP
155 .ie t \(bu
156 .el o
157 IPMICTL_UNREGISTER_FOR_CMD
158 .br
159 Unregister to receive a specific command
160 .RE

162 .SH SEE ALSO
170 .sp
171 .LP
163 \fBipmitool\fR(1), \fBioctl\fR(2), \fBsmbios\fR(7d)
164 .sp
165 .LP
166 \fIIntelligent Platform Management Interface Specification Second
167 Generation\fR, v2.0 \(em
168 June 12, 2009 Markup
169 .SH NOTES
179 .sp
180 .LP
170 Not all systems include an \fBIPMI\fR.
```

```
**********************************************************
   14416 Mon Aug 26 06:56:04 2019
new/usr/src/man/man7fs/pcfs.7fs
11622 clean up rarer mandoc lint warnings
**********************************************************
   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 PCFS 7FS "Nov 6, 2007"
   7 .SH NAME
   8 pcfs \- FAT formatted file system
   9 .SH SYNOPSIS
  10 .LP
  10 .nf
  11 #include <sys/param.h>
  12 #include <sys/mount.h>
  13 #include <sys/fs/pc_fs.h>

  15 \fBint\fR \fBmount\fR(\fBconst char *\fR\fIspec\fR,
  16      \fBconst char *\fR \fIdir\fR, \fBint\fR  \fImflag\fR,
  17      \fB"pcfs", NULL, 0,\fR \fBchar *\fR\fIoptptr\fR,
  18      \fBint\fR \fIoptlen\fR);
  19 .fi

  21 .SH DESCRIPTION
  23 .sp
  24 .LP
  22 \fBpcfs\fR is a file system type that enables direct access to files on
  23 \fBFAT\fR formatted disks from within the SunOS operating system.
  24 .sp
  25 .LP
  26 Once mounted, \fBpcfs\fR provides standard SunOS file operations and semantics.
  27 Using \fBpcfs\fR, you can create, delete, read, and write files on a \fBFAT\fR
  28 formatted disk. You can also create and delete directories and list files in a
  29 directory.
  30 .sp
  31 .LP
  32 \fBpcfs\fR supports FAT12 (floppies) and FAT16 and FAT32 file systems.
  33 .sp
  34 .LP
  35 \fBpcfs\fR file systems can be force umounted using the \fB-f\fR argument to
  36 \fBumount\fR(1M).
  37 .sp
  38 .LP
  39 The \fBpcfs\fR file system contained on the block special file identified by
  40 \fIspec\fR is mounted on the directory identified by \fIdir\fR. \fIspec\fR and
  41 \fIdir\fR are pointers to pathnames. \fImflag\fR specifies the \fBmount\fR
  42 options. The \fBMS_DATA\fR bit in \fImflag\fR must be set. Mount options can be
  43 passed to \fBpcfs\fR  using the  optptr and optlen  arguments.  See
  44 \fBmount_pcfs\fR(1M) for a list of mount options supported by \fBpcfs\fR.
  45 .sp
  46 .LP
  47 Because FAT formatted media can record file timestamps between January 1st 1980
  48 and December 31st 2127, it's not possible to fully represent UNIX \fBtime_t\fR
  49 in \fBpcfs\fR for 32 bit or 64 bit programs. In particular, if post-2038
  50 timestamps are present on a FAT formatted medium and \fBpcfs\fR returns these,
  51 32bit applications may unexpectedly fail with \fBEOVERFLOW\fR errors. To
  52 prevent this, the default behaviour of \fBpcfs\fR has been modified to clamp
  53 post-2038 timestamps to the latest possible value for a 32bit \fBtime_t\fR,
  54 which is January 19th 2038, 03:14:06 UTC when setting and retrieving file
  55 timestamps. You can override this behavior using the \fBnoclamptime\fR mount
  56 option, as described in \fBmount_pcfs\fR(1M).
  57 .sp
  58 .LP
```



```
  59 Timestamps on FAT formatted media are recorded in local time. If the recording
  60 and  receiving systems use different timezones, the representation of
  61 timestamps shown on the two systems for the same medium might vary. To correct
  62 this,  \fBpcfs\fR  provides a  timezone mount option to force  interpretation
  63 of timestamps as read from a FAT formatted medium in a given  timezone (that of
  64 the recorder). By default, the local timezone of the  receiver is used. See
  65 \fBmount_pcfs\fR(1M) for details.
  66 .sp
  67 .LP
  68 The root directory of a FAT formatted medium has no timestamps and \fBpcfs\fR
  69 returns the time when the mount was done as timestamp  for the root of the
  70 filesystem.
  71 .sp
  72 .LP
  73 The FAT filesystem doesn't support multiple links. As a result, the link count
  74 for all files and directories in \fBpcfs\fR is hard-coded as "1."
  75 .SS "Mounting File Systems"
  79 .sp
  80 .LP
  76 Use the following command to mount \fBpcfs\fR from diskette:
  77 .sp
  78 .in +2
  79 .nf
  80 mount \fB-F\fR pcfs \fIdevice-special\fR \fIdirectory-name\fR
  81 .fi
  82 .in -2
  83 .sp

  85 .sp
  86 .LP
  87 You can use:
  88 .sp
  89 .in +2
  90 .nf
  91 mount \fIdirectory-name\fR
  92 .fi
  93 .in -2
  94 .sp

  96 .sp
  97 .LP
  98 if the following line is in your \fB/etc/vfstab\fR file:
  99 .sp
 100 .in +2
 101 .nf
 102 \fIdevice-special\fR - \fIdirectory-name\fRpcfs \(mi no rw
 103 .fi
 104 .in -2

 106 .sp
 107 .LP
 108 Use the following command to mount \fBpcfs\fR from non-diskette media:
 109 .sp
 110 .in +2
 111 .nf
 112 mount \fB-F\fR pcfs \fIdevice-special\fR:\fIlogical-drive\fR \fBdirectory-name\f
 113 .fi
 114 .in -2
 115 .sp

 117 .sp
 118 .LP
 119 You can use:
 120 .sp
 121 .in +2
 122 .nf
```

```
123 mount \fIdirectory-name\fR
124 .fi
125 .in -2
126 .sp

128 .sp
129 .LP
130 if the following line is in your \fB/etc/vfstab\fR file:
131 .sp
132 .in +2
133 .nf
134 \fIdevice-special\fR:\fIlogical_drive\fR \fB\(mi\fR \fIdirectory-name\fR \fBpcfs
135 .fi
136 .in -2

138 .sp
139 .LP
140 \fIdevice-special\fR specifies the special block device file for the diskette
141 (\fB/dev/diskette\fIN\fR\fR) or the entire hard disk
142 (\fB/dev/dsk/c\fIN\fRt\fIN\fRd\fIN\fRp0\fR for a SCSI disk, and
143 \fB/dev/dsk/c\fIN\fRd\fIN\fRp0\fR for \fBIDE\fR disks) or the \fBPCMCIA\fR
144 pseudo-floppy memory card (\fB/dev/dsk/c\fIN\fRt\fIN\fRd\fIN\fRs\fIN\fR\fR).
145 .sp
146 .LP
147 \fIlogical-drive\fR specifies either the \fBDOS\fR logical drive letter
148 (\fBc\fR through \fBz\fR) or a drive number (\fB1\fR through \fB24\fR). Drive
149 letter \fBc\fR is equivalent to drive number \fB1\fR and represents the Primary
150 \fBDOS\fR partition on the disk; drive letters \fBd\fR through \fBz\fR are
151 equivalent to drive numbers \fB2\fR through \fB24\fR, and represent \fBDOS\fR
152 drives within the Extended \fBFAT\fR partition. Note that \fIdevice-special\fR
153 and \fIlogical-drive\fR must be separated by a colon.
154 .sp
155 .LP
156 \fIdirectory-name\fR specifies the location where the file system is mounted.
157 .sp
158 .LP
159 For example, to mount the Primary \fBDOS\fR partition from a SCSI hard disk,
160 use:
161 .sp
162 .in +2
163 .nf
164 mount \fB-F\fR pcfs /dev/dsk/c\fIN\fRt\fIN\fRd\fIN\fRp0:c /pcfs/c
165 .fi
166 .in -2
167 .sp

169 .sp
170 .LP
171 To mount the first logical drive in the Extended \fBDOS\fR partition from an
172 \fBIDE\fR hard disk, use:
173 .sp
174 .in +2
175 .nf
176 mount \fB-F\fR pcfs /dev/dsk/c\fIN\fRd\fIN\fRp0:d /pcfs/d
177 .fi
178 .in -2
179 .sp

181 .sp
182 .LP
183 To mount a \fBDOS\fR diskette in the first floppy drive when volume management
184 is not running use:
185 .sp
186 .in +2
187 .nf
188 mount \fB-F\fR pcfs /dev/diskette /pcfs/a
```



```
189 .fi
190 .in -2
191 .sp

193 .sp
194 .LP
195 If Volume Management is running, run \fBvolcheck\fR(1) to automatically mount
196 the floppy and some removable disks.
197 .sp
198 .LP
199 To mount a \fBPCMCIA\fR pseudo-floppy memory card, with Volume Management not
200 running (or not managing the \fBPCMCIA\fR media), use:
201 .sp
202 .in +2
203 .nf
204 mount \fB-F\fR pcfs /dev/dsk/c\fIN\fRt\fIN\fRd\fIN\fRs\fIN\fR /pcfs
205 .fi
206 .in -2
207 .sp

209 .SS "Conventions"
215 .sp
216 .LP
210 Files and directories created through \fBpcfs\fR must comply with either the
211 \fBFAT\fR short file name convention or the long file name convention
212 introduced with Windows 95. The \fBFAT\fR short file name convention is of the
213 form \fIfilename\fR[.\fIext\fR], where \fIfilename\fR generally consists of
214 from one to eight upper-case characters, while the optional \fIext\fR consists
215 of from one to three upper-case characters.
216 .sp
217 .LP
218 The long file name convention is much closer to Solaris file names. A long file
219 name can consist of any characters valid in a short file name, lowercase
220 letters, non-leading spaces, the characters \fB+,;=[]\fR, any number of
221 periods, and can be up to 255 characters long. Long file names have an
222 associated short file name for systems that do not support long file names
223 (including earlier releases of Solaris). The short file name is not visible if
224 the system recognizes long file names. \fBpcfs\fR generates a unique short name
225 automatically when creating a long file name.
226 .sp
227 .LP
228 Given a long file name such as \fBThis is a really long filename.TXT\fR, the
229 short file name will generally be of the form \fBTHISIS~\fR\fIN\fR\fB\&.TXT\fR,
230 where \fIN\fR is a number. The long file name will probably get the short name
231 \fBTHISIS~1.TXT\fR, or \fBTHISIS~2.TXT\fR if \fBTHISIS~1.TXT\fR already exits
232 (or \fBTHISIS~3.TXT\fR if both exist, and so forth). If you use \fBpcfs\fR file
233 systems on systems that do not support long file names, you may want to
234 continue following the short file name conventions. See \fBEXAMPLES\fR.
235 .sp
236 .LP
237 When creating a file name, \fBpcfs\fR creates a short file name if it fits the
238 \fBFAT\fR short file name format, otherwise it creates a long file name. This
239 is because long file names take more directory space. Because the root
240 directory of a \fBpcfs\fR file system is fixed size, long file names in the
241 root directory should be avoided if possible.
242 .sp
243 .LP
244 When displaying file names, \fBpcfs\fR shows them exactly as they are on the
245 media. This means that short names are displayed as uppercase and long file
246 names retain their case. Earlier versions of \fBpcfs\fR folded all names to
247 lowercase, which can be forced with the \fBPCFS_MNT_FOLDCASE\fR mount option.
248 All file name searches within \fBpcfs\fR, however, are treated as if they were
249 uppercase, so \fBreadme.txt\fR and \fBReAdMe.TxT\fR refer to the same file.
250 .sp
251 .LP
252 To format a diskette or a \fBPCMCIA\fR pseudo-floppy memory card in \fBFAT\fR
```

```
253 format in the SunOS system, use either the \fBfdformat\fR \fB-d\fR or the
254 \fBDOS\fR \fBFORMAT\fR command.
255 .SS "Boot Partitions"
263 .sp
264 .LP
256 On x86 systems, hard drives may contain an \fBfdisk\fR partition reserved for
257 the Solaris boot utilities. These partitions are special instances of
258 \fBpcfs\fR. You can mount an x86 boot partition with the command:
259 .sp
260 .in +2
261 .nf
262 mount \fB-F\fR pcfs \fIdevice-special\fR:boot \fIdirectory-name\fR
263 .fi
264 .in -2
265 .sp

267 .sp
268 .LP
269 or you can use:
270 .sp
271 .in +2
272 .nf
273 \fBmount \fIdirectory-name\fR\fR
274 .fi
275 .in -2
276 .sp

278 .sp
279 .LP
280 if the following line is in your \fB/etc/vfstab\fR file:
281 .sp
282 .in +2
283 .nf
284 \fIdevice-special\fR:boot \(mi \fIdirectory-name\fR pcfs \(mi no rw
285 .fi
286 .in -2
287 .sp

289 .sp
290 .LP
291 \fIdevice-special\fR specifies the special block device file for the entire
292 hard disk (\fB/dev/dsk/c\fIN\fRt\fIN\fRd\fIN\fRp0\fR)
293 .sp
294 .LP
295 \fIdirectory-name\fR specifies the location where the file system is mounted.
296 .sp
297 .LP
298 All files on a boot partition are owned by super-user. Only the super-user may
299 create, delete, or modify files on a boot partition.
300 .SH EXAMPLES
310 .LP
301 \fBExample 1 \fRSample Displays of File Names
302 .sp
303 .LP
304 If you copy a file \fBfinancial.data\fR from a UNIX file system to \fBpcfs\fR,
305 it displays as \fBfinancial.data\fR in \fBpcfs\fR, but may show up as
306 \fBFINANC~1.DAT\fR in systems that do not support long file names.

308 .sp
309 .LP
310 The following are legal long file names. They are also \fBillegal\fR short file
311 names:

323 .br
313 .in +2
314 \fBtest.sh.orig\fR
```



```
315 .in -2
316 .br
317 .in +2
318 \fBdata+\fR
319 .in -2
320 .br
321 .in +2
322 \fB\&.login\fR
323 .in -2
324 .sp
325 .LP
326 Other systems that do not support long file names may see:

339 .br
328 .in +2
329 \fBTESTSH~1.ORI\fR
330 .in -2
331 .br
332 .in +2
333 \fBDATA~1\fR
334 .in -2
335 .br
336 .in +2
337 \fBLOGIN~1\fR
338 .in -2
339 .sp
340 .LP
341 The short file name is generated from the initial characters of the long file
342 name, so differentiate names in the first few characters. For example, these
343 names:

357 .br
345 .in +2
346 \fBWorkReport.January.Data\fR
347 .in -2
348 .br
349 .in +2
350 \fBWorkReport.February.Data\fR
351 .in -2
352 .br
353 .in +2
354 \fBWorkReport.March.Data\fR
355 .in -2
356 .sp
357 .LP
358 result in these short names, which are not distinguishable:

373 .br
360 .in +2
361 \fBWORKRE~1.DAT\fR
362 .in -2
363 .br
364 .in +2
365 \fBWORKRE~2.DAT\fR
366 .in -2
367 .br
368 .in +2
369 \fBWORKRE~13.DAT\fR
370 .in -2
371 .sp
372 .LP
373 These names, however:

389 .br
375 .in +2
376 \fBJanuary.WorkReport.Data\fR
```

```
.in -2
.br
.in +2
\fBFebruary.WorkReport.Data\fR
.in -2
.br
.in +2
\fBMarch.WorkReport.Data\fR
.in -2
.sp
.LP
result in the more descriptive short names:

.br
.in +2
\fBJANUAR~1.DAT\fR
.in -2
.br
.in +2
\fBFEBRUA~1.DAT\fR
.in -2
.br
.in +2
\fBMARCHW~1.DAT\fR
.in -2
.SH FILES
.sp
.ne 2
.na
\fB\fB/usr/lib/fs/pcfs/mount\fR\fR
.ad
.RS 26n
\fBpcfs\fR \fBmount\fR command
.RE

.sp
.ne 2
.na
\fB\fB/usr/kernel/fs/pcfs\fR\fR
.ad
.RS 26n
32-bit kernel module
.RE

.SH ENVIRONMENT VARIABLES
.sp
.LP
See \fBenviron\fR(5) for descriptions of the following environment variables
for the current locale setting: LANG, LC_ALL, LC_CTYPE, and LC_COLLATE.
.SH SEE ALSO
.sp
.LP
\fBchgrp\fR(1), \fBchown\fR(1), \fBdos2unix\fR(1), \fBeject\fR(1),
\fBfdformat\fR(1), \fBunix2dos\fR(1), \fBvolcheck\fR(1), \fBmount\fR(1M),
\fBmount_pcfs\fR(1M), \fBumount\fR(1M), \fBctime\fR(3C), \fBvfstab\fR(4),
\fBenviron\fR(5), \fBpcmem\fR(7D)
.SH WARNINGS
.sp
.LP
Do not physically eject a \fBFAT\fR floppy while the device is mounted as
\fBpcfs\fR. If Volume Management is managing a device, use the \fBeject\fR(1)
command before physically removing media.
.sp
.LP
When mounting \fBpcfs\fR on a hard disk, make sure the first block on that
device contains a valid \fBfdisk\fR partition table.
```



```
.sp
.LP
Because \fBpcfs\fR has no provision for handling owner-IDs or group-IDs on
files, \fBchown\fR(1) or \fBchgrp\fR(1) may generate various errors. This is a
limitation of \fBpcfs\fR, but it should not cause problems other than error
messages.
.SH NOTES
.sp
.LP
Only the following characters are allowed in \fBpcfs\fR short file names and
extensions:
.br
.in +2
\fB0-9\fR
.in -2
.br
.in +2
\fBA-Z\fR
.in -2
.br
.in +2
\fB$#&@!%()-{}<>‘_^~|’\fR
.in -2
.sp
.LP
SunOS and \fBFAT\fR use different character sets and have different
requirements for the text file format. Use the \fBdos2unix\fR(1) and
\fBunix2dos\fR(1) commands to convert files between them.
.sp
.LP
\fBpcfs\fR offers a convenient transportation vehicle for files between Sun
workstations and \fBPCs.\fR Because the \fBFAT\fR disk format was designed for
use under \fBDOS,\fR it does not operate efficiently under the SunOS system and
should not be used as the format for a regular local storage. Instead, use
\fBufs\fR for local storage within the SunOS system.
.sp
.LP
Although long file names can contain spaces (just as in UNIX file names), some
utilities may be confused by them.
.sp
.LP
This implementation of \fBpcfs\fR conforms to the behavior exhibited by Windows
95 version 4.00.950.
.sp
.LP
When \fBpcfs\fR encounters long file names with non-ASCII characters, it
converts such long file names in Unicode scalar values into UTF-8 encoded
filenames so that they are legible and usable with any of Solaris UTF-8
locales. In the same context, when new file names with non-ASCII characters are
created, \fBpcfs\fR expects that such file names are in UTF-8. This feature
increases the interoperability of \fBpcfs\fR on Solaris with other operating
systems.
.SH BUGS
.sp
.LP
\fBpcfs\fR should handle the disk change condition in the same way that
\fBDOS\fR does, so you do not need to unmount the file system to change
floppies.
```

```
**********************************************************
   10953 Mon Aug 26 06:56:05 2019
new/usr/src/man/man7i/prnio.7i
11622 clean up rarer mandoc lint warnings
**********************************************************
   1 '\" te
   2 .\"  Copyright (c) 20002 Sun Microsystems, Inc.
   3 .\"  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 PRNIO 7I "Jan 2, 2002"
   8 .SH NAME
   9 prnio \- generic printer interface
  10 .SH SYNOPSIS
  11 .LP
  11 .nf
  12 \fB#include <sys/prnio.h>\fR
  13 .fi

  15 .SH DESCRIPTION
  17 .sp
  18 .LP
  16 The \fBprnio\fR generic printer interface defines ioctl commands and data
  17 structures for printer device drivers.
  18 .sp
  19 .LP
  20 \fBprnio\fR defines and provides facilities for five basic phases of the
  21 printing process:
  22 .RS +4
  23 .TP
  24 .ie t \(bu
  25 .el o
  26 Identification \(em Retrieve device information/attributes
  27 .RE
  28 .RS +4
  29 .TP
  30 .ie t \(bu
  31 .el o
  32 Setup \(em Set device attributes
  33 .RE
  34 .RS +4
  35 .TP
  36 .ie t \(bu
  37 .el o
  38 Transfer \(em Transfer data to or from the device
  39 .RE
  40 .RS +4
  41 .TP
  42 .ie t \(bu
  43 .el o
  44 Cleanup \(em Transfer phase conclusion
  45 .RE
  46 .RS +4
  47 .TP
  48 .ie t \(bu
  49 .el o
  50 Abort \(em Transfer phase interruption
  51 .RE
  52 .sp
  53 .LP
  54 During the Identification phase, the application retrieves a set of device
  55 capabilities and additional information using the \fBPRNIOC_GET_IFCAP\fR,
  56 \fBPRNIOC_GET_STATUS\fR, \fBPRNIOC_GET_TIMEOUTS\fR, \fBPRNIOC_GET_IFINFO\fR and
  57 \fBPRNIOC_GET_1284_DEVID\fR commands.
  58 .sp
```



```
  59 .LP
  60 During the Setup phase the application sets some interface attributes and
  61 probably resets the printer as described in the \fBPRNIOC_SET_IFCAP\fR,
  62 \fBPRNIOC_SET_TIMEOUTS\fR and \fBPRNIOC_RESET\fR sections.
  63 .sp
  64 .LP
  65 During the Transfer phase, data is transferred in a forward (host to
  66 peripheral) or reverse direction (peripheral to host). Transfer is accomplished
  67 using \fBwrite\fR(2) and \fBread\fR(2) system calls. For \fBprnio\fR compliant
  68 printer drivers, forward transfer support is mandatory, while reverse transfer
  69 support is optional. Applications can also use \fBPRNIOC_GET_STATUS\fR and
  70 \fBPRNIOC_GET_1284_STATUS\fR commands during the transfer to monitor the device
  71 state.
  72 .sp
  73 .LP
  74 The Cleanup phase is accomplished by closing the device using \fBclose\fR(2).
  75 Device drivers supporting \fBprnio\fR may set non-zero error code as
  76 appropriate. Applications should explicitly \fBclose\fR(2) a device before
  77 exiting and check \fBerrno\fR value.
  78 .sp
  79 .LP
  80 The Abort phase is accomplished by interrupting the \fBwrite\fR(2) and
  81 \fBread\fR(2) system calls. The application can perform some additional cleanup
  82 during the Abort phase as described in \fBPRNIOC_GET_IFCAP\fR section.
  83 .SH IOCTLS
  87 .sp
  84 .ne 2
  85 .na
  86 \fB\fBPRNIOC_GET_IFCAP\fR\fR
  87 .ad
  88 .RS 21n
  89 Application can retrieve printer interface capabilities using this command. The
  90 \fBioctl\fR(2) argument is a pointer to \fBuint_t\fR, a bit field representing
  91 a set of properties and services provided by a printer driver. Set bit means
  92 supported capability. The following values are defined:
  93 .br
  94 .in +2
  95 \fBPRN_BIDI\fR - When this bit is set,  the interface operates in a
  96 bidirectional mode,  instead of forward-only mode.
  97 .in -2
  98 .br
  99 .in +2
 100 \fBPRN_HOTPLUG\fR - If this bit is set, the interface allows device
 101 hot-plugging.
 102 .in -2
 103 .br
 104 .in +2
 105 \fBPRN_1284_DEVID\fR - If this bit is set, the device is capable of returning
 106 \fI1284\fR device ID (see \fBPRNIOC_GET_1284_DEVID\fR.)
 107 .in -2
 108 .br
 109 .in +2
 110 \fBPRN_1284_STATUS\fR - If this bit is set, the device driver can return device
 111 status lines (see \fBPRNIOC_GET_1284_STATUS\fR). Some devices support this
 112 ioctl in unidirectional mode only.
 113 .in -2
 114 .br
 115 .in +2
 116 \fBPRN_TIMEOUTS\fR - If this bit is set the peripheral may stall during the
 117 transfer phase and the driver can timeout and return from the \fBwrite\fR(2)
 118 and \fBread\fR(2) returning the number of bytes that have been transferred. If
 119 \fBPRN_TIMEOUTS\fR is set, the driver supports this functionality and the
 120 timeout values can be retrieved and modified via the \fBPRNIOC_GET_TIMEOUTS\fR
 121 and \fBPRNIOC_SET_TIMEOUTS\fR ioctls. Otherwise, applications can implement
 122 their own timeouts and abort phase.
 123 .in -2
```

```
 124 .br
 125 .in +2
 126 \fBPRN_STREAMS\fR - This bit impacts the application abort phase behaviour. If
 127 the device claimed \fBPRN_STREAMS\fR capability, the application must issue an
 128 \fBI_FLUSH\fR \fBioctl\fR(2) before \fBclose\fR(2) to dismiss the untransferred
 129 data. Only STREAMS drivers can support this capability.
 130 .in -2
 131 .RE

 133 .sp
 134 .ne 2
 135 .na
 136 \fBPRNIOC_SET_IFCAP\fR
 137 .ad
 138 .RS 21n
 139 This ioctl can be used to change interface capabilities. The argument is a
 140 pointer to \fBuint_t\fR bit field that is described in detail in the
 141 \fBPRNIOC_GET_IFCAP\fR section. Capabilities should be set one at a time;
 142 otherwise the command will return \fBEINVAL\fR. The following capabilities can
 143 be changed by this ioctl:
 144 .br
 145 .in +2
 146 \fBPRN_BIDI\fR - When this capability is set, the interface operates in a
 147 bidirectional mode, instead of forward-only mode. Devices that support only one
 148 mode will not return error; applications should use \fBPRNIOC_GET_IFCAP\fR to
 149 check if the mode was successfully changed. Because some capabilities may be
 150 altered as a side effect of changing other capabilities, this command should be
 151 followed by \fBPRNIOC_GET_IFCAP\fR.
 152 .in -2
 153 .RE

 155 .sp
 156 .ne 2
 157 .na
 158 \fBPRNIOC_GET_IFINFO\fR
 159 .ad
 160 .RS 21n
 161 This command can be used to retrieve printer interface info string, which is an
 162 arbitrary format string usually describing the bus type. The argument is a
 163 pointer to \fBstruct prn_interface_info\fR as described below.
 164 .RE

 166 .sp
 167 .in +2
 168 .nf
 169 struct prn_interface_info {
 170   uint_t          if_len;   /* length of buffer */
 171   uint_t          if_rlen;  /* actual info length */
 172   char   *if_data;  /* buffer address */
 173 };
_____unchanged_portion_omitted_
 246 .fi
 247 .in -2

 249 .sp
 250 .LP
 251 For convenience, the two-byte length field is not considered part of device ID
 252 string and is not returned in the user buffer. Instead, \fBid_rlen\fR  value
 253 shall be set to (length - 2) by the driver, where length is the ID length field
 254 value. If buffer length is less than \fBid_rlen\fR, the driver returns the
 255 first \fBid_len\fR bytes of the ID.
 256 .sp
 257 .LP
 258 The printer driver must return the most up-to-date value of the device ID.
 259 .sp
 260 .ne 2
```



```
 261 .na
 262 \fBPRNIOC_GET_STATUS\fR
 263 .ad
 264 .RS 21n
 265 This command can be used by applications to retrieve current device status. The
 266 argument is a pointer to \fBuint_t\fR, where the status word is returned.
 267 Status is a combination of the following bits:
 268 .RE

 274 .br
 270 .in +2
 271 \fBPRN_ONLINE\fR - For devices that support \fBPRN_HOTPLUG\fR capability,
 272 this bit is set when the device is online, otherwise the device is offline.
 273 Devices without \fBPRN_HOTPLUG\fR support should always have this bit set.
 274 .in -2
 275 .br
 276 .in +2
 277 \fBPRN_READY\fR - This bit indicates if the device is ready to receive/send
 278 data. Applications may use this bit for an outbound flow control
 279 .in -2
 280 .sp
 281 .ne 2
 282 .na
 283 \fB\fBPRNIOC_GET_1284_STATUS\fR\fR
 284 .ad
 285 .RS 26n
 286 Devices that support \fBPRN_1284_STATUS\fR capability accept this ioctl to
 287 retrieve the device status lines defined in \fIIEEE 1284\fR for use in
 288 Compatibility mode. The following bits may be set by the driver:
 289 .br
 290 .in +2
 291 \fBPRN_1284_NOFAULT\fR - Device is not in error state
 292 .in -2
 293 .br
 294 .in +2
 295 \fBPRN_1284_SELECT\fR - Device is selected
 296 .in -2
 297 .br
 298 .in +2
 299 \fBPRN_1284_PE\fR - Paper error
 300 .in -2
 301 .br
 302 .in +2
 303 \fBPRN_1284_BUSY\fR - Device is busy
 304 .in -2
 305 .RE

 307 .sp
 308 .ne 2
 309 .na
 310 \fB\fBPRNIOC_GET_TIMEOUTS\fR\fR
 311 .ad
 312 .RS 26n
 313 This command  retrieves current transfer timeout values for  the driver. The
 314 argument is  a pointer to \fBstruct prn_timeouts\fR as described below.
 315 .RE

 317 .sp
 318 .in +2
 319 .nf
 320 struct prn_timeouts {
 321   uint_t        tmo_forward;  /* forward transfer timeout */
 322   uint_t        tmo_reverse;  /* reverse transfer timeout */
 323         };
 324 .fi
 325 .in -2
```

```
.sp
.LP
\fBtmo_forward\fR and \fBtmo_reverse\fR define forward and reverse transfer
timeouts in seconds. This command is only valid for drivers that support
\fBPRN_TIMEOUTS\fR capability.
.sp
.ne 2
.na
\fB\fBPRNIOC_SET_TIMEOUTS\fR\fR
.ad
.RS 23n
This command sets current transfer  timeout values for the driver. The
argument is  a pointer to \fBstruct prn_timeouts\fR. See
\fBPRNIOC_GET_TIMEOUTS\fR for description of  this structure. This  command is
only valid for drivers that support \fBPRN_TIMEOUTS\fR capability.
.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
_
Architecture     SPARC, IA
_
Interface Stability     Evolving
.TE

.SH SEE ALSO
.sp
.LP
\fBclose\fR(2), \fBioctl\fR(2), \fBread\fR(2), \fBwrite\fR(2),
\fBattributes\fR(5), \fBecpp\fR(7D), \fBusbprn\fR(7D), \fBlp\fR(7D)
.sp
.LP
\fIIEEE Std 1284-1994\fR
```