```
**********************************************************
    7190 Fri Jan 11 21:14:00 2019
new/usr/src/man/man1/expand.1
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
.\"
.\" 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) 1995, Sun Microsystems, Inc.  All Rights Reserved.
.\"
.TH EXPAND 1 "Feb 1, 1995"
.SH NAME
expand, unexpand \- expand TAB characters to SPACE characters, and vice versa
.SH SYNOPSIS
.LP
.nf
\fBexpand\fR [\fB-t\fR \fItablist\fR] [\fIfile\fR]...
.fi

.LP
.nf
\fBexpand\fR [\fB-\fItabstop\fR\fR] [\fB-\fItab1,\fR\fR \fItab2,.\fR \fI\&.\fR \
.fi
```



```
.LP
.nf
\fBunexpand\fR [\fB-a\fR] [\fB-t\fR \fItablist\fR] [\fIfile\fR]...
.fi

.SH DESCRIPTION
.sp
.LP
The \fBexpand\fR utility copies \fIfile\fRs (or the standard input) to the
standard output, with TAB characters expanded to SPACE characters. BACKSPACE
characters are preserved into the output and decrement the column count for TAB
calculations. \fBexpand\fR is useful for pre-processing character files (before
sorting, looking at specific columns, and so forth) that contain TAB
characters.
.sp
.LP
\fBunexpand\fR copies \fIfile\fRs (or the standard input) to the standard
output, putting TAB characters back into the data. By default, only leading
SPACE and TAB characters are converted to strings of tabs, but this can be
overridden by the \fB-a\fR option (see the OPTIONS section below).
.SH OPTIONS
.sp
.LP
The following options are supported for \fBexpand\fR:
.sp
.ne 2
.na
\fB\fB-t\fR \fItablist\fR \fR
.ad
.RS 26n
Specifies the tab stops. The argument \fItablist\fR must consist of a single
positive decimal integer or multiple positive decimal integers, separated by
blank characters or commas, in ascending order. If a single number is given,
tabs will be set \fItablist\fR column positions apart instead of the default
\fB8\fR. If multiple numbers are given, the tabs will be set at those specific
column positions.
.sp
Each tab-stop position \fIN\fR must be an integer value greater than zero, and
the list must be in strictly ascending order. This is taken to mean that, from
the start of a line of output, tabbing to position \fIN\fR causes the next
character output to be in the (\fIN\fR+1)th column position on that line.
.sp
In the event of \fBexpand\fR having to process a tab character at a position
beyond the last of those specified in a multiple tab-stop list, the tab
character is replaced by a single space character in the output.
.RE

.sp
.ne 2
.na
\fB\fB-\fR\fItabstop\fR \fR
.ad
.RS 26n
Specifies as a single argument, sets TAB characters \fItabstop\fR SPACE
characters apart instead of the default \fB8\fR.
.RE

.sp
.ne 2
.na
\fB\fB-\fR\fItab1\fR\fI,\|tab2,...,tabn\fR \fR
.ad
.RS 26n
Sets TAB characters at the columns specified by
\fB-\fR\fItab1,tab2,...,\|tabn\fR
```

```
123 .RE

125 .sp
126 .LP
127 The following options are supported for \fBunexpand\fR:
128 .sp
129 .ne 2
130 .na
131 \fB\fB-a\fR \fR
132 .ad
133 .RS 15n
134 Inserts TAB characters when replacing a run of two or more SPACE characters
135 would produce a smaller output file.
136 .RE

138 .sp
139 .ne 2
140 .na
141 \fB\fB-t\fR \fItablist\fR \fR
142 .ad
143 .RS 15n
144 Specifies the tab stops. The option-argument \fItablist\fR must be a single
145 argument consisting of a single positive decimal integer or multiple positive
146 decimal integers, separated by blank characters or commas, in ascending order.
147 If a single number is given, tabs will be set \fItablist\fR column positions
148 apart instead of the default \fB8\fR. If multiple numbers are given, the tabs
149 will be set at those specific column positions. Each tab-stop position \fIN\fR
150 must be an integer value greater than zero, and the list must be in strictly
151 ascending order. This is taken to mean that, from the start of a line of
152 output, tabbing to position \fIN\fR will cause the next character output to be
153 in the (\fIN\fR+1)th column position on that line. When the \fB-t\fR option is
154 not specified, the default is the equivalent of specifying \fB-t\fR \fB8\fR
155 (except for the interaction with \fB-a\fR, described below).
156 .sp
157 No space-to-tab character conversions occur for characters at positions beyond
158 the last of those specified in a multiple tab-stop list.
159 .sp
160 When \fB-t\fR is specified, the presence or absence of the \fB-a\fR option is
161 ignored; conversion will not be limited to the processing of leading blank
162 characters.
163 .RE

165 .SH OPERANDS
168 .sp
166 .LP
167 The following operand is supported for \fBexpand\fR and \fBunexpand\fR:
170 The following ooperand is supported for \fBexpand\fR and \fBunexpand\fR:
168 .sp
169 .ne 2
170 .na
171 \fB\fB\fIfile\fR\fR \fR
172 .ad
173 .RS 9n
174 The path name of a text file to be used as input.
175 .RE

177 .SH ENVIRONMENT VARIABLES
181 .sp
178 .LP
179 See \fBenviron\fR(5) for descriptions of the following environment variables
180 that affect the execution of \fBexpand\fR and \fBunexpand\fR: \fBLANG\fR,
181 \fBLC_ALL\fR, \fBLC_CTYPE\fR, \fBLC_MESSAGES\fR, and \fBNLSPATH\fR.
182 .SH EXIT STATUS
187 .sp
183 .LP
184 The following exit values are returned:
```



```
185 .sp
186 .ne 2
187 .na
188 \fB\fB0\fR \fR
189 .ad
190 .RS 7n
191 Successful completion
192 .RE

194 .sp
195 .ne 2
196 .na
197 \fB\fB>0\fR \fR
198 .ad
199 .RS 7n
200 An error occurred.
201 .RE

203 .SH ATTRIBUTES
209 .sp
204 .LP
205 See \fBattributes\fR(5) for descriptions of the following attributes:
206 .sp

208 .sp
209 .TS
210 box;
211 c | c
212 l | l .
213 ATTRIBUTE TYPE  ATTRIBUTE VALUE
214 _
215 CSI     enabled
216 _
217 Interface Stability     Standard
218 .TE

220 .SH SEE ALSO
227 .sp
221 .LP
222 \fBtabs\fR(1), \fBattributes\fR(5), \fBenviron\fR(5), \fBstandards\fR(5)
```

```
**********************************************************
   15595 Fri Jan 11 21:14:00 2019
new/usr/src/man/man1/kbd.1
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
   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 KBD 1 "Jan 29, 2007"
   7 .SH NAME
   8 kbd \- manipulate the state of keyboard, or display the type of keyboard, or
   9 change the default keyboard abort sequence effect
  10 .SH SYNOPSIS
  11 .LP
  12 .nf
  13 \fBkbd\fR [\fB-r\fR] [\fB-t\fR ] [\fB-l\fR] [\fB-a\fR enable | disable | alterna
  14      [\fB-c\fR on | off] [\fB-d\fR \fIkeyboard device\fR]
  15      [\fB-D\fR \fIautorepeat delay\fR] [\fB-R\fR \fIautorepeat rate\fR]
  16 .fi

  18 .LP
  19 .nf
  20 \fBkbd\fR [\fB-i\fR] [\fB-d\fR \fIkeyboard device\fR]
  21 .fi

  23 .LP
  24 .nf
  25 \fBkbd\fR \fB-s\fR [\fIlanguage\fR]
  26 .fi

  28 .LP
  29 .nf
  30 \fBkbd\fR \fB-b\fR [\fBkeyboard\fR | \fBconsole\fR] \fIfrequency\fR
  31 .fi

  33 .SH DESCRIPTION
  34 .LP
  35 The \fBkbd\fR utility manipulates the state of the keyboard, or displays the
  36 keyboard type, or allows the default keyboard abort sequence effect to be
  37 changed. The abort sequence also applies to serial console devices. The
  38 \fBkbd\fR utility sets the \fB/dev/kbd\fR default keyboard device.
  39 .SH EXTENDED DESCRIPTION
  40 .LP
  41 The \fB-i\fR option reads and processes default values for the keyclick and
  42 keyboard abort settings from the \fB/etc/default/kbd\fR keyboard default file.
  43 Only keyboards that support a clicker respond to the \fB-c\fR option. To turn
  44 clicking on by default, add or change the value of the \fBKEYCLICK\fR variable
  45 in the \fB/etc/default/kbd\fR file to:
  46 .sp
  47 .in +2
  48 .nf
  49 KEYCLICK=on
  50 .fi
  51 .in -2
  52 .sp

  54 .sp
  55 .LP
  56 Next, run the command \fBkbd\fR \fB-i\fR to change the setting. Valid settings
  57 for the \fBKEYCLICK\fR variable are \fBon\fR and \fBoff\fR; all other values
  58 are ignored. If the \fBKEYCLICK\fR variable is not specified in the default
```



```
  59 file, the setting is unchanged.
  60 .sp
  61 .LP
  62 The keyboard abort sequence effect can only be changed by a super-user using
  63 the \fB-a\fR option. This sequence is typically Stop-A or L1-A and Shift-Pause
  64 on the keyboard on \fBSPARC\fR systems, F1-A and Shift-Pause on x86 systems,
  65 and BREAK on the serial console input device on most systems.
  66 .sp
  67 .LP
  68 A \fBBREAK\fR condition that originates from an erroneous electrical signal
  69 cannot be distinguished from one deliberately sent by remote \fBDCE\fR. As a
  70 remedy, use the \fB-a\fR option with Alternate Break to switch break
  71 interpretation. Due to the risk of incorrect sequence interpretation, binary
  72 protocols such as \fBSLIP\fR and others should not be run over the serial
  73 console port when Alternate Break sequence is in effect.
  74 .sp
  75 .LP
  76 Although PPP is a binary protocol, it has the ability to avoid using characters
  77 that interfere with serial operation. The default alternate break sequence is
  78 CTRL-m \fB~\fR CTRL-b, or \fB0D 7E 02\fR in hexadecimal. In PPP, this can be
  78 CTRL-m \fB~\fR CTRL-b, or \fB0D 7E 02\fR in hexidecimal. In PPP, this can be
  79 avoided by setting either \fB0x00000004\fR or \fB0x00002000\fR in the ACCM.
  80 This forces an escape for the CTRL-b or CTRL-m characters, respectively.
  81 .sp
  82 .LP
  83 To do this in Solaris PPP 4.0, add:
  84 .sp
  85 .in +2
  86 .nf
  87 asyncmap 0x00002000
  88 .fi
  89 .in -2
  90 .sp

  92 .sp
  93 .LP
  94 to the \fB/etc/ppp/options\fR file or any of the other configuration files used
  95 for the connection. See \fBpppd\fR(1M).
  96 .sp
  97 .LP
  98 SLIP has no comparable capability, and must not be used if the Alternate Break
  99 sequence is in use.
 100 .sp
 101 .LP
 102 The Alternate Break sequence has no effect on the keyboard abort. For more
 103 information on the Alternate Break sequence, see \fBzs\fR(7D), \fBse\fR(7D),
 104 and \fBasy\fR(7D).
 105 .sp
 106 .LP
 107 On many systems, the default effect of the keyboard abort sequence is to
 108 suspend the operating system and enter the debugger or the monitor. Some
 109 systems feature key switches with a \fBsecure\fR position. On these systems,
 110 setting the key switch to the \fBsecure\fR position overrides any software
 111 default set with this command.
 112 .sp
 113 .LP
 114 To permanently change the software default effect of the keyboard abort
 115 sequence, first add or change the value of the \fBKEYBOARD_ABORT\fR variable in
 116 the \fB/etc/default/kbd\fR file to:
 117 .sp
 118 .in +2
 119 .nf
 120 KEYBOARD_ABORT=disable
 121 .fi
 122 .in -2
 123 .sp
```

```
.sp
.LP
Next, run the command \fBkbd\fR \fB-i\fR to change the setting. Valid settings
are \fBenable\fR, \fBdisable\fR, and \fBalternate\fR; all other values are
ignored. If the variable is not specified in the default file, the setting is
unchanged.
.sp
.LP
To set the abort sequence to the hardware BREAK, set the value of the
\fBKEYBOARD_ABORT\fR variable in the \fB/etc/default/kbd\fR file to:
.sp
.in +2
.nf
KEYBOARD_ABORT=enable
.fi
.in -2
.sp

.sp
.LP
To change the current setting, run the command \fBkbd\fR \fB-i\fR. To set the
abort sequence to the Alternate Break character sequence, first set the current
value of the \fBKEYBOARD_ABORT\fR variable in the \fB/etc/default/kbd\fR file
to:
.sp
.in +2
.nf
KEYBOARD_ABORT=alternate
.fi
.in -2
.sp

.sp
.LP
Next, run the command \fBkbd\fR \fB-i\fR to change the setting. When the
Alternate Break sequence is in effect, only serial console devices are
affected.
.sp
.LP
To set the autorepeat delay by default, set the \fBREPEAT_DELAY\fR variable in
the file \fB/etc/default/kbd\fR to the expected value with units in
milliseconds (ms). To avoid making the keyboard unusable due to a typographical
error, delay values below \fBKIOCRPTDELAY_MIN\fR (defined in
\fB/usr/include/sys/kbio.h\fR) are rejected with \fBEINVAL\fR:
.sp
.in +2
.nf
REPEAT_DELAY=500
.fi
.in -2
.sp

.sp
.LP
To set the autorepeat rate by default, set the \fBREPEAT_RATE\fR variable in
the file \fB/etc/default/kbd\fR to the expected value with units in
milliseconds. Negative and zero repeat rates are ejected with \fBEINVAL\fR:
.sp
.in +2
.nf
REPEAT_RATE=33
.fi
.in -2
.sp
```



```
.sp
.LP
To change the current settings of \fIdelay\fR and \fIrate\fR, run the command,
\fBkbd\fR \fB-i\fR. When the Auto Repeat Delay and/or Auto Repeat Rate are in
effect, only command line mode is affected.
.sp
.LP
To set the language by default, set the \fBLAYOUT\fR variable in the file
\fB/etc/default/kbd\fR to the expected language. These languages supported in
kernel can be found by running \fBkbd\fR \fB-s\fR. Other values are ignored.
For example, the following sets Spanish layout to the keyboard:
.sp
.in +2
.nf
LAYOUT=Spanish
.fi
.in -2
.sp

.sp
.LP
Next, run the \fBkbd\fR \fB-i\fR to change the setting. When Solaris reboots,
the Spanish key table is loaded into the kernel. These layouts are valid for
\fBusb\fR and \fBps/2\fR keyboards.
.sp
.LP
To set the keyboard beeper frequency by default, set the \fBKBD_BEEPER_FREQ\fR
variable in the file \fB/etc/default/kbd\fR to the expected value with units in
HZ. This value should be between 0 and 32767, inclusive. Otherwise will be
rejected with \fBEINVAL\fR:
.sp
.in +2
.nf
KBD_BEEPER_FREQ=2000
.fi
.in -2
.sp

.sp
.LP
To set the console beeper frequency by default, set the
\fBCONSOLE_BEEPER_FREQ\fR variable in the file \fB/etc/default/kbd\fR to the
expected value with units in HZ. This value should be between 0 and 32767,
inclusive. Otherwise will be rejected with \fBEINVAL\fR:
.sp
.in +2
.nf
CONSOLE_BEEPER_FREQ=900
.fi
.in -2
.sp

.sp
.LP
To change the current settings of keyboard beeper frequency and console beeper
frequency, run \fBkbd\fR \fB-i\fR.
.SH OPTIONS
.LP
The following options are supported:
.sp
.ne 2
.na
\fB\fB-a\fR \fBenable\fR | \fBdisable\fR | \fBalternate\fR\fR
.ad
.sp .6
.RS 4n
```

```
256 Enables, disables, or alternates the keyboard abort sequence effect. By
257 default, a keyboard abort sequence suspends the operating system on most
258 systems. This sequence is typically Stop-A or L1-A and Shift-Pause on the
259 keyboard on \fBSPARC\fR systems, F1-A and Shift-Pause on x86 systems, and BREAK
260 on the serial console device.
261 .sp
262 The default keyboard behavior can be changed using this option. The \fB-a\fR
263 option can only be used by a super-user.
264 .sp
265 .ne 2
266 .na
267 \fB\fBenable\fR\fR
268 .ad
269 .RS 13n
270 Enables the default effect of the keyboard abort sequence (suspend the
271 operating system and enter the debugger or the monitor).
272 .RE

274 .sp
275 .ne 2
276 .na
277 \fB\fBdisable\fR\fR
278 .ad
279 .RS 13n
280 Disables the default/alternate effect and ignores keyboard abort sequences.
281 .RE

283 .sp
284 .ne 2
285 .na
286 \fB\fBalternate\fR\fR
287 .ad
288 .RS 13n
289 Enables the alternate effect of the keyboard abort sequences (suspend the
290 operating system and enter the debugger or the monitor) upon receiving the
291 Alternate Break character sequence on the console. The Alternate Break sequence
292 is defined by the drivers \fBzs\fR(7D), \fBse\fR(7D), \fBasy\fR(7D). Due to a
293 risk of incorrect sequence interpretation, binary protocols cannot be run over
294 the serial console port when this value is used.
295 .RE

297 .RE

299 .sp
300 .ne 2
301 .na
302 \fB\fB-b\fR \fBkeyboard\fR | \fBconsole\fR\fR
303 .ad
304 .sp .6
305 .RS 4n
306 Sets the beeper frequency for keyboard or console.
307 .sp
308 .ne 2
309 .na
310 \fB\fBkeyboard\fR\fR
311 .ad
312 .RS 12n
313 Set the keyboard beeper frequency to the operand in HZ. See \fBOPERANDS\fR.
314 .RE

316 .sp
317 .ne 2
318 .na
319 \fB\fBconsole\fR\fR
320 .ad
321 .RS 12n
```



```
322 Sets the console beeper frequency to the operand in HZ. See \fBOPERANDS\fR.
323 .RE

325 .RE

327 .sp
328 .ne 2
329 .na
330 \fB\fB-c\fR \fBon\fR | \fBoff\fR\fR
331 .ad
332 .sp .6
333 .RS 4n
334 Turns the clicking of the keyboard on or off.
335 .sp
336 .ne 2
337 .na
338 \fB\fBon\fR\fR
339 .ad
340 .RS 7n
341 Enables clicking
342 .RE

344 .sp
345 .ne 2
346 .na
347 \fB\fBoff\fR\fR
348 .ad
349 .RS 7n
350 Disables clicking
351 .RE

353 .RE

355 .sp
356 .ne 2
357 .na
358 \fB\fB-d\fR \fIkeyboard device\fR\fR
359 .ad
360 .sp .6
361 .RS 4n
362 Specifies the keyboard device being set. The default setting is \fB/dev/kbd\fR.
363 .RE

365 .sp
366 .ne 2
367 .na
368 \fB\fB-D\fR \fIautorepeat delay\fR\fR
369 .ad
370 .sp .6
371 .RS 4n
372 Sets the autorepeat delay in milliseconds.
373 .RE

375 .sp
376 .ne 2
377 .na
378 \fB\fB-i\fR\fR
379 .ad
380 .sp .6
381 .RS 4n
382 Sets keyboard properties from the keyboard default file. With the exception of
383 \fB-d\fR \fIkeyboard device\fR, this option cannot be used with any other
384 option. The \fB-i\fR option instructs the keyboard command to read and process
385 keyclick and keyboard abort default values from the \fB/etc/default/kbd\fR
386 file. The \fB-i\fR option can only be used by a user or role with the Device
387 Security Rights Profile.
```

```
.RE

.sp
.ne 2
.na
\fB\fB-l\fR\fR
.ad
.sp .6
.RS 4n
Returns the layout code of the keyboard being used, and the autorepeat delay
and autorepeat rate being used.
.sp
If used with -R or -D option, this option returns the value before the changes.
.RE

.sp
.ne 2
.na
\fB\fB-r\fR\fR
.ad
.sp .6
.RS 4n
Resets the keyboard as if power-up.
.RE

.sp
.ne 2
.na
\fB\fB-R\fR \fIautorepeat rate\fR\fR
.ad
.sp .6
.RS 4n
Sets the autorepeat rate in milliseconds.
.RE

.sp
.ne 2
.na
\fB\fB\fR\fB-s\fR \fB[\fR\fIlanguage\fR\fB]\fR\fR
.ad
.sp .6
.RS 4n
Sets the keyboard layout into the kernel.
.sp
If \fIlanguage\fR is specified, the layout is set to \fIlanguage\fR, and
\fBloadkeys\fR(1) runs implicitly. If \fIlanguage\fR is not specified, a list
of available layouts are presented, prompting for the user to specify the
\fIlanguage\fR. See \fBOPERANDS\fR.
.RE

.sp
.ne 2
.na
\fB\fB-t\fR\fR
.ad
.sp .6
.RS 4n
Returns the type of the keyboard being used.
.RE

.SH OPERANDS
.LP
The following operands are supported:
.sp
.ne 2
.na
```



```
\fBfrequency\fR
.ad
.RS 13n
The frequency value specified to be set in kernel. The receiver of this value
is specified by the \fB-b\fR option. This value should be between 0 and 32767
otherwise will be ejected with \fBEINVAL\fR.
.RE

.sp
.ne 2
.na
\fBlanguage\fR
.ad
.RS 13n
The language specified to be set in kernel. If the language is not found, the
languages supported are listed for selection. It only applies to \fB-s\fR
option.
.RE

.SH EXAMPLES
.LP
\fBExample 1 \fRDisplaying the Keyboard Type
.sp
.LP
The following example displays the keyboard type:

.sp
.in +2
.nf
example% kbd -t
Type 4 Sun keyboard
example%
.fi
.in -2
.sp

.LP
\fBExample 2 \fRSetting Keyboard Defaults
.sp
.LP
The following example sets the keyboard defaults as specified in the keyboard
default file:

.sp
.in +2
.nf
example# kbd -i
example#
.fi
.in -2
.sp

.LP
\fBExample 3 \fRDisplaying Information
.sp
.LP
The following example displays keyboard type and layout code. It also displays
auto repeat delay and rate settings.

.sp
.in +2
.nf
example% kbd -l
type=4
layout=43 (0x2b)
delay(ms)=500
```

```
520 rate(ms)=33
521 example%
522 .fi
523 .in -2
524 .sp

526 .LP
527 \fBExample 4 \fRSetting Keyboard Autorepeat Delay
528 .sp
529 .LP
530 The following example sets the keyboard autorepeat delay:

532 .sp
533 .in +2
534 .nf
535 example% kbd -D 300
536 example%
537 .fi
538 .in -2
539 .sp

541 .LP
542 \fBExample 5 \fRSetting Keyboard Autorepeat Rate
543 .sp
544 .LP
545 The following example sets the keyboard autorepeat rate:

547 .sp
548 .in +2
549 .nf
550 example% kbd -R 50
551 example%
552 .fi
553 .in -2
554 .sp

556 .LP
557 \fBExample 6 \fRSelecting and Setting the Keyboard Language
558 .sp
559 .LP
560 The following example selects and sets the keyboard language from a list of
561 languages specified:

563 .sp
564 .in +2
565 .nf
566 example% kbd -s
567 1. Albanian                     16. Malta_UK
568 2. Belarusian                  17. Malta_US
569 3. Belgian                     18. Norwegian
570 4. Bulgarian                  19. Portuguese
571 5. Croatian                   20. Russian
572 6. Danish                     21. Serbia-And-Montenegro
573 7. Dutch                      22. Slove
574 \&......

576 To select the keyboard layout, enter a number [default n]:

578 example%
579 .fi
580 .in -2
581 .sp

583 .sp
584 .LP
585 The following example sets the keyboard language specified:
```



```
587 .sp
588 .in +2
589 .nf
590 example% kbd -s Dutch
591 example%
592 .fi
593 .in -2
594 .sp

596 .LP
597 \fBExample 7 \fRSetting the Keyboard Beeper Frequency
598 .sp
599 .LP
600 The following example sets the keyboard beeper frequency:

602 .sp
603 .in +2
604 .nf
605 example% kbd -b keyboard 1000
606 example%
607 .fi
608 .in -2
609 .sp

611 .SH FILES
612 .ne 2
613 .na
614 \fB\fB/dev/kbd\fR\fR
615 .ad
616 .RS 20n
617 Keyboard device file.
618 .RE

620 .sp
621 .ne 2
622 .na
623 \fB\fB/etc/default/kbd\fR\fR
624 .ad
625 .RS 20n
626 Keyboard default file containing software defaults for keyboard configurations.
627 .RE

629 .SH SEE ALSO
630 .LP
631 \fBloadkeys\fR(1), \fBsvcs\fR(1), \fBinetd\fR(1M), \fBinetadm\fR(1M),
632 \fBkadb\fR(1M), \fBsvcadm\fR(1M), \fBpppd\fR(1M), \fBkeytables\fR(4),
633 \fBattributes\fR(5), \fBsmf\fR(5), \fBkb\fR(7M), \fBzs\fR(7D), \fBse\fR(7D),
634 \fBasy\fR(7D), \fBvirtualkm\fR(7D)
635 .SH NOTES
636 .LP
637 Some server systems have key switches with a \fBsecure\fR key position that can
638 be read by system software. This key position overrides the normal default of
639 the keyboard abort sequence effect and changes the default so the effect is
640 disabled. When the key switch is in the \fBsecure\fR position on these systems,
641 the keyboard abort sequence effect cannot be overridden by the software
642 default, which is settable with the \fBkbd\fR utility.
643 .sp
644 .LP
645 Currently, there is no way to determine the state of the keyboard click
646 setting.
647 .sp
648 .LP
649 The \fBkdb\fR service is managed by the service management facility,
650 \fBsmf\fR(5), under the service identifier:
651 .sp
```

```
652 .in +2
653 .nf
654 svc:/system/keymap:default
655 .fi
656 .in -2
657 .sp

659 .sp
660 .LP
661 Administrative actions on this service, such as enabling, disabling, or
662 requesting restart, can be performed using \fBsvcadm\fR(1M). Responsibility for
663 initiating and restarting this service is delegated to \fBinetd\fR(1M). Use
664 \fBinetadm\fR(1M) to make configuration changes and to view configuration
665 information for this service. The service's status can be queried using the
666 \fBsvcs\fR(1) command.
```

```
**********************************************************
   14755 Fri Jan 11 21:14:00 2019
new/usr/src/man/man1/msgfmt.1
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
```

```
   1 '\" te
   2 .\"  Copyright (c) 2001, Sun Microsystems, Inc. All Rights Reserved
   3 .\" The contents of this file are subject to the terms of the Common Development
   4 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
   5 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
   6 .TH MSGFMT 1 "Sep 17, 2001"
   7 .SH NAME
   8 msgfmt \- create a message object from a message file
   9 .SH SYNOPSIS
  10 .LP
  11 .nf
  12 \fBmsgfmt\fR [\fB-D\fR \fIdir\fR | \fB-\(midirectory\fR=\fIdir\fR]
  13      [\fB-f\fR | \fB-\(miuse-fuzzy\fR] [\fB-g\fR]
  14      [\fB-o\fR \fIoutput-file\fR | \fB-\(mioutput-file\fR=\fIoutput-file\fR]
  15      [\fB-s\fR] [\fB-\(mistrict\fR] [\fB-v\fR] [\fB-\(miverbose\fR] \fIfilename\
  16 .fi

  18 .SH DESCRIPTION
  19 .sp
  19 .LP
  20 The \fBmsgfmt\fR utility creates message object files from portable object
  21 files (\fIfilename\fR\fB\&.po\fR), without changing the portable object files.
  22 .sp
  23 .LP
  24 The \fB\&.po\fR file contains messages displayed to users by system commands or
  25 by application programs. \fB\&.po\fR files can be edited. The messages in these
  26 files can be rewritten in any language supported by the system.
  27 .sp
  28 .LP
  29 The \fBxgettext\fR(1) command can be used to create \fB\&.po\fR files from
  30 script or programs.
  31 .sp
  32 .LP
  33 \fBmsgfmt\fR interprets data as characters according to the current setting of
  34 the \fBLC_CTYPE\fR locale category or according to the codeset specified in the
  35 \fB\&.po\fR file.
  36 .SH OPTIONS
  38 .sp
  37 .LP
  38 The following options are supported:
  39 .sp
  40 .ne 2
  41 .na
  42 \fB\fB-D\fR \fIdir\fR\fR
  43 .ad
  44 .br
  45 .na
  46 \fB\fB-\(midirectory=\fR\fIdir\fR\fR
  47 .ad
  48 .RS 27n
  49 Adds \fIdir\fR to the list for input files search.
  50 .RE

  52 .sp
  53 .ne 2
  54 .na
  55 \fB\fB-f\fR\fR
  56 .ad
```



```
  57 .br
  58 .na
  59 \fB\fB-\(miuse-fuzzy\fR\fR
  60 .ad
  61 .RS 27n
  62 Uses fuzzy entries in output. If this option is not specified, fuzzy entries
  63 are not included into the output. These options are ignored if Solaris message
  64 catalogs are processed.
  65 .RE

  67 .sp
  68 .ne 2
  69 .na
  70 \fB\fB-g\fR\fR
  71 .ad
  72 .RS 27n
  73 Directs the utility to generate the GNU-compatible message catalog file. This
  74 option cannot be specified with the \fB-s\fR option.
  75 .RE

  77 .sp
  78 .ne 2
  79 .na
  80 \fB\fB-o\fR \fIoutput-file\fR\fR
  81 .ad
  82 .br
  83 .na
  84 \fB\fB-\(mioutput=\fR\fIoutput-file\fR\fR
  85 .ad
  86 .RS 27n
  87 Specifies the output file name as \fIoutput-file\fR. All domain directives and
  88 duplicate msgids in the .\fBpo\fR file are ignored.
  89 .RE

  91 .sp
  92 .ne 2
  93 .na
  94 \fB\fB-s\fR\fR
  95 .ad
  96 .RS 27n
  97 Directs the utility to generate the Solaris message catalog file. This option
  98 cannot be specified with the \fB-g\fR option.
  99 .RE

 101 .sp
 102 .ne 2
 103 .na
 104 \fB\fB-\(mistrict\fR\fR
 105 .ad
 106 .RS 27n
 107 Directs the utility to append the suffix \fB\&.mo\fR to the generating message
 108 object file name if it doesn't have this suffix. This option is ignored if
 109 Solaris message catalogs are processed.
 110 .RE

 112 .sp
 113 .ne 2
 114 .na
 115 \fB\fB-v\fR\fR
 116 .ad
 117 .br
 118 .na
 119 \fB\fB-\(miverbose\fR\fR
 120 .ad
 121 .RS 27n
 122 Verbose. Lists duplicate message identifiers if Solaris message catalog files
```

```
123 are processed. Message strings are not redefined.
124 .sp
125 If GNU-compatible message files are processed, this option detects and
126 diagnoses input file anomalies which might represent translation errors. The
127 msgid and msgstr strings are studied and compared. It is considered abnormal if
128 one string starts or ends with a newline while the other does not. Also, if the
129 string represents a format string used in a printf-like function, both strings
130 should have the same number of % format specifiers, with matching types. If the
131 flag \fBc-format\fR appears in the special comment '\fB#\fR' for this entry, a
132 check is performed.
133 .RE

135 .SH USAGE
138 .sp
136 .LP
137 The format of portable object files (\fB\&.po\fR files) is defined as follows.
138 Each \fB\&.po\fR file contains one or more lines, with each line containing
139 either a comment or a statement. Comments start the line with a pound sign
140 (\fB#\fR) and end with the newline character. All comments (except special
141 comments described later) and empty lines are ignored. The format of a
142 statement is:
143 .sp
144 .in +2
145 .nf
146 \fIdirective\fR     \fIvalue\fR
147 .fi
148 .in -2
149 .sp

151 .sp
152 .LP
153 Each \fIdirective\fR starts at the beginning of the line and is separated from
154 \fIvalue\fR by white space (such as one or more space or tab characters).
155 \fIvalue\fR consists of one or more quoted strings separated by white space.
156 Use any of the following types of directives for the Solaris message file:
157 .sp
158 .in +2
159 .nf
160 domain \fIdomainname\fR
161 msgid \fImessage_identifier\fR
162 msgstr \fImessage_string\fR
163 .fi
164 .in -2
165 .sp

167 .sp
168 .LP
169 For a GNU-compatible message file, use any of the following types of
170 directives:
171 .sp
172 .in +2
173 .nf
174 domain \fIdomainname\fR
175 msgid \fImessage_identifier\fR
176 msgid_plural \fIuntranslated_string_plural\fR
177 msgstr \fImessage_string\fR
178 msgstr[\fIn\fR] \fImessage_string\fR
179 .fi
180 .in -2
181 .sp

183 .sp
184 .LP
185 The behavior of the \fBdomain\fR directive is affected by the options used. See
186 OPTIONS for the behavior when the \fB-o\fR or \fB-\(mioutput-file\fR options
187 are specified. If the \fB-o\fR or \fB-\(mioutput-file\fR options are not
```



```
188 specified, the behavior of the \fBdomain\fR directive is as follows:
189 .RS +4
190 .TP
191 .ie t \(bu
192 .el o
193 All msgids from the beginning of each \fB\&.po\fR file to the first
194 \fBdomain\fR directive are put into a default message object file. The default
195 message object file is named \fBmessages.mo\fR, if the Solaris message catalog
196 file format is used to generate the message object file or if the
197 \fB-\(mistrict\fR option is specified. Otherwise, the default message object
198 file is named \fBmessages\fR.
199 .RE
200 .RS +4
201 .TP
202 .ie t \(bu
203 .el o
204 When \fBmsgfmt\fR encounters a \fBdomain\fR \fIdomainname\fR directive in the
205 \fB\&.po\fR file, all following msgids until the next \fBdomain\fR directive
206 are put into the message object file, named \fBdomainname.mo\fR, if the Solaris
207 message catalog file format is used to generate the message object file or if
208 the \fB-\(mistrict\fR option is specified. Otherwise, the msgids are put into
209 the message object file named \fBdomainname\fR.
210 .RE
211 .RS +4
212 .TP
213 .ie t \(bu
214 .el o
215 Duplicate msgids are defined in the scope of each domain. That is, a msgid is
216 considered a duplicate only if the identical msgid exists in the same domain.
217 .RE
218 .RS +4
219 .TP
220 .ie t \(bu
221 .el o
222 All duplicate msgids are ignored.
223 .RE
224 .sp
225 .LP
226 The \fBmsgid\fR directive specifies the value of a message identifier
227 associated with the directive that follows it. The \fBmsgid_plural\fR directive
228 specifies the plural form message specified to the plural message handling
229 functions \fBngettext()\fR, \fBdngettext()\fR, or \fBdcngettext()\fR. The
230 \fImessage_identifier\fR string identifies a target string to be used at
231 retrieval time. Each statement containing a \fBmsgid\fR directive must be
232 followed by a statement containing a \fBmsgstr\fR directive or
233 \fBmsgstr\fR[\fIn\fR] directives.
234 .sp
235 .LP
236 The \fBmsgstr\fR directive specifies the target string associated with the
237 \fImessage_identifier\fR string declared in the immediately preceding
238 \fBmsgid\fR directive.
239 .sp
240 .LP
241 The directive \fBmsgstr\fR[\fIn\fR] (where \fIn\fR = 0, 1, 2, ...) specifies
242 the target string to be used with plural form handling functions
243 \fBngettext()\fR, \fBdngettext()\fR, and \fBdcngettext()\fR.
246 \fBngettext()\fR, \fBdngettext()\fR, and \fBdcngetttext()\fR.
244 .sp
245 .LP
246 Message strings can contain the escape sequences \fB\\n\fR for newline,
247 \fB\\t\fR for tab, \fB\\v\fR for vertical tab, \fB\\b\fR for backspace,
248 \fB\\r\fR for carriage return, \fB\\f\fR for formfeed, \fB\\\fR for backslash,
249 \fB\\"\fR for double quote, \fB\\a\fR for alarm, \fB\\ddd\fR for octal bit
250 pattern, and \fB\\xDD\fR for hexadecimal bit pattern.
251 .sp
252 .LP
```

```
253 Comments for a GNU-compatible message file should be in one of the following
254 formats (the \fBmsgfmt\fR utility will ignore these comments when processing
255 Solaris message files):
256 .sp
257 .in +2
258 .nf
259 # \fItranslator-comments\fR
260 #. \fIautomatic-comments\fR
261 #: \fIreference\fR..
262 #, \fIflag\fR
263 .fi
264 .in -2
265 .sp

267 .sp
268 .LP
269 The ’\fB#:\fR’ comments indicate the location of the msgid string in the source
270 files in \fIfilename\fR:\fIline\fR format. The ’\fB#\fR’, ’\fB#.\fR’,
271 and ’\fB#:\fR’ comments are informative only and are silently ignored by the
272 \fBmsgfmt\fR utility. The ’\fB#,\fR’ comments require one or more flags
273 separated by the comma character. The following \fIflag\fRs can be specified:
274 .sp
275 .ne 2
276 .na
277 \fB\fBfuzzy\fR\fR
278 .ad
279 .RS 15n
280 This flag can be inserted by the translator. It shows that the \fBmsgstr\fR
281 string might not be a correct translation (anymore). Only the translator can
282 judge if the translation requires further modification or is acceptable as is.
283 Once satisfied with the translation, the translator removes this \fBfuzzy\fR
284 flag. If this flag is specified, the \fBmsgfmt\fR utility will not generate the
285 entry for the immediately following msgid in the output message catalog.
286 .RE

288 .sp
289 .ne 2
290 .na
291 \fB\fBc-format\fR\fR
292 .ad
293 .br
294 .na
295 \fB\fBno-c-format\fR\fR
296 .ad
297 .RS 15n
298 The \fBc-format\fR flag indicates that the \fBmsgid\fR string is used as a
299 format string by printf-like functions. In case the \fBc-format\fR flag is
300 given for a string, the \fBmsgfmt\fR utility does some more tests to check the
301 validity of the translation.
302 .RE

304 .sp
305 .LP
306 In the GNU-compatible message file, the \fBmsgid\fR entry with empty string
307 ("") is called the header entry and treated specially. If the message string
308 for the header entry contains \fBnplurals\fR=\fIvalue\fR, the value indicates
309 the number of plural forms. For example, if \fBnplurals\fR=4, there are four
310 plural forms. If \fBnplurals\fR is defined, the same line should contain
311 \fBplural=\fR\fIexpression\fR, separated by a semicolon character. The
312 \fIexpression\fR is a C language expression to determine which version of
313 \fBmsgstr\fR[\fIn\fR] is to be used based on the value of \fIn\fR, the last
314 argument of \fBngettext()\fR, \fBdngettext()\fR, or \fBdcngettext()\fR. For
315 example,
316 .sp
317 .in +2
318 .nf
```



```
319 nplurals=2; plural= n == 1 ? 0 : 1
320 .fi
321 .in -2
322 .sp

324 .sp
325 .LP
326 indicates that there are two plural forms in the language. msgstr[0] is used if
327 n == 1, otherwise msgstr[1] is used. For another example:
328 .sp
329 .in +2
330 .nf
331 nplurals=3; plural= n == 1 ? 0 : n == 2 ? 1 : 2
332 .fi
333 .in -2
334 .sp

336 .sp
337 .LP
338 indicates that there are three plural forms in the language. msgstr[0] is used
339 if n == 1, msgstr[1] is used if n == 2, otherwise msgstr[2] is used.
340 .sp
341 .LP
342 If the header entry contains a \fBcharset\fR=\fIcodeset\fR string, the
343 \fIcodeset\fR is used to indicate the codeset to be used to encode the message
344 strings. If the output string’s codeset is different from the message string’s
345 codeset, codeset conversion from the message string’s codeset to the output
346 string’s codeset will be performed upon the call of \fBgettext()\fR,
347 \fBdgettext()\fR, \fBdcgettext()\fR, \fBngettext()\fR, \fBdngettext()\fR, and
348 \fBdcngettext()\fR for the GNU-compatible message catalogs. The output string’s
349 codeset is determined by the current locale’s codeset (the return value of
350 \fBnl_langinfo(CODESET\fR)) by default, and can be changed by the call of
351 \fBbind_textdomain_codeset()\fR.
352 .SS "Message catalog file format"
356 .sp
353 .LP
354 The \fBmsgfmt\fR utility can generate the message object both in Solaris
355 message catalog file format and in GNU-compatible message catalog file format.
356 If the \fB-s\fR option is specified and the input file is a Solaris \fB\&.po\fR
357 file, the \fBmsgfmt\fR utility generates the message object in Solaris message
358 catalog file format. If the \fB-g\fR option is specified and the input file is
359 a GNU \fB\&.po\fR file, the \fBmsgfmt\fR utility generates the message object
360 in GNU-compatible message catalog file format. If neither the \fB-s\fR nor
361 \fB-g\fR option is specified, the \fBmsgfmt\fR utility determines the message
362 catalog file format as follows:
363 .RS +4
364 .TP
365 .ie t \(bu
366 .el o
367 If the \fB\&.po\fR file contains a valid GNU header entry (having an empty
368 string for \fBmsgid\fR), the \fBmsgfmt\fR utility uses the GNU-compatible
369 message catalog file format.
370 .RE
371 .RS +4
372 .TP
373 .ie t \(bu
374 .el o
375 Otherwise, the \fBmsgfmt\fR utility uses the Solaris message catalog file
376 format.
377 .RE
378 .sp
379 .LP
380 If the \fBmsgfmt\fR utility determined that the Solaris message catalog file
381 format is used, as above, but found the \fB\&.po\fR file contains directives
382 that are specific to the GNU-compatible message catalog file format, such as
383 \fBmsgid_plural\fR and \fBmsgstr\fR[\fIn\fR], the \fBmsgfmt\fR utility handles
```

```
384 those directives as invalid specifications.
385 .SH EXAMPLES
386 .LP
387 \fBExample 1 \fRCreating message objects from message files
388 .sp
389 .LP
390 In this example, \fBmodule1.po\fR and \fBmodule2.po\fR are portable message
391 objects files.

393 .sp
394 .in +2
395 .nf
396 example% \fBcat module1.po\fR
397 # default domain "messages.mo"
398 msgid  "msg 1"
399 msgstr "msg 1 translation"
400 #
401 domain "help_domain"
402 msgid  "help 2"
403 msgstr "help 2 translation"
404 #
405 domain "error_domain"
406 msgid  "error 3"
407 msgstr "error 3 translation"
408 example% \fBcat module2.po\fR
409 # default domain "messages.mo"
410 msgid  "mesg 4"
411 msgstr "mesg 4 translation"
412 #
413 domain "error_domain"
414 msgid  "error 5"
415 msgstr "error 5 translation"
416 #
417 domain "window_domain"
418 msgid  "window 6"
419 msgstr "window 6 translation"
420 .fi
421 .in -2
422 .sp

424 .sp
425 .LP
426 The following command will produce the output files \fBmessages.mo\fR,
427 \fBhelp_domain.mo\fR, and \fBerror_domain.mo\fR in Solaris message catalog file
428 format:

430 .sp
431 .in +2
432 .nf
433 example% \fBmsgfmt module1.po\fR
434 .fi
435 .in -2
436 .sp

438 .sp
439 .LP
440 The following command will produce the output files \fBmessages.mo\fR,
441 \fBhelp_domain.mo\fR, \fBerror_domain.mo\fR, and \fBwindow_domain.mo\fR in
442 Solaris message catalog file format:

444 .sp
445 .in +2
446 .nf
447 example% \fBmsgfmt module1.po module2.po\fR
448 .fi
449 .in -2
```



```
450 .sp

452 .sp
453 .LP
454 The following command will produce the output file \fBhello.mo\fR in Solaris
455 message catalog file format:

457 .sp
458 .in +2
459 .nf
460 example% \fBmsgfmt -o hello.mo module1.po module2.po\fR
461 .fi
462 .in -2
463 .sp

465 .SH ENVIRONMENT VARIABLES
470 .sp
466 .LP
467 See \fBenviron\fR(5) for descriptions of the following environmental variables
468 that affect the execution of \fBmsgfmt\fR: \fBLC_CTYPE\fR, \fBLC_MESSAGES\fR,
469 and \fBNLSPATH\fR.
470 .SH ATTRIBUTES
476 .sp
471 .LP
472 See \fBattributes\fR(5) for descriptions of the following attributes:
473 .sp

475 .sp
476 .TS
477 box;
478 c | c
479 l | l .
480 ATTRIBUTE TYPE  ATTRIBUTE VALUE
481 _
482 CSI     Enabled
483 .TE

485 .SH SEE ALSO
492 .sp
486 .LP
487 \fBxgettext\fR(1), \fBgettext\fR(3C), \fBsetlocale\fR(3C), \fBattributes\fR(5),
488 \fBenviron\fR(5)
489 .SH NOTES
497 .sp
490 .LP
491 Installing message catalogs under the C locale is pointless, since they are
492 ignored for the sake of efficiency.
```

```
**********************************************************
    7389 Fri Jan 11 21:14:00 2019
new/usr/src/man/man1/mt.1
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
'\" te
.\"  Copyright (c) 2007, Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH MT 1 "Jun 21, 2007"
.SH NAME
mt \- magnetic tape control
.SH SYNOPSIS
.LP
.nf
\fBmt\fR [\fB-f\fR \fItapename\fR] \fIcommand\fR... [\fIcount\fR]
.fi

.SH DESCRIPTION
.sp
.LP
The \fBmt\fR utility sends commands to a magnetic tape drive. If \fB-f\fR
\fItapename\fR is not specified, the environment variable \fBTAPE\fR is used.
If \fBTAPE\fR does not exist, \fBmt\fR uses the device \fB/dev/rmt/0n\fR.
.SH OPTIONS
.sp
.LP
The following options are supported:
.sp
.ne 2
.na
\fB\fB-f\fR \fItapename\fR\fR
.ad
.RS 15n
Specifies the raw tape device.
.RE

.SH OPERANDS
.sp
.LP
The following operands are supported:
.sp
.ne 2
.na
\fB\fIcount\fR\fR
.ad
.RS 11n
The number of times that the requested operation is to be performed. By
default, \fBmt\fR performs \fIcommand\fR once. Multiple operations of
\fIcommand\fR can be performed by specifying \fIcount\fR.
.RE

.sp
.ne 2
.na
\fB\fIcommand\fR\fR
.ad
.RS 11n
The following available commands that can be sent to a magnetic tape drive are
supported. Only as many characters as are required to uniquely identify a
\fIcommand\fR need be specified.
.sp
```



```
.ne 2
.na
\fB\fBasf\fR\fR
.ad
.RS 10n
Specifies absolute space to \fIcount\fR file number. This is equivalent to a
\fBrewind\fR followed by a \fBfsf\fR \fIcount\fR.
.RE

.sp
.ne 2
.na
\fB\fBbsf\fR\fR
.ad
.RS 10n
Back spaces over \fIcount\fR EOF marks. The tape is positioned on the
beginning-of-tape side of the EOF mark.
.RE

.sp
.ne 2
.na
\fB\fBbsr\fR\fR
.ad
.RS 10n
Back spaces \fIcount\fR records.
.RE

.sp
.ne 2
.na
\fB\fBbssf\fR\fR
.ad
.RS 10n
Back spaces over the requested number of sequential file marks. Sequential file
marks are where the file marks are one right after the other with no other
blocks of any kind between the file marks. The number argument specifies how
many sequential file marks to which to space. For example, \fBbssf 4\fR
searches backwards to the first place where there are 4 sequential file marks
and positions to the BOP side of the 4th file mark.
.sp
This command is not supported by all drives.
.RE

.sp
.ne 2
.na
\fB\fBeof\fR\fR
.ad
.br
.na
\fB\fBweof\fR\fR
.ad
.RS 10n
Writes \fIcount\fR EOF marks at the current position on the tape.
.RE

.sp
.ne 2
.na
\fB\fBfsf\fR\fR
.ad
.RS 10n
Forward spaces over \fIcount\fR EOF marks. The tape is positioned on the first
block of the file.
.RE
```

```
123 .sp
124 .ne 2
125 .na
126 \fB\fBfsr\fR\fR
127 .ad
128 .RS 10n
129 Forward spaces \fIcount\fR records.
130 .RE

132 .sp
133 .ne 2
134 .na
135 \fB\fBfssf\fR\fR
136 .ad
137 .RS 10n
138 Forward spaces the over requested number of sequential file marks. Sequential
139 file marks are where the file marks are one right after the other with no other
140 blocks of any kind between the file marks. The number argument specifies how
141 many sequential file marks to which to space. For example, \fBfssf 4\fR
142 searches forwards to the first place where there are 4 sequential file marks
143 and positions after the 4th file mark.
144 .sp
145 This command is not supported by all drives.
146 .RE

148 .sp
149 .ne 2
150 .na
151 \fB\fBload\fR\fR
152 .ad
153 .RS 10n
154 Requests drive load and thread current media. Not supported by all drives.
155 .RE

157 .sp
158 .ne 2
159 .na
160 \fB\fBlock\fR\fR
161 .ad
162 .RS 10n
163 Prevents media removal.
164 .RE

166 .sp
167 .ne 2
168 .na
169 \fB\fBnbsf\fR\fR
170 .ad
171 .RS 10n
172 Back spaces \fIcount\fR files. The tape is positioned on the first block of the
173 file. This is equivalent to \fIcount+1\fR \fBbsf\fRs followed by one \fBfsf\fR.
176 file. This is equivalent to \fIcount+1\fR \fBbsf\fRs ollowed by one \fBfsf\fR.
174 .RE

176 .sp
177 .ne 2
178 .na
179 \fB\fBseek\fR\fR
180 .ad
181 .RS 10n
182 Positions to requested logical tape position.
183 .RE

185 .sp
186 .ne 2
```



```
187 .na
188 \fB\fBtell\fR\fR
189 .ad
190 .RS 10n
191 Gets and prints current logical tape position.
192 .RE

194 .sp
195 .ne 2
196 .na
197 \fB\fBunlock\fR\fR
198 .ad
199 .RS 10n
200 Allows media removal.
201 .RE

203 If \fIcount\fR is specified with any of the following commands, the \fIcount\fR
204 is ignored and the command is performed only once.
205 .sp
206 .ne 2
207 .na
208 \fB\fBconfig\fR\fR
209 .ad
210 .RS 16n
211 Reads the drives current configuration from the driver and displays it in
212 \fBst.conf\fR format. See \fBst\fR(7D) for definition of fields and there
213 meanings.
214 .RE

216 .sp
217 .ne 2
218 .na
219 \fB\fBeom\fR\fR
220 .ad
221 .RS 16n
222 Spaces to the end of recorded media on the tape. This is useful for appending
223 files onto previously written tapes.
224 .RE

226 .sp
227 .ne 2
228 .na
229 \fB\fBerase\fR\fR
230 .ad
231 .RS 16n
232 Erases the entire tape.
233 .sp
234 Some tape drives have option settings where only portions of the tape can be
235 erased. Be sure to select the correct setting to erase the whole tape. Erasing
236 a tape can take a long time depending on the device and/or tape. Refer to the
237 device specific manual for time details.
238 .RE

240 .sp
241 .ne 2
242 .na
243 \fB\fBforcereserve\fR\fR
244 .ad
245 .RS 16n
246 Attempts to break a SCSI II reserve issued by another initiator. When this
247 command completes, the drive is not reserved for the current initiator, but is
248 available for use. This command can be only be executed by those with
249 super-user privileges.
250 .RE

252 .sp
```

```
.ne 2
.na
\fB\fBoffline\fR\fR
.ad
.br
.na
\fB\fBrewoffl\fR\fR
.ad
.RS 16n
Rewinds the tape and, if appropriate, takes the drive unit off-line by
unloading the tape.
.RE

.sp
.ne 2
.na
\fB\fBrelease\fR\fR
.ad
.RS 16n
Re-establishes the default behavior of releasing at close.
.RE

.sp
.ne 2
.na
\fB\fBreserve\fR\fR
.ad
.RS 16n
Allows the tape drive to remain reserved after closing the device. The drive
must then be explicitly released.
.RE

.sp
.ne 2
.na
\fB\fBretension\fR\fR
.ad
.RS 16n
Rewinds the cartridge tape completely, then winds it forward to the end of the
reel and back to beginning-of-tape to smooth out tape tension.
.RE

.sp
.ne 2
.na
\fB\fBrewind\fR\fR
.ad
.RS 16n
Rewinds the tape.
.RE

.sp
.ne 2
.na
\fB\fBstatus\fR\fR
.ad
.RS 16n
Prints status information about the tape unit.
.sp
Status information can include the sense key reported by the drive, the
residual and retries for the last operation, the current tape position reported
in file number, and the number of blocks from the beginning of that file. It
might also report that WORM media is loaded in that drive.
.RE

.RE
```



```
.SH EXIT STATUS
.sp
.ne 2
.na
\fB\fB0\fR\fR
.ad
.RS 5n
All operations were successful.
.RE

.sp
.ne 2
.na
\fB\fB1\fR\fR
.ad
.RS 5n
Command was unrecognized or \fBmt\fR was unable to open the specified tape
drive.
.RE

.sp
.ne 2
.na
\fB\fB2\fR\fR
.ad
.RS 5n
An operation failed.
.RE

.SH FILES
.sp
.ne 2
.na
\fB\fB/dev/rmt/*\fR\fR
.ad
.RS 14n
magnetic tape interface
.RE

.SH SEE ALSO
.sp
.LP
\fBtar\fR(1), \fBtcopy\fR(1), \fBar.h\fR(3HEAD), \fBattributes\fR(5),
\fBmtio\fR(7I), \fBst\fR(7D)
.SH BUGS
.sp
.LP
Not all devices support all options. Some options are hardware-dependent. Refer
to the corresponding device manual page.
.sp
.LP
\fBmt\fR is architecture sensitive. Heterogeneous operation (that is, SPARC to
x86 or the reverse) is not supported.
```

```
**********************************************************
   20462 Fri Jan 11 21:14:00 2019
new/usr/src/man/man1/truss.1
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
'\" te
.\" Copyright (c) 2007, Sun Microsystems, Inc. All Rights Reserved.
.\" Copyright 1989 AT&T
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH TRUSS 1 "Jul 31, 2004"
.SH NAME
truss \- trace system calls and signals
.SH SYNOPSIS
.LP
.nf
\fBtruss\fR [\fB-fcaeildDE\fR] [\fB-\fR [tTvx] [!] \fIsyscall\fR ,...]
     [\fB-\fR [sS] [!] \fIsignal\fR ,...] [\fB-\fR [mM] [!] \fIfault\fR ,...]
     [\fB-\fR [rw] [!] \fIfd\fR ,...]
     [\fB-\fR [uU] [!] \fIlib\fR ,... : [:] [!] \fIfunc\fR ,...]
     [\fB-o\fR \fIoutfile\fR] \fIcommand\fR | \fB-p\fR \fIpid\fR[\fI/lwps\fR]...
.fi

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

.sp
```



```
.ne 2
.na
\fB\fB-c\fR\fR
.ad
.sp .6
.RS 4n
Counts traced system calls, faults, and signals rather than displaying the
trace line-by-line. A summary report is produced after the traced command
terminates or when \fBtruss\fR is interrupted. If \fB-f\fR is also specified,
the counts include all traced system calls, faults, and signals for child
processes.
.RE

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

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

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

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

```
125 In contrast to  the \fB-D\fR option, this is the amount of time spent within
126 the system call.
127 .RE

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

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

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

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

184 .sp
185 .ne 2
186 .na
187 \fB\fB-M\fR [\fB!\fR]\fIfault\fR,...\fR
188 .ad
189 .sp .6
190 .RS 4n
```



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

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

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

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

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

(see \fB<sys/signal.h>\fR). If the list begins with a \fB!\fR, the specified signals are excluded from the trace output. Default is \fB-sall\fR.
.RE

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

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

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

.sp
.ne 2
.na
\fB\fB-u\fR
[\fB!\fR]\fIlib\fR,...\fB:\fR[\fB:\fR][\fB!\fR]\fIfunc\fR,\|.\|.\|.\fR
.ad
.sp .6
.RS 4n
User-level function call tracing. \fIlib\fR,\|.\|.\|. is a comma-separated list of dynamic library names, excluding the ‘‘\fB\&.so.\fR\fIn\fR’’ suffix. \fIfunc\fR,\|.\|.\|. is a comma-separated list of function names. In both cases the names can include name-matching metacharacters \fB*\fR,\fB?\fR,\fB[]\fR with the same meanings as those of \fBsh\fR(1) but as applied to the

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

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

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

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

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

```
.ad
.sp .6
.RS 4n
Shows the contents of the I/O buffer for each \fBwrite()\fR on any of the
specified file descriptors (see the \fB-r\fR option). Default is
\fB\fR\fB-w\fR\fB!all\fR.
.RE

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

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

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

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

.sp
.in +2
```



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

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

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

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

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

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

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

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

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

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

```
.in -2
.sp

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

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

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

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

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

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

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

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

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



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

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

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

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

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

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

```
**********************************************************
    3683 Fri Jan 11 21:14:00 2019
new/usr/src/man/man1b/file.1b
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
```

```
'\" te
.\" Copyright (c) 1992, Sun Microsystems, 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 FILE 1B "Sep 14, 1992"
.SH NAME
file \- determine the type of a file by examining its contents
.SH SYNOPSIS
.LP
.nf
\fB/usr/ucb/file\fR [\fB-f\fR \fIffile\fR] [\fB-cL\fR] [\fB-m\fR \fImfile\fR] \f
.fi

.SH DESCRIPTION
.sp
.LP
\fBfile\fR performs a series of tests on each  \fIfilename\fR in an attempt to
determine what it contains. If the contents of a file appear to be  \fBASCII\fR
text, \fBfile\fR examines the first 512 bytes and tries to guess its language.
.sp
.LP
\fBfile\fR uses the file \fB/etc/magic\fR to identify files that have some sort
of \fImagic number\fR, that is, any file containing a numeric or string
constant that indicates its type.
.SH OPTIONS
.sp
.ne 2
.na
\fB\fB-c\fR\fR
.ad
.RS 12n
Check for format errors in the magic number file. For reasons of efficiency,
this validation is not normally carried out. No file type-checking is done
under \fB-c\fR.
.RE

.sp
.ne 2
.na
\fB\fB-f\fR \fIffile\fR\fR
.ad
.RS 12n
Get a list of filenames to identify from  \fIffile.\fR
.RE

.sp
.ne 2
.na
\fB\fB-L\fR\fR
.ad
.RS 12n
If a file is a symbolic link, test the file the link references rather than the
link itself.
.RE

.sp
.ne 2
```



```
.na
\fB\fB-m\fR\fI mfile\fR\fR
.ad
.RS 12n
Use \fImfile\fR as the name of an alternate magic number file.
.RE

.SH EXAMPLES
.LP
\fBExample 1 \fRUsing \fBfile\fR on all the files in a specific user's
directory.
.sp
.LP
This example illustrates the use of \fBfile\fR on all the files in a specific
user's directory:

.sp
.in +2
.nf
example% \fBpwd
/usr/blort/misc\fR
.fi
.in -2
.sp

.sp
.in +2
.nf
example% \fB/usr/ucb/file  *\fR

code:           mc68020 demand paged executable
code.c:         c program text
counts:         ascii text
doc:            roff,nroff, or eqn input text
empty.file:     empty
libz:           archive random library
memos:          directory
project:        symbolic link to /usr/project
project:        symboliclink to /usr/project
script:         executable shell script
titles:         ascii text
s5.stuff:       cpio archive


example%
.fi
.in -2
.sp

.SH ENVIRONMENT VARIABLES
.sp
.LP
The environment variables \fBLC_CTYPE\fR, \fBLANG\fR, and \fBLC_default\fR
control the character classification throughout \fBfile\fR. On entry to
\fBfile\fR, these environment variables are checked in the following order:
\fBLC_CTYPE\fR, \fBLANG\fR, and \fBLC_default\fR. When a valid value is found,
remaining environment variables for character classification are ignored. For
example, a new setting for \fBLANG\fR does not override the current valid
character classification rules of \fBLC_CTYPE\fR. When none of the values is
valid, the shell character classification defaults to the   POSIX.1 "C"
locale.
.SH FILES
.sp
.LP
\fB/etc/magic\fR
.SH SEE ALSO
```

```
124 .sp
120 .LP
121 \fBmagic\fR(4), \fBattributes\fR(5)
122 .SH BUGS
128 .sp
123 .LP
124 \fBfile\fR often makes mistakes.  In particular, it often suggests that command
125 files are C programs.
126 .sp
127 .LP
128 \fBfile\fR does not recognize Pascal or  LISP.
```

```
**********************************************************
   27902 Fri Jan 11 21:14:01 2019
new/usr/src/man/man1has/vi.1has
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
.\"
.\" 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) 2007, Sun Microsystems, Inc.  All Rights Reserved.
.\"
.TH VI 1HAS "May 16, 2007"
.SH NAME
vi, view, vedit \- screen-oriented (visual) display editor based on ex
.SH SYNOPSIS
.LP
.nf
\fB/usr/bin/vi\fR [\fB-|\fR \fB-s\fR] [\fB-l\fR] [\fB-L\fR] [\fB-R\fR] [\fB-r\fR
     [\fB-t\fR \fItag\fR] [\fB-v\fR] [\fB-V\fR] [\fB-x\fR] [\fB-w\fR\fIn\fR] [\f
     [\fB+\fR\fIcommand\fR | \fB-c\fR \fIcommand\fR] \fIfilename\fR...
.fi

.LP
```



```
.nf
\fB/usr/bin/view\fR [\fB-|\fR \fB-s\fR] [\fB-l\fR] [\fB-L\fR] [\fB-R\fR] [\fB-r\
     [\fB-t\fR \fItag\fR] [\fB-v\fR] [\fB-V\fR] [\fB-x\fR] [\fB-w\fR\fIn\fR] [\f
     [\fB+\fR\fIcommand\fR | \fB-c\fR \fIcommand\fR] \fIfilename\fR...
.fi

.LP
.nf
\fB/usr/bin/vedit\fR [\fB-|\fR \fB-s\fR] [\fB-l\fR] [\fB-L\fR] [\fB-R\fR] [\fB-r
     [\fB-t\fR \fItag\fR] [\fB-v\fR] [\fB-V\fR] [\fB-x\fR] [\fB-w\fR\fIn\fR] [\f
     [\fB+\fR\fIcommand\fR | \fB-c\fR \fIcommand\fR] \fIfilename\fR...
.fi

.LP
.nf
\fB/usr/xpg4/bin/vi\fR [\fB-|\fR \fB-s\fR] [\fB-l\fR] [\fB-L\fR] [\fB-R\fR] [\fB
     [\fB-S\fR] [\fB-t\fR \fItag\fR] [\fB-v\fR] [\fB-V\fR] [\fB-x\fR] [\fB-w\fR\
     [\fB+\fR\fIcommand\fR | \fB-c\fR \fIcommand\fR] \fIfilename\fR...
.fi

.LP
.nf
\fB/usr/xpg4/bin/view\fR [\fB-|\fR \fB-s\fR] [\fB-l\fR] [\fB-L\fR] [\fB-R\fR] [\
     [\fB-S\fR] [\fB-t\fR \fItag\fR] [\fB-v\fR] [\fB-V\fR] [\fB-x\fR] [\fB-w\fR\
     [\fB+\fR\fIcommand\fR | \fB-c\fR \fIcommand\fR] \fIfilename\fR...
.fi

.LP
.nf
\fB/usr/xpg4/bin/vedit\fR [\fB-|\fR \fB-s\fR] [\fB-l\fR] [\fB-L\fR] [\fB-R\fR] [
     [\fB-S\fR] [\fB-t\fR \fItag\fR] [\fB-v\fR] [\fB-V\fR] [\fB-x\fR] [\fB-w\fR\
     [\fB+\fR\fIcommand\fR | \fB-c\fR \fIcommand\fR] \fIfilename\fR...
.fi

.LP
.nf
\fB/usr/xpg6/bin/vi\fR [\fB-|\fR \fB-s\fR] [\fB-l\fR] [\fB-L\fR] [\fB-R\fR] [\fB
     [\fB-S\fR] [\fB-t\fR \fItag\fR] [\fB-v\fR] [\fB-V\fR] [\fB-x\fR] [\fB-w\fR\
     [\fB+\fR\fIcommand\fR | \fB-c\fR \fIcommand\fR] \fIfilename\fR...
.fi

.LP
.nf
\fB/usr/xpg6/bin/view\fR [\fB-|\fR \fB-s\fR] [\fB-l\fR] [\fB-L\fR] [\fB-R\fR] [\
     [\fB-S\fR] [\fB-t\fR \fItag\fR] [\fB-v\fR] [\fB-V\fR] [\fB-x\fR] [\fB-w\fR\
     [\fB+\fR\fIcommand\fR | \fB-c\fR \fIcommand\fR] \fIfilename\fR...
.fi

.LP
.nf
\fB/usr/xpg6/bin/vedit\fR [\fB-|\fR \fB-s\fR] [\fB-l\fR] [\fB-L\fR] [\fB-R\fR] [
     [\fB-S\fR] [\fB-t\fR \fItag\fR] [\fB-v\fR] [\fB-V\fR] [\fB-x\fR] [\fB-w\fR\
     [\fB+\fR\fIcommand\fR | \fB-c\fR \fIcommand\fR] \fIfilename\fR...
.fi

.SH DESCRIPTION
.sp
.LP
The \fBvi\fR (visual) utility is a display-oriented text editor based on an
underlying line editor \fBex\fR. It is possible to use the command mode of
\fBex\fR from within \fBvi\fR and to use the command mode of \fBvi\fR from
within \fBex\fR. The visual commands are described on this manual page; how to
set options (like automatically numbering lines and automatically starting a
new output line when you type carriage return) and all \fBex\fR line editor
commands are described on the \fBex\fR(1) manual page.
.sp
```

```
.LP
When using \fBvi\fR, changes you make to the file are reflected in what you see
on your terminal screen. The position of the cursor on the screen indicates the
position within the file.
.sp
.LP
The \fBview\fR invocation is the same as \fBvi\fR except that the
\fBreadonly\fR flag is set.
.sp
.LP
The \fBvedit\fR invocation is intended for beginners. It is the same as
\fBvi\fR except that the \fBreport\fR flag is set to \fB1\fR, the
\fBshowmode\fR and \fBnovice\fR flags are set, and \fBmagic\fR is turned off.
These defaults make it easier to learn how to use \fBvi\fR.
.SH OPTIONS
.sp
.LP
The following options are supported:
The following options are supporrted:
.SS "Invocation Options"
.sp
.LP
The following invocation options are interpreted by \fBvi\fR (previously
documented options are discussed under NOTES):
.sp
.ne 2
.na
\fB\fB\(mi\fR | \fB-s\fR\fR
.ad
.RS 25n
Suppresses all interactive user feedback. This is useful when processing editor
scripts.
.RE

.sp
.ne 2
.na
\fB\fB-C\fR\fR
.ad
.RS 25n
Encryption option. Same as the \fB-x\fR option, except that \fBvi\fR simulates
the \fBC\fR command of \fBex\fR. The \fBC\fR command is like the \fBX\fR
command of \fBex\fR, except that all text read in is assumed to have been
encrypted.
.RE

.sp
.ne 2
.na
\fB\fB-l\fR\fR
.ad
.RS 25n
Sets up for editing \fBLISP\fR programs.
.RE

.sp
.ne 2
.na
\fB\fB-L\fR\fR
.ad
.RS 25n
Lists the name of all files saved as the result of an editor or system crash.
.RE

.sp
.ne 2
```



```
.na
\fB\fB-r\fR \fIfilename\fR\fR
.ad
.RS 25n
Edits \fIfilename\fR after an editor or system crash. (Recovers the version of
\fIfilename\fR that was in the buffer when the crash occurred.)
.RE

.sp
.ne 2
.na
\fB\fB-R\fR\fR
.ad
.RS 25n
\fBReadonly\fR mode. The \fBreadonly\fR flag is set, preventing accidental
overwriting of the file.
.RE

.sp
.ne 2
.na
\fB\fB-S\fR\fR
.ad
.RS 25n
This option is used in conjunction with the \fB-t\fR \fItag\fR option to tell
\fBvi\fR that the tags file can not be sorted and that, if the binary search
(which relies on a sorted tags file) for \fItag\fR fails to find it, the much
slower linear search should also be done. Since the linear search is slow,
users of large tags files should ensure that the tags files are sorted rather
than use this flag. Creation of tags files normally produces sorted tags files.
See \fBctags\fR(1) for more information on tags files.
.RE

.sp
.ne 2
.na
\fB\fB-t\fR \fItag\fR\fR
.ad
.RS 25n
Edits the file containing \fItag\fR and position the editor at its definition.
It is an error to specify more than one \fB-t\fR option.
.RE

.sp
.ne 2
.na
\fB\fB-v\fR\fR
.ad
.RS 25n
Starts up in display editing state, using \fBvi\fR. You can achieve the same
effect by typing the \fBvi\fR command itself.
.RE

.sp
.ne 2
.na
\fB\fB-V\fR\fR
.ad
.RS 25n
Verbose. When \fBex\fR commands are read by means of standard input, the input
is echoed to standard error. This can be useful when processing \fBex\fR
commands within shell scripts.
.RE

.sp
.ne 2
```

```
.na
\fB\fB-w\fR\fIn\fR\fR
.ad
.RS 25n
Sets the default window size to \fIn\fR. This is useful when using the editor
over a slow speed line.
.RE

.sp
.ne 2
.na
\fB\fB-x\fR\fR
.ad
.RS 25n
Encryption option. When used, \fBvi\fR simulates the \fBX\fR command of
\fBex\fR and prompts the user for a key. This key is used to encrypt and
decrypt text using the algorithm of the \fBcrypt\fR command. The \fBX\fR
command makes an educated guess to determine whether text read in is encrypted
or not. The temporary buffer file is encrypted also, using a transformed
version of the key typed in for the \fB-x\fR option. If an empty encryption
key is entered (that is, if the return key is pressed right after the prompt),
the file is not encrypted. This is a good way to decrypt a file erroneously
encrypted with a mistyped encryption key, such as a backspace or undo key.
.RE

.sp
.ne 2
.na
\fB\fB-\fR\fIcommand\fR | \fB-c\fR \fIcommand\fR\fR
.ad
.RS 25n
Begins editing by executing the specified editor \fIcommand\fR (usually a
search or positioning command).
.RE

.SS "/usr/xpg4/bin/vi and /usr/xpg6/bin/vi"
.sp
.LP
If both the \fB-t\fR \fItag\fR and the \fB-c\fR \fIcommand\fR options are
given, the \fB-t\fR \fItag\fR option is processed first. That is, the file
containing \fItag\fR is selected by \fB-t\fR and then the command is executed.
.SH OPERANDS
.sp
.LP
The following operands are supported:
.sp
.ne 2
.na
\fB\fIfilename\fR\fR
.ad
.RS 12n
A file to be edited.
.RE

.SH COMMAND SUMMARY
.sp
.LP
The \fBvi\fR command modes are summarized in this section.
.SS "vi Modes"
.sp
.ne 2
.na
\fBCommand\fR
.ad
.RS 13n
```



```
Normal and initial mode. Other modes return to command mode upon completion.
\fIESC\fR (escape) is used to cancel a partial command.
.RE

.sp
.ne 2
.na
\fBInput\fR
.ad
.RS 13n
Entered by setting any of the following options:
.sp
.in +2
.nf
a A i I o O c C s S R
.fi
.in -2
.sp

Arbitrary text can then be entered. Input mode is normally terminated with the
\fIESC\fR character, or, abnormally, with an interrupt.
.RE

.sp
.ne 2
.na
\fBLast line\fR
.ad
.RS 13n
Reading input for \fB: / ?\fR or \fB!\fR. Terminate by typing a carriage
return. An interrupt cancels termination.
.RE

.SS "Sample Commands"
.sp
.LP
In the descriptions, \fICR\fR stands for carriage return and \fIESC\fR stands
for the escape key.
.sp
.ne 2
.na
\fB\(<-, \(->\fR
.ad
.br
.na
\fBdown-arrow\fR
.ad
.br
.na
\fBup-arrow\fR
.ad
.RS 14n
arrow keys move the cursor
.RE

.sp
.ne 2
.na
\fBh j k l\fR
.ad
.RS 14n
same as arrow keys
.RE

.sp
.ne 2
```

```
.na
\fBi\fItext\fR\fIESC\fR\fR
.ad
.RS 14n
insert \fItext\fR
.RE

.sp
.ne 2
.na
\fBcw\fInew\fR\fIESC\fR\fR
.ad
.RS 14n
change word to \fInew\fR
.RE

.sp
.ne 2
.na
\fBea\fIs\fR\fIESC\fR\fR
.ad
.RS 14n
pluralize word (end of word; append \fBs\fR; escape from input state)
.RE

.sp
.ne 2
.na
\fBx\fR
.ad
.RS 14n
delete a character
.RE

.sp
.ne 2
.na
\fBdw\fR
.ad
.RS 14n
delete a word
.RE

.sp
.ne 2
.na
\fBdd\fR
.ad
.RS 14n
delete a line
.RE

.sp
.ne 2
.na
\fB3dd\fR
.ad
.RS 14n
delete 3 lines
.RE

.sp
.ne 2
.na
\fBu\fR
.ad
```



```
.RS 14n
undo previous change
.RE

.sp
.ne 2
.na
\fBZZ\fR
.ad
.RS 14n
exit \fBvi\fR, saving changes
.RE

.sp
.ne 2
.na
\fB:q!\fICR\fR\fR
.ad
.RS 14n
quit, discarding changes
.RE

.sp
.ne 2
.na
\fB/\fItext\fR\fICR\fR\fR
.ad
.RS 14n
search for \fItext\fR
.RE

.sp
.ne 2
.na
\fB^U ^D\fR
.ad
.RS 14n
scroll up or down
.RE

.sp
.ne 2
.na
\fB:\fIcmd\fR\fICR\fR\fR
.ad
.RS 14n
any \fBex\fR or \fBed\fR command
.RE

.SS "Counts Before vi Commands"
.sp
.LP
Numbers can be typed as a prefix to some commands. They are interpreted in one
of these ways:
.sp
.ne 2
.na
\fBline/column number\fR
.ad
.RS 22n
z  G  |
.RE

.sp
.ne 2
.na
```

```
510 \fBscroll amount\fR
511 .ad
512 .RS 22n
513 ^D  ^U
514 .RE

516 .sp
517 .ne 2
518 .na
519 \fBrepeat effect\fR
520 .ad
521 .RS 22n
522 most of the rest
523 .RE

525 .SS "Interrupting, Canceling"
535 .sp
526 .ne 2
527 .na
528 \fB\fIESC\fR\fR
529 .ad
530 .RS 7n
531 end insert or incomplete command
532 .RE

534 .sp
535 .ne 2
536 .na
537 \fB\fIDEL\fR\fR
538 .ad
539 .RS 7n
540 (delete or rubout) interrupts
541 .RE

543 .SS "File Manipulation"
554 .sp
544 .ne 2
545 .na
546 \fBZZ\fR
547 .ad
548 .RS 15n
549 if file modified, write and exit; otherwise, exit
550 .RE

552 .sp
553 .ne 2
554 .na
555 \fB:w\fICR\fR\fR
556 .ad
557 .RS 15n
558 write back changes
559 .RE

561 .sp
562 .ne 2
563 .na
564 \fB:w!\fICR\fR\fR
565 .ad
566 .RS 15n
567 forced write, if permission originally not valid
568 .RE

570 .sp
571 .ne 2
572 .na
573 \fB:q\fICR\fR\fR
```



```
574 .ad
575 .RS 15n
576 quit
577 .RE

579 .sp
580 .ne 2
581 .na
582 \fB:q!\fICR\fR\fR
583 .ad
584 .RS 15n
585 quit, discard changes
586 .RE

588 .sp
589 .ne 2
590 .na
591 \fB:e \fIname\fR\fICR\fR\fR
592 .ad
593 .RS 15n
594 edit file \fIname\fR
595 .RE

597 .sp
598 .ne 2
599 .na
600 \fB:e!\fICR\fR\fR
601 .ad
602 .RS 15n
603 reedit, discard changes
604 .RE

606 .sp
607 .ne 2
608 .na
609 \fB:e + \fIname\fR\fICR\fR\fR
610 .ad
611 .RS 15n
612 edit, starting at end
613 .RE

615 .sp
616 .ne 2
617 .na
618 \fB:e +\fIn\fR\fICR\fR\fR
619 .ad
620 .RS 15n
621 edit, starting at line \fIn\fR
622 .RE

624 .sp
625 .ne 2
626 .na
627 \fB:e #\fICR\fR\fR
628 .ad
629 .RS 15n
630 edit alternate file
631 .RE

633 .sp
634 .ne 2
635 .na
636 \fB:e! #\fICR\fR\fR
637 .ad
638 .RS 15n
639 edit alternate file, discard changes
```

```
 640 .RE

 642 .sp
 643 .ne 2
 644 .na
 645 \fB:w \fIname\fR\fICR\fR\fR
 646 .ad
 647 .RS 15n
 648 write file \fIname\fR
 649 .RE

 651 .sp
 652 .ne 2
 653 .na
 654 \fB:w! \fIname\fR\fICR\fR\fR
 655 .ad
 656 .RS 15n
 657 overwrite file \fIname\fR
 658 .RE

 660 .sp
 661 .ne 2
 662 .na
 663 \fB:sh\fICR\fR\fR
 664 .ad
 665 .RS 15n
 666 run shell, then return
 667 .RE

 669 .sp
 670 .ne 2
 671 .na
 672 \fB:!\fIcmd\fR\fICR\fR\fR
 673 .ad
 674 .RS 15n
 675 run \fIcmd\fR, then return
 676 .RE

 678 .sp
 679 .ne 2
 680 .na
 681 \fB:n\fICR\fR\fR
 682 .ad
 683 .RS 15n
 684 edit next file in arglist
 685 .RE

 687 .sp
 688 .ne 2
 689 .na
 690 \fB:n \fIargs\fR\fICR\fR\fR
 691 .ad
 692 .RS 15n
 693 specify new arglist
 694 .RE

 696 .sp
 697 .ne 2
 698 .na
 699 \fB^G\fR
 700 .ad
 701 .RS 15n
 702 show current file and line
 703 .RE

 705 .sp
```



```
 706 .ne 2
 707 .na
 708 \fB:ta \fItag\fR\fICR\fR\fR
 709 .ad
 710 .RS 15n
 711 position cursor to \fItag\fR
 712 .RE

 714 .sp
 715 .LP
 716 In general, any \fBex\fR or \fBed\fR command (such as \fIsubstitute\fR or
 717 \fIglobal\fR) can be typed, preceded by a colon and followed by a carriage
 718 return.
 719 .SS "Positioning Within a File"
 731 .sp
 720 .ne 2
 721 .na
 722 \fBF\fR
 723 .ad
 724 .RS 14n
 725 forward screen
 726 .RE

 728 .sp
 729 .ne 2
 730 .na
 731 \fB^B\fR
 732 .ad
 733 .RS 14n
 734 backward screen
 735 .RE

 737 .sp
 738 .ne 2
 739 .na
 740 \fB^D\fR
 741 .ad
 742 .RS 14n
 743 scroll down half screen
 744 .RE

 746 .sp
 747 .ne 2
 748 .na
 749 \fB^U\fR
 750 .ad
 751 .RS 14n
 752 scroll up half screen
 753 .RE

 755 .sp
 756 .ne 2
 757 .na
 758 \fB\fIn\fRG\fR
 759 .ad
 760 .RS 14n
 761 go to the beginning of the specified line (end default), where \fIn\fR is a
 762 line number
 763 .RE

 765 .sp
 766 .ne 2
 767 .na
 768 \fB/\fIpat\fR\fR
 769 .ad
 770 .RS 14n
```

```
771 next line matching \fIpat\fR
772 .RE

774 .sp
775 .ne 2
776 .na
777 \fB?\fIpat\fR\fR
778 .ad
779 .RS 14n
780 previous line matching \fIpat\fR
781 .RE

783 .sp
784 .ne 2
785 .na
786 \fBn\fR
787 .ad
788 .RS 14n
789 repeat last \fB/\fR or \fB?\fR command
790 .RE

792 .sp
793 .ne 2
794 .na
795 \fBN\fR
796 .ad
797 .RS 14n
798 reverse last \fB/\fR or \fB?\fR command
799 .RE

801 .sp
802 .ne 2
803 .na
804 \fB/\fIpat\fR/+\fIn\fR\fR
805 .ad
806 .RS 14n
807 \fIn\fRth line after \fIpat\fR
808 .RE

810 .sp
811 .ne 2
812 .na
813 \fB?\fIpat\fR?\(mi\fIn\fR\fR
814 .ad
815 .RS 14n
816 \fIn\fRth line before \fIpat\fR
817 .RE

819 .sp
820 .ne 2
821 .na
822 \fB]]\fR
823 .ad
824 .RS 14n
825 next section/function
826 .RE

828 .sp
829 .ne 2
830 .na
831 \fB[[\fR
832 .ad
833 .RS 14n
834 previous section/function
835 .RE
```



```
837 .sp
838 .ne 2
839 .na
840 \fB(\fR
841 .ad
842 .RS 14n
843 beginning of sentence
844 .RE

846 .sp
847 .ne 2
848 .na
849 \fB)\fR
850 .ad
851 .RS 14n
852 end of sentence
853 .RE

855 .sp
856 .ne 2
857 .na
858 \fB{\fR
859 .ad
860 .RS 14n
861 beginning of paragraph
862 .RE

864 .sp
865 .ne 2
866 .na
867 \fB}\fR
868 .ad
869 .RS 14n
870 end of paragraph
871 .RE

873 .sp
874 .ne 2
875 .na
876 \fB%\fR
877 .ad
878 .RS 14n
879 find matching \fB( )\fR or \fB{ }\fR
880 .RE

882 .SS "Adjusting the Screen"
895 .sp
883 .ne 2
884 .na
885 \fB^L\fR
886 .ad
887 .RS 16n
888 clear and redraw window
889 .RE

891 .sp
892 .ne 2
893 .na
894 \fB^R\fR
895 .ad
896 .RS 16n
897 clear and redraw window if \fB^L\fR is \(-> key
898 .RE

900 .sp
901 .ne 2
```

```
 902 .na
 903 \fBz\fICR\fR\fR
 904 .ad
 905 .RS 16n
 906 redraw screen with current line at top of window
 907 .RE

 909 .sp
 910 .ne 2
 911 .na
 912 \fBz\(mi\fICR\fR\fR
 913 .ad
 914 .RS 16n
 915 redraw screen with current line at bottom of window
 916 .RE

 918 .sp
 919 .ne 2
 920 .na
 921 \fBz.\fICR\fR\fR
 922 .ad
 923 .RS 16n
 924 redraw screen with current line at center of window
 925 .RE

 927 .sp
 928 .ne 2
 929 .na
 930 \fB/\fIpat\fR/z\(mi\fICR\fR\fR
 931 .ad
 932 .RS 16n
 933 move \fIpat\fR line to bottom of window
 934 .RE

 936 .sp
 937 .ne 2
 938 .na
 939 \fBz\fIn\fR.\fICR\fR\fR
 940 .ad
 941 .RS 16n
 942 use \fIn\fR\(miline window
 943 .RE

 945 .sp
 946 .ne 2
 947 .na
 948 \fB^E\fR
 949 .ad
 950 .RS 16n
 951 scroll window down one line
 952 .RE

 954 .sp
 955 .ne 2
 956 .na
 957 \fB^Y\fR
 958 .ad
 959 .RS 16n
 960 scroll window up one line
 961 .RE

 963 .SS "Marking and Returning"
 977 .sp
 964 .ne 2
 965 .na
 966 \fB\(ga\(ga\fR
```



```
 967 .ad
 968 .RS 12n
 969 move cursor to previous context
 970 .RE

 972 .sp
 973 .ne 2
 974 .na
 975 \fBa\’a\’\fR
 976 .ad
 977 .RS 12n
 978 move cursor to first non-white space in line
 979 .RE

 981 .sp
 982 .ne 2
 983 .na
 984 \fBm\fIx\fR\fR
 985 .ad
 986 .RS 12n
 987 mark current position with the \fBASCII\fR lower-case letter \fIx\fR
 988 .RE

 990 .sp
 991 .ne 2
 992 .na
 993 \fB\(ga\fIx\fR\fR
 994 .ad
 995 .RS 12n
 996 move cursor to mark \fIx\fR
 997 .RE

 999 .sp
1000 .ne 2
1001 .na
1002 \fBa\’\fIx\fR\fR
1003 .ad
1004 .RS 12n
1005 move cursor to first non-white space in line marked by \fIx\fR
1006 .RE

1008 .SS "Line Positioning"
1023 .sp
1009 .ne 2
1010 .na
1011 \fBH\fR
1012 .ad
1013 .RS 14n
1014 top line on screen
1015 .RE

1017 .sp
1018 .ne 2
1019 .na
1020 \fBL\fR
1021 .ad
1022 .RS 14n
1023 last line on screen
1024 .RE

1026 .sp
1027 .ne 2
1028 .na
1029 \fBM\fR
1030 .ad
1031 .RS 14n
```

```
1032 middle line on screen
1033 .RE

1035 .sp
1036 .ne 2
1037 .na
1038 \fB+\fR
1039 .ad
1040 .RS 14n
1041 next line, at first non-white space character
1042 .RE

1044 .sp
1045 .ne 2
1046 .na
1047 \fB\(mi\fR
1048 .ad
1049 .RS 14n
1050 previous line, at first non-white space character
1051 .RE

1053 .sp
1054 .ne 2
1055 .na
1056 \fB\fICR\fR\fR
1057 .ad
1058 .RS 14n
1059 return, same as \fB+\fR
1060 .RE

1062 .sp
1063 .ne 2
1064 .na
1065 \fB\fBdown-arrow\fR\fR
1066 .ad
1067 .br
1068 .na
1069 \fBor \fBj\fR\fR
1070 .ad
1071 .RS 14n
1072 next line, same column
1073 .RE

1075 .sp
1076 .ne 2
1077 .na
1078 \fB\fBup-arrow\fR\fR
1079 .ad
1080 .br
1081 .na
1082 \fBor \fBk\fR\fR
1083 .ad
1084 .RS 14n
1085 previous line, same column
1086 .RE

1088 .SS "Character Positioning"
1104 .sp
1089 .ne 2
1090 .na
1091 \fB^\fR
1092 .ad
1093 .RS 13n
1094 first non-white space character
1095 .RE
```



```
1097 .sp
1098 .ne 2
1099 .na
1100 \fB0\fR
1101 .ad
1102 .RS 13n
1103 beginning of line
1104 .RE

1106 .sp
1107 .ne 2
1108 .na
1109 \fB$\fR
1110 .ad
1111 .RS 13n
1112 end of line
1113 .RE

1115 .sp
1116 .ne 2
1117 .na
1118 \fB\fBl\fR or \fB\(->\fR\fR
1119 .ad
1120 .RS 13n
1121 forward
1122 .RE

1124 .sp
1125 .ne 2
1126 .na
1127 \fB\fBh\fR or \fB\(<-\fR\fR
1128 .ad
1129 .RS 13n
1130 backward
1131 .RE

1133 .sp
1134 .ne 2
1135 .na
1136 \fB^H\fR
1137 .ad
1138 .RS 13n
1139 same as \fB\(<-\fR (backspace)
1140 .RE

1142 .sp
1143 .ne 2
1144 .na
1145 \fBspace\fR
1146 .ad
1147 .RS 13n
1148 same as \fB\(->\fR (space bar)
1149 .RE

1151 .sp
1152 .ne 2
1153 .na
1154 \fBf\fIx\fR\fR
1155 .ad
1156 .RS 13n
1157 find next \fIx\fR
1158 .RE

1160 .sp
1161 .ne 2
1162 .na
```

```
1163 \fBF\fIx\fR\fR
1164 .ad
1165 .RS 13n
1166 find previous \fIx\fR
1167 .RE

1169 .sp
1170 .ne 2
1171 .na
1172 \fBt\fIx\fR\fR
1173 .ad
1174 .RS 13n
1175 move to character following the next \fIx\fR
1176 .RE

1178 .sp
1179 .ne 2
1180 .na
1181 \fBT\fIx\fR\fR
1182 .ad
1183 .RS 13n
1184 move to character following the previous \fIx\fR
1185 .RE

1187 .sp
1188 .ne 2
1189 .na
1190 \fB;\fR
1191 .ad
1192 .RS 13n
1193 repeat last \fBf\fR, \fBF\fR, \fBt\fR, or \fBT\fR
1194 .RE

1196 .sp
1197 .ne 2
1198 .na
1199 \fB,\fR
1200 .ad
1201 .RS 13n
1202 repeat inverse of last \fBf\fR, \fBF\fR, \fBt\fR, or \fBT\fR
1203 .RE

1205 .sp
1206 .ne 2
1207 .na
1208 \fB\fIn\fR|\fR
1209 .ad
1210 .RS 13n
1211 move to column \fIn\fR
1212 .RE

1214 .sp
1215 .ne 2
1216 .na
1217 \fB%\fR
1218 .ad
1219 .RS 13n
1220 find matching \fB( )\fR or \fB{ }\fR
1221 .RE

1223 .SS "Words, Sentences, Paragraphs"
1240 .sp
1224 .ne 2
1225 .na
1226 \fBw\fR
1227 .ad
```



```
1228 .RS 5n
1229 forward a word
1230 .RE

1232 .sp
1233 .ne 2
1234 .na
1235 \fBb\fR
1236 .ad
1237 .RS 5n
1238 back a word
1239 .RE

1241 .sp
1242 .ne 2
1243 .na
1244 \fBe\fR
1245 .ad
1246 .RS 5n
1247 end of word
1248 .RE

1250 .sp
1251 .ne 2
1252 .na
1253 \fB)\fR
1254 .ad
1255 .RS 5n
1256 to next sentence
1257 .RE

1259 .sp
1260 .ne 2
1261 .na
1262 \fB}\fR
1263 .ad
1264 .RS 5n
1265 to next paragraph
1266 .RE

1268 .sp
1269 .ne 2
1270 .na
1271 \fB(\fR
1272 .ad
1273 .RS 5n
1274 back a sentence
1275 .RE

1277 .sp
1278 .ne 2
1279 .na
1280 \fB{\fR
1281 .ad
1282 .RS 5n
1283 back a paragraph
1284 .RE

1286 .sp
1287 .ne 2
1288 .na
1289 \fBW\fR
1290 .ad
1291 .RS 5n
1292 forward a blank-delimited word
1293 .RE
```

```
.sp
.ne 2
.na
\fBB\fR
.ad
.RS 5n
back a blank-delimited word
.RE

.sp
.ne 2
.na
\fBE\fR
.ad
.RS 5n
end of a blank-delimited word
.RE

.SS "Corrections During Insert"
.sp
.ne 2
.na
\fB^H\fR
.ad
.RS 16n
erase last character (backspace)
.RE

.sp
.ne 2
.na
\fB^W\fR
.ad
.RS 16n
erase last word
.RE

.sp
.ne 2
.na
\fBerase\fR
.ad
.RS 16n
your erase character, same as \fB^H\fR (backspace)
.RE

.sp
.ne 2
.na
\fBkill\fR
.ad
.RS 16n
your kill character, erase this line of input
.RE

.sp
.ne 2
.na
\fB\e\fR
.ad
.RS 16n
quotes your erase and kill characters
.RE

.sp
```



```
.ne 2
.na
\fB\fIESC\fR\fR
.ad
.RS 16n
ends insertion, back to command mode
.RE

.sp
.ne 2
.na
\fBControl\(miC\fR
.ad
.RS 16n
interrupt, suspends insert mode
.RE

.sp
.ne 2
.na
\fB^D\fR
.ad
.RS 16n
backtab one character; reset left margin of \fIautoindent\fR
.RE

.sp
.ne 2
.na
\fB^^D\fR
.ad
.RS 16n
caret (\fB^\fR) followed by control-d (\fB^D\fR); backtab to beginning of line;
do not reset left margin of \fIautoindent\fR
.RE

.sp
.ne 2
.na
\fB0^D\fR
.ad
.RS 16n
backtab to beginning of line; reset left margin of \fIautoindent\fR
.RE

.sp
.ne 2
.na
\fB^V\fR
.ad
.RS 16n
quote non-printable character
.RE

.SS "Insert and Replace"
.sp
.ne 2
.na
\fBa\fR
.ad
.RS 12n
append after cursor
.RE

.sp
.ne 2
```

```
.na
\fBA\fR
.ad
.RS 12n
append at end of line
.RE

.sp
.ne 2
.na
\fBi\fR
.ad
.RS 12n
insert before cursor
.RE

.sp
.ne 2
.na
\fBI\fR
.ad
.RS 12n
insert before first non-blank
.RE

.sp
.ne 2
.na
\fBo\fR
.ad
.RS 12n
open line below
.RE

.sp
.ne 2
.na
\fBO\fR
.ad
.RS 12n
open line above
.RE

.sp
.ne 2
.na
\fBr\fIx\fR\fR
.ad
.RS 12n
replace single character with \fIx\fR
.RE

.sp
.ne 2
.na
\fBR\fItext\fR\fIESC\fR\fR
.ad
.RS 12n
replace characters
.RE

.SS "Operators"
.sp
.LP
Operators are followed by a cursor motion and affect all text that would have
been moved over. For example, since \fBw\fR moves over a word, \fBdw\fR deletes
```



```
the word that would be moved over. Double the operator, for example \fBdd\fR,
to affect whole lines.
.sp
.ne 2
.na
\fBd\fR
.ad
.RS 5n
delete
.RE

.sp
.ne 2
.na
\fBc\fR
.ad
.RS 5n
change
.RE

.sp
.ne 2
.na
\fBy\fR
.ad
.RS 5n
yank lines to buffer
.RE

.sp
.ne 2
.na
\fB<\fR
.ad
.RS 5n
left shift
.RE

.sp
.ne 2
.na
\fB>\fR
.ad
.RS 5n
right shift
.RE

.sp
.ne 2
.na
\fB!\fR
.ad
.RS 5n
filter through command
.RE

.SS "Miscellaneous Operations"
.sp
.ne 2
.na
\fBC\fR
.ad
.RS 5n
change rest of line (\fBc$\fR)
.RE
```

```
.sp
.ne 2
.na
\fBD\fR
.ad
.RS 5n
delete rest of line (\fBd$\fR)
.RE

.sp
.ne 2
.na
\fBs\fR
.ad
.RS 5n
substitute characters (\fBcl\fR)
.RE

.sp
.ne 2
.na
\fBS\fR
.ad
.RS 5n
substitute lines (\fBcc\fR)
.RE

.sp
.ne 2
.na
\fBJ\fR
.ad
.RS 5n
join lines
.RE

.sp
.ne 2
.na
\fBx\fR
.ad
.RS 5n
delete characters (\fBdl\fR)
.RE

.sp
.ne 2
.na
\fBX\fR
.ad
.RS 5n
delete characters before cursor \fBdh\fR)
.RE

.sp
.ne 2
.na
\fBY\fR
.ad
.RS 5n
yank lines (\fByy\fR)
.RE

.SS "Yank and Put"
.sp
.LP
```



```
Put inserts the text most recently deleted or yanked; however, if a buffer is
named (using the \fBASCII\fR lower-case letters \fBa\fR - \fBz\fR), the text in
that buffer is put instead.
.sp
.ne 2
.na
\fB3yy\fR
.ad
.RS 7n
yank 3 lines
.RE

.sp
.ne 2
.na
\fB3yl\fR
.ad
.RS 7n
yank 3 characters
.RE

.sp
.ne 2
.na
\fBp\fR
.ad
.RS 7n
put back text after cursor
.RE

.sp
.ne 2
.na
\fBP\fR
.ad
.RS 7n
put back text before cursor
.RE

.sp
.ne 2
.na
\fB\fI"x\fRp\fR
.ad
.RS 7n
put from buffer \fIx\fR
.RE

.sp
.ne 2
.na
\fB"\fIx\fRy\fR
.ad
.RS 7n
yank to buffer \fIx\fR
.RE

.sp
.ne 2
.na
\fB"\fIx\fRd\fR
.ad
.RS 7n
delete into buffer \fIx\fR
.RE
```

```
.SS "Undo, Redo, Retrieve"
.sp
.ne 2
.na
\fBu\fR
.ad
.RS 7n
undo last change
.RE

.sp
.ne 2
.na
\fBU\fR
.ad
.RS 7n
restore current line
.RE

.sp
.ne 2
.na
\fB\&.\fR
.ad
.RS 7n
repeat last change
.RE

.sp
.ne 2
.na
\fB"\fId\fRp\fR
.ad
.RS 7n
retrieve \fId\fR'th last delete
.RE

.SH USAGE
.sp
.LP
See \fBlargefile\fR(5) for the description of the behavior of \fBvi\fR and
\fBview\fR when encountering files greater than or equal to 2 Gbyte ( 2^31
bytes).
.SH ENVIRONMENT VARIABLES
.sp
.LP
See \fBenviron\fR(5) for descriptions of the following environment variables
that affect the execution of \fBvi\fR: \fBLANG\fR, \fBLC_ALL\fR,
\fBLC_COLLATE\fR, \fBLC_CTYPE\fR, \fBLC_TIME\fR, \fBLC_MESSAGES\fR,
\fBNLSPATH\fR, \fBPATH\fR, \fBSHELL\fR, and \fBTERM\fR.
.sp
.ne 2
.na
\fB\fBCOLUMNS\fR\fR
.ad
.RS 11n
Override the system-selected horizontal screen size.
.RE

.sp
.ne 2
.na
\fB\fBEXINIT\fR\fR
.ad
.RS 11n
Determine a list of \fBex\fR commands that are executed on editor start-up,
```



```
before reading the first file. The list can contain multiple commands by
separating them using a vertical-line (\fB|\fR) character.
.RE

.sp
.ne 2
.na
\fB\fBLINES\fR\fR
.ad
.RS 11n
Override the system-selected vertical screen size, used as the number of lines
in a screenful and the vertical screen size in visual mode.
.RE

.SH FILES
.sp
.ne 2
.na
\fB\fB/var/tmp\fR\fR
.ad
.sp .6
.RS 4n
default directory where temporary work files are placed; it can be changed
using the \fBdirectory\fR option (see the \fBex\fR(1) command)
.RE

.sp
.ne 2
.na
\fB\fB/usr/share/lib/terminfo/?/*\fR\fR
.ad
.sp .6
.RS 4n
compiled terminal description database
.RE

.sp
.ne 2
.na
\fB\fB/usr/lib/.COREterm/?/*\fR\fR
.ad
.sp .6
.RS 4n
subset of compiled terminal description database
.RE

.SH ATTRIBUTES
.sp
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.SS "/usr/bin/vi, /usr/bin/view, /usr/bin/vedit"
.sp

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
CSI     Not enabled
.TE

.SS "/usr/xpg4/bin/vi, /usr/xpg4/bin/view, /usr/xpg4/bin/vedit"
.sp
```

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

.SS "/usr/xpg6/bin/vi, /usr/xpg6/bin/view, /usr/xpg6/bin/vedit"
.sp

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

.SH SEE ALSO
.sp
.LP
\fBIntro\fR(1), \fBctags\fR(1), \fBed\fR(1), \fBedit\fR(1), \fBex\fR(1),
\fBattributes\fR(5), \fBenviron\fR(5), \fBlargefile\fR(5), \fBstandards\fR(5)
.sp
.LP
\fISolaris Advanced User\&’s Guide\fR
.SH AUTHOR
.sp
.LP
\fBvi\fR and \fBex\fR were developed by The University of California, Berkeley
California, Computer Science Division, Department of Electrical Engineering and
Computer Science.
.SH NOTES
.sp
.LP
Two options, although they continue to be supported, have been replaced in the
documentation by options that follow the Command Syntax Standard (see
\fBIntro\fR(1)). An \fB-r\fR option that is not followed with an
option-argument has been replaced by \fB-L\fR and \fB+\fR\fBcommand\fR has been
replaced by \fB-c\fR \fBcommand\fR.
.sp
.LP
The message \fBfile too large to recover with\fR \fB-r\fR \fBoption\fR, which
is seen when a file is loaded, indicates that the file can be edited and saved
successfully, but if the editing session is lost, recovery of the file with the
\fB-r\fR option is not possible.
.sp
.LP
The editing environment defaults to certain configuration options. When an
editing session is initiated, \fBvi\fR attempts to read the \fBEXINIT\fR
environment variable. If it exists, the editor uses the values defined in
\fBEXINIT\fR; otherwise the values set in \fB$HOME/.exrc\fR are used. If
\fB$HOME/.exrc\fR does not exist, the default values are used.
.sp
.LP
To use a copy of \fB\&.exrc\fR located in the current directory other than
\fB$HOME\fR, set the \fIexrc\fR option in \fBEXINIT\fR or \fB$HOME/.exrc\fR.
```



```
Options set in \fBEXINIT\fR can be turned off in a local \fB\&.exrc\fR only if
\fIexrc\fR is set in \fBEXINIT\fR or \fB$HOME/.exrc\fR. In order to be used,
\fI\&.exrc\fR in \fB$HOME\fR or the current directory must fulfill these
conditions:
.RS +4
.TP
.ie t \(bu
.el o
It must exist.
.RE
.RS +4
.TP
.ie t \(bu
.el o
It must be owned by the same userid as the real userid of the process, or the
process has appropriate privileges.
.RE
.RS +4
.TP
.ie t \(bu
.el o
It is not writable by anyone other than the owner.
.RE
.sp
.LP
Tampering with entries in \fB/usr/share/lib/terminfo/?/*\fR or
\fB/usr/share/lib/terminfo/?/*\fR (for example, changing or removing an entry)
can affect programs such as \fBvi\fR that expect the entry to be present and
correct. In particular, removing the "dumb" terminal can cause unexpected
problems.
.sp
.LP
Software tabs using \fB^T\fR work only immediately after the \fIautoindent\fR.
.sp
.LP
Left and right shifts on intelligent terminals do not make use of insert and
delete character operations in the terminal.
.sp
.LP
Loading an alternate \fBmalloc()\fR library using the environment variable
\fBLD_PRELOAD\fR can cause problems for \fB/usr/bin/vi\fR.
.sp
.LP
The \fBvi\fR utility currently has the following limitations:
.RS +4
.TP
1.
Lines, including the trailing NEWLINE character, can contain no more than
4096 bytes.
.sp
If a longer line is found, \fBLine too long\fR is displayed in the status line.
.RE
.RS +4
.TP
2.
The editor’s temporary work file can be no larger than 128Mb.
.sp
If a larger temporary file is needed, \fBTmp file too large\fR is displayed in
the status line.
.RE
```

```
**********************************************************
   35059 Fri Jan 11 21:14:01 2019
new/usr/src/man/man1m/boot.1m
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
```

```
'\" te
.\" Copyright 2018 OmniOS Community Edition (OmniOSce) Association.
.\" Copyright 2015 Nexenta Systems Inc.
.\" Copyright (c) 2008 Sun Microsystems, Inc. All Rights Reserved
.\" Copyright 1989 AT&T
.\" The contents of this file are subject to the terms of the Common Development
.\" See the License for the specific language governing permissions and limitati
.\" fields enclosed by brackets "[]" replaced with your own identifying informat
.TH BOOT 1M "Jul 20, 2018"
.SH NAME
boot \- start the system kernel or a standalone program
.SH SYNOPSIS
.SS "SPARC"
.LP
.nf
\fBboot\fR [\fIOBP\fR \fInames\fR] [\fIfile\fR] [\fB-aLV\fR] [\fB-F\fR \fIobject
     [\fB-Z\fR \fIdataset\fR] [\fIboot-flags\fR] [\fB\(mi\(mi\fR] [\fIclient-pro
.fi

.SS "x86"
.LP
.nf
\fBboot\fR [\fIboot-flags\fR] [\fB-B\fR \fIprop\fR=\fIval\fR [,\fIval\fR...]]
.fi

.SH DESCRIPTION
.LP
Bootstrapping is the process of loading and executing a standalone program. For
the purpose of this discussion, bootstrapping means the process of loading and
executing the bootable operating system. Typically, the standalone program is
the operating system kernel (see \fBkernel\fR(1M)), but any standalone program
can be booted instead. On a SPARC-based system, the diagnostic monitor for a
machine is a good example of a standalone program other than the operating
system that can be booted.
.sp
.LP
If the standalone is identified as a dynamically-linked executable, \fBboot\fR
will load the interpreter (linker/loader) as indicated by the executable format
and then transfer control to the interpreter. If the standalone is
statically-linked, it will jump directly to the standalone.
.sp
.LP
Once the kernel is loaded, it starts the UNIX system, mounts the necessary file
systems (see \fBvfstab\fR(4)), and runs \fB/sbin/init\fR to bring the system to
the "initdefault" state specified in \fB/etc/inittab\fR. See \fBinittab\fR(4).
.SS "SPARC Bootstrap Procedure"
.LP
On SPARC based systems, the bootstrap procedure on most machines consists of
the following basic phases.
.sp
.LP
After the machine is turned on, the system firmware (in PROM) executes power-on
self-test (POST). The form and scope of these tests depends on the version of
the firmware in your system.
.sp
.LP
After the tests have been completed successfully, the firmware attempts to
autoboot if the appropriate flag has been set in the non-volatile storage area
```



```
used by the firmware. The name of the file to load, and the device to load it
from can also be manipulated.
.sp
.LP
These flags and names can be set using the \fBeeprom\fR(1M) command from the
shell, or by using \fBPROM\fR commands from the \fBok\fR prompt after the
system has been halted.
.sp
.LP
The second level program is either a filesystem-specific boot block (when
booting from a disk), or \fBinetboot\fR (when booting across
the network).
.sp
.LP
Network Booting
.sp
.LP
Network booting occurs in two steps: the client first obtains an IP address and
any other parameters necessary to permit it to load the second-stage booter.
The second-stage booter in turn loads the boot archive from the boot device.
.sp
.LP
An IP address can be obtained in one of three ways: RARP, DHCP, or manual
configuration, depending on the functions available in and configuration of the
PROM. Machines of the \fBsun4u\fR and \fBsun4v\fR kernel architectures have
DHCP-capable PROMs.
.sp
.LP
The boot command syntax for specifying the two methods of network booting are:
.sp
.in +2
.nf
boot net:rarp
boot net:dhcp
.fi
.in -2
.sp

.sp
.LP
The command:
.sp
.in +2
.nf
boot net
.fi
.in -2
.sp

.sp
.LP
without a \fBrarp\fR or \fBdhcp\fR specifier, invokes the default method for
network booting over the network interface for which \fBnet\fR is an alias.
.sp
.LP
The sequence of events for network booting using RARP/\fBbootparams\fR is
described in the following paragraphs. The sequence for DHCP follows the
RARP/\fBbootparams\fR description.
.sp
.LP
When booting over the network using RARP/\fBbootparams\fR, the PROM begins by
broadcasting a reverse ARP request until it receives a reply. When a reply is
received, the PROM then broadcasts a TFTP request to fetch the first block of
\fBinetboot\fR. Subsequent requests will be sent to the server that initially
answered the first block request. After loading, \fBinetboot\fR will also use
```

reverse ARP to fetch its IP address, then broadcast \fBbootparams\fR RPC calls (see \fBbootparams\fR(4)) to locate configuration information and its root file system. \fBinetboot\fR then loads the boot archive by means of NFS and transfers control to that archive.
.sp
.LP
When booting over the network using DHCP, the PROM broadcasts the hardware address and kernel architecture and requests an IP address, boot parameters, and network configuration information. After a DHCP server responds and is selected (from among potentially multiple servers), that server sends to the client an IP address and all other information needed to boot the client. After receipt of this information, the client PROM examines the name of the file to be loaded, and will behave in one of two ways, depending on whether the file's name appears to be an HTTP URL. If it does not, the PROM downloads \fBinetboot\fR, loads that file into memory, and executes it. \fBinetboot\fR loads the boot archive, which takes over the machine and releases \fBinetboot\fR. Startup scripts then initiate the DHCP agent (see \fBdhcpagent\fR(1M)), which implements further DHCP activities.

.SS "iSCSI Boot"
.LP
iSCSI boot is currently supported only on x86. The host being booted must be equipped with NIC(s) capable of iBFT (iSCSI Boot Firmware Table) or have the mainboard's BIOS be iBFT-capable. iBFT, defined in the Advanced Configuration and Power Interface (ACPI) 3.0b specification, specifies a block of information that contains various parameters that are useful to the iSCSI Boot process.
.sp
.LP
Firmware implementing iBFT presents an iSCSI disk in the BIOS during startup as a bootable device by establishing the connection to the iSCSI target. The rest of the process of iSCSI booting is the same as booting from a local disk.
.sp
.LP
To configure the iBFT properly, users need to refer to the documentation from their hardware vendors.
.SS "Booting from Disk"
.LP
When booting from disk, the OpenBoot PROM firmware reads the boot blocks from blocks 1 to 15 of the partition specified as the boot device. This standalone booter usually contains a file system-specific reader capable of reading the boot archive.
.sp
.LP
If the pathname to the standalone is relative (does not begin with a slash), the second level boot will look for the standalone in a platform-dependent search path. This path is guaranteed to contain \fB/platform/\fR\fIplatform-name\fR. Many SPARC platforms next search the platform-specific path entry \fB/platform/\fR\fIhardware-class-name\fR. See \fBfilesystem\fR(5). If the pathname is absolute, \fBboot\fR will use the specified path. The \fBboot\fR program then loads the standalone at the appropriate address, and then transfers control.
.sp
.LP
Once the boot archive has been transferred from the boot device, Solaris can initialize and take over control of the machine. This process is further described in the "Boot Archive Phase," below, and is identical on all platforms.
.sp
.LP
If the filename is not given on the command line or otherwise specified, for example, by the \fBboot-file\fR NVRAM variable, \fBboot\fR chooses an appropriate default file to load based on what software is installed on the system and the capabilities of the hardware and firmware.
.sp
.LP
The path to the kernel must not contain any whitespace.

.SS "Booting from ZFS"
.LP
Booting from ZFS differs from booting from UFS in that, with ZFS, a device specifier identifies a storage pool, not a single root file system. A storage pool can contain multiple bootable datasets (that is, root file systems). Therefore, when booting from ZFS, it is not sufficient to specify a boot device. One must also identify a root file system within the pool that was identified by the boot device. By default, the dataset selected for booting is the one identified by the pool's \fBbootfs\fR property. This default selection can be overridden by specifying an alternate bootable dataset with the \fB-Z\fR option.
.SS "Boot Archive Phase"
.LP
The boot archive contains a file system image that is mounted using an in-memory disk. The image is self-describing, specifically containing a file system reader in the boot block. This file system reader mounts and opens the RAM disk image, then reads and executes the kernel contained within it. By default, this kernel is in:
.sp
.in +2
.nf
/platform/‘uname -i‘/kernel/unix
.fi
.in -2
.sp

.sp
.LP
If booting from ZFS, the pathnames of both the archive and the kernel file are resolved in the root file system (that is, dataset) selected for booting as described in the previous section.
.sp
.LP
The initialization of the kernel continues by loading necessary drivers and modules from the in-memory filesystem until I/O can be turned on and the root filesystem mounted. Once the root filesystem is mounted, the in-memory filesystem is no longer needed and is discarded.
.SS "OpenBoot PROM \fBboot\fR Command Behavior"
.LP
The OpenBoot \fBboot\fR command takes arguments of the following form:
.sp
.in +2
.nf
ok boot [\fIdevice-specifier\fR] [\fIarguments\fR]
.fi
.in -2
.sp

.sp
.LP
The default \fBboot\fR command has no arguments:
.sp
.in +2
.nf
ok boot
.fi
.in -2
.sp

.sp
.LP
If no \fIdevice-specifier\fR is given on the \fBboot\fR command line, OpenBoot typically uses the \fIboot-device\fR or \fIdiag-device\fR \fBNVRAM\fR variable. If no optional \fIarguments\fR are given on the command line, OpenBoot typically uses the \fIboot-file\fR or \fIdiag-file\fR \fBNVRAM\fR variable as default \fBboot\fR arguments. (If the system is in diagnostics mode,

\fIdiag-device\fR and \fIdiag-file\fR are used instead of \fIboot-device\fR and \fIboot-file\fR).
.sp
.LP
\fIarguments\fR may include more than one string. All \fIargument\fR strings are passed to the secondary booter; they are not interpreted by OpenBoot.
.sp
.LP
If any \fIarguments\fR are specified on the \fBboot\fR command line, then neither the \fIboot-file\fR nor the \fIdiag-file\fR \fBNVRAM\fR variable is used. The contents of the \fBNVRAM\fR variables are not merged with command line arguments. For example, the command:
.sp
.in +2
.nf
ok \fBboot\fR \fB-s\fR
.fi
.in -2
.sp

.sp
.LP
ignores the settings in both \fIboot-file\fR and \fIdiag-file\fR; it interprets the string \fB"-s"\fR as \fIarguments\fR. \fBboot\fR will not use the contents of \fIboot-file\fR or \fIdiag-file\fR.
.sp
.LP
With older PROMs, the command:
.sp
.in +2
.nf
ok \fBboot net\fR
.fi
.in -2
.sp

.sp
.LP
took no arguments, using instead the settings in \fIboot-file\fR or \fIdiag-file\fR (if set) as the default file name and arguments to pass to boot. In most cases, it is best to allow the \fBboot\fR command to choose an appropriate default based upon the system type, system hardware and firmware, and upon what is installed on the root file system. Changing \fIboot-file\fR or \fIdiag-file\fR can generate unexpected results in certain circumstances.
.sp
.LP
This behavior is found on most OpenBoot 2.x and 3.x based systems. Note that differences may occur on some platforms.
.sp
.LP
The command:
.sp
.LP
.nf
ok \fBboot cdrom\fR
.sp
.LP
\&...also normally takes no arguments. Accordingly, if \fIboot-file\fR is set to the 64-bit kernel filename and you attempt to boot the installation CD or DVD with \fBboot cdrom\fR, boot will fail if the installation media contains only a 32-bit kernel.
.sp
.LP
Because the contents of \fIboot-file\fR or \fIdiag-file\fR can be ignored depending on the form of the \fBboot\fR command used, reliance upon \fIboot-file\fR should be discouraged for most production systems.
.sp

.LP
Modern PROMs have enhanced the network boot support package to support the following syntax for arguments to be processed by the package:
.sp
.LP
[\fIprotocol\fR,] [\fIkey\fR=\fIvalue\fR,]*
.sp
.LP
All arguments are optional and can appear in any order. Commas are required unless the argument is at the end of the list. If specified, an argument takes precedence over any default values, or, if booting using DHCP, over configuration information provided by a DHCP server for those parameters.
.sp
.LP
\fIprotocol\fR, above, specifies the address discovery protocol to be used.
.sp
.LP
Configuration parameters, listed below, are specified as \fIkey\fR=\fIvalue\fR attribute pairs.
.sp
.ne 2
.na
\fB\fBtftp-server\fR\fR
.ad
.sp .6
.RS 4n
IP address of the TFTP server
.RE

.sp
.ne 2
.na
\fB\fBfile\fR\fR
.ad
.sp .6
.RS 4n
file to download using TFTP
.RE

.sp
.ne 2
.na
\fB\fBhost-ip\fR\fR
.ad
.sp .6
.RS 4n
IP address of the client (in dotted-decimal notation)
.RE

.sp
.ne 2
.na
\fB\fBrouter-ip\fR\fR
.ad
.sp .6
.RS 4n
IP address of the default router
.RE

.sp
.ne 2
.na
\fB\fBsubnet-mask\fR\fR
.ad
.sp .6
.RS 4n

```
388 subnet mask (in dotted-decimal notation)
389 .RE

391 .sp
392 .ne 2
393 .na
394 \fB\fBclient-id\fR\fR
395 .ad
396 .sp .6
397 .RS 4n
398 DHCP client identifier
399 .RE

401 .sp
402 .ne 2
403 .na
404 \fB\fBhostname\fR\fR
405 .ad
406 .sp .6
407 .RS 4n
408 hostname to use in DHCP transactions
409 .RE

411 .sp
412 .ne 2
413 .na
414 \fB\fBhttp-proxy\fR\fR
415 .ad
416 .sp .6
417 .RS 4n
418 HTTP proxy server specification (IPADDR[:PORT])
419 .RE

421 .sp
422 .ne 2
423 .na
424 \fB\fBtftp-retries\fR\fR
425 .ad
426 .sp .6
427 .RS 4n
428 maximum number of TFTP retries
429 .RE

431 .sp
432 .ne 2
433 .na
434 \fB\fBdhcp-retries\fR\fR
435 .ad
436 .sp .6
437 .RS 4n
438 maximum number of DHCP retries
439 .RE

441 .sp
442 .LP
443 The list of arguments to be processed by the network boot support package is
444 specified in one of two ways:
445 .RS +4
446 .TP
447 .ie t \(bu
448 .el o
449 As arguments passed to the package’s \fBopen\fR method, or
450 .RE
451 .RS +4
452 .TP
453 .ie t \(bu
```



```
454 .el o
455 arguments listed in the NVRAM variable \fBnetwork-boot-arguments\fR.
456 .RE
457 .sp
458 .LP
459 Arguments specified in \fBnetwork-boot-arguments\fR will be processed only if
460 there are no arguments passed to the package’s \fBopen\fR method.
461 .sp
462 .LP
463 Argument Values
464 .sp
465 .LP
466 \fIprotocol\fR specifies the address discovery protocol to be used. If present,
467 the possible values are \fBrarp\fR or \fBdhcp\fR.
468 .sp
469 .LP
470 If other configuration parameters are specified in the new syntax and style
471 specified by this document, absence of the \fIprotocol\fR parameter implies
472 manual configuration.
473 .sp
474 .LP
475 If no other configuration parameters are specified, or if those arguments are
476 specified in the positional parameter syntax currently supported, the absence
477 of the \fIprotocol\fR parameter causes the network boot support package to use
478 the platform-specific default address discovery protocol.
479 .sp
480 .LP
481 Manual configuration requires that the client be provided its IP address, the
482 name of the boot file, and the address of the server providing the boot file
483 image. Depending on the network configuration, it might be required that
484 \fBsubnet-mask\fR and \fBrouter-ip\fR also be specified.
485 .sp
486 .LP
487 If the \fIprotocol\fR argument is not specified, the network boot support
488 package uses the platform-specific default address discovery protocol.
489 .sp
490 .LP
491 \fBtftp-server\fR is the IP address (in standard IPv4 dotted-decimal notation)
492 of the TFTP server that provides the file to download if using TFTP.
493 .sp
494 .LP
495 When using DHCP, the value, if specified, overrides the value of the TFTP
496 server specified in the DHCP response.
497 .sp
498 .LP
499 The TFTP RRQ is unicast to the server if one is specified as an argument or in
500 the DHCP response. Otherwise, the TFTP RRQ is broadcast.
501 .sp
502 .LP
503 \fIfile\fR specifies the file to be loaded by TFTP from the TFTP server.
504 .sp
505 .LP
506 When using RARP and TFTP, the default file name is the ASCII hexadecimal
507 representation of the IP address of the client, as documented in a preceding
508 section of this document.
509 .sp
510 .LP
511 When using DHCP, this argument, if specified, overrides the name of the boot
512 file specified in the DHCP response.
513 .sp
514 .LP
515 When using DHCP and TFTP, the default file name is constructed from the root
516 node’s \fBname\fR property, with commas (,) replaced by periods (.).
517 .sp
518 .LP
519 When specified on the command line, the filename must not contain slashes
```

(\fB/\fR).
.sp
.LP
\fBhost-ip\fR specifies the IP address (in standard IPv4 dotted-decimal notation) of the client, the system being booted. If using RARP as the address discovery protocol, specifying this argument makes use of RARP unnecessary.
.sp
.LP
If DHCP is used, specifying the \fBhost-ip\fR argument causes the client to follow the steps required of a client with an "Externally Configured Network Address", as specified in RFC 2131.
.sp
.LP
\fBrouter-ip\fR is the IP address (in standard IPv4 dotted-decimal notation) of a router on a directly connected network. The router will be used as the first hop for communications spanning networks. If this argument is supplied, the router specified here takes precedence over the preferred router specified in the DHCP response.
.sp
.LP
\fBsubnet-mask\fR (specified in standard IPv4 dotted-decimal notation) is the subnet mask on the client's network. If the subnet mask is not provided (either by means of this argument or in the DHCP response), the default mask appropriate to the network class (Class A, B, or C) of the address assigned to the booting client will be assumed.
.sp
.LP
\fBclient-id\fR specifies the unique identifier for the client. The DHCP client identifier is derived from this value. Client identifiers can be specified as:
.RS +4
.TP
.ie t \(bu
.el o
The ASCII hexadecimal representation of the identifier, or
.RE
.RS +4
.TP
.ie t \(bu
.el o
a quoted string
.RE
.sp
.LP
Thus, \fBclient-id="openboot"\fR and \fBclient-id=6f70656e626f6f74\fR both represent a DHCP client identifier of 6F70656E626F6F74.
.sp
.LP
Identifiers specified on the command line must must not include slash (\fB/\fR) or spaces.
.sp
.LP
The maximum length of the DHCP client identifier is 32 bytes, or 64 characters representing 32 bytes if using the ASCII hexadecimal form. If the latter form is used, the number of characters in the identifier must be an even number. Valid characters are 0-9, a-f, and A-F.
.sp
.LP
For correct identification of clients, the client identifier must be unique among the client identifiers used on the subnet to which the client is attached. System administrators are responsible for choosing identifiers that meet this requirement.
.sp
.LP
Specifying a client identifier on a command line takes precedence over any other DHCP mechanism of specifying identifiers.
.sp

.LP
\fBhostname\fR (specified as a string) specifies the hostname to be used in DHCP transactions. The name might or might not be qualified with the local domain name. The maximum length of the hostname is 255 characters.
.LP
Note -
.sp
.RS 2
The \fBhostname\fR parameter can be used in service environments that require that the client provide the desired hostname to the DHCP server. Clients provide the desired hostname to the DHCP server, which can then register the hostname and IP address assigned to the client with DNS.
.RE
.sp
.LP
\fBhttp-proxy\fR is specified in the following standard notation for a host:
.sp
.in +2
.nf
\fIhost\fR [":"" \fIport\fR]
.fi
.in -2
.sp

.sp
.LP
\&...where \fIhost\fR is specified as an IP ddress (in standard IPv4 dotted-decimal notation) and the optional \fIport\fR is specified in decimal. If a port is not specified, port 8080 (decimal) is implied.
.sp
.LP
\fBtftp-retries\fR is the maximum number of retries (specified in decimal) attempted before the TFTP process is determined to have failed. Defaults to using infinite retries.
.sp
.LP
\fBdhcp-retries\fR is the maximum number of retries (specified in decimal) attempted before the DHCP process is determined to have failed. Defaults to of using infinite retries.
.SS "x86 Bootstrap Procedure"
.LP
On x86 based systems, the bootstrapping process consists of two conceptually distinct phases, kernel loading and kernel initialization. Kernel loading is implemented in the boot loader using the BIOS ROM on the system board, and BIOS extensions in ROMs on peripheral boards. The BIOS loads boot loader, starting with the first physical sector from a hard disk, DVD, or CD. If supported by the ROM on the network adapter, the BIOS can also download the \fBpxeboot\fR binary from a network boot server. Once the boot loader is loaded, it in turn will load the \fBunix\fR kernel, a pre-constructed boot archive containing kernel modules and data, and any additional files specified in the boot loader configuration. Once specified files are loaded, the boot loader will start the kernel to complete boot.
.sp
.LP
If the device identified by the boot loader as the boot device contains a ZFS storage pool, the \fBmenu.lst\fR file used to create the Boot Environment menu will be found in the dataset at the root of the pool's dataset hierarchy. This is the dataset with the same name as the pool itself. There is always exactly one such dataset in a pool, and so this dataset is well-suited for pool-wide data such as the \fBmenu.lst\fR file. After the system is booted, this dataset is mounted at /\fIpoolname\fR in the root file system.
.sp
.LP
There can be multiple bootable datasets (that is, root file systems) within a pool. The default file system to load the kernel is identified by the boot pool \fBbootfs\fR property (see \fBzpool\fR(1M)). All bootable datasets are

```
652 listed in the \fBmenu.lst\fR file, which is used by the boot loader to compose
653 the Boot Environment menu, to implement support to load a kernel and boot from
654 an alternate Boot Environment.
655 .sp
656 .LP
657 Kernel initialization starts when the boot loader finishes loading the files
658 specified in the boot loader configuration and hands control over to the
659 \fBunix\fR binary. The Unix operating system initializes, links in the
660 necessary modules from the boot archive and mounts the root file system on
661 the real root device. At this point, the kernel regains
662 storage I/O, mounts additional file systems (see \fBvfstab\fR(4)), and starts
663 various operating system services (see \fBsmf\fR(5)).

665 .SH OPTIONS
666 .SS "SPARC"
667 .LP
668 The following SPARC options are supported:
669 .sp
670 .ne 2
671 .na
672 \fB\fB-a\fR\fR
673 .ad
674 .sp .6
675 .RS 4n
676 The boot program interprets this flag to mean \fBask me\fR, and so it prompts
677 for the name of the standalone. The \fB\&'\fR\fB-a\fR\fB\&'\fR flag is then
678 passed to the standalone program.
679 .RE

681 .sp
682 .ne 2
683 .na
684 \fB\fB-D\fR \fIdefault-file\fR\fR
685 .ad
686 .sp .6
687 .RS 4n
688 Explicitly specify the \fIdefault-file\fR. On some systems, \fBboot\fR chooses
689 a dynamic default file, used when none is otherwise specified. This option
690 allows the \fIdefault-file\fR to be explicitly set and can be useful when
691 booting \fBkmdb\fR(1) since, by default, \fBkmdb\fR loads the default-file as
692 exported by the \fBboot\fR program.
693 .RE

695 .sp
696 .ne 2
697 .na
698 \fB\fB-F\fR \fIobject\fR\fR
699 .ad
700 .sp .6
701 .RS 4n
702 Boot using the named object. The object must be either an ELF executable or
703 bootable object containing a boot block. The primary use is to boot the
704 failsafe boot archive.
705 .RE

707 .sp
708 .ne 2
709 .na
710 \fB\fB-L\fR\fR
711 .ad
712 .sp .6
713 .RS 4n
714 List the bootable datasets within a ZFS pool. You can select one of the
715 bootable datasets in the list, after which detailed instructions for booting
716 that dataset are displayed. Boot the selected dataset by following the
717 instructions. This option is supported only when the boot device contains a ZFS
```



```
718 storage pool.
719 .RE

721 .sp
722 .ne 2
723 .na
724 \fB\fB-V\fR\fR
725 .ad
726 .sp .6
727 .RS 4n
728 Display verbose debugging information.
729 .RE

731 .sp
732 .ne 2
733 .na
734 \fB\fIboot-flags\fR\fR
735 .ad
736 .sp .6
737 .RS 4n
738 The boot program passes all \fIboot-flags\fR to \fBfile\fR. They are not
739 interpreted by \fBboot\fR. See the \fBkernel\fR(1M) and \fBkmdb\fR(1) manual
740 pages for information about the options available with the default standalone
741 program.
742 .RE

744 .sp
745 .ne 2
746 .na
747 \fB\fIclient-program-args\fR\fR
748 .ad
749 .sp .6
750 .RS 4n
751 The \fBboot\fR program passes all \fIclient-program-args\fR to \fIfile\fR. They
752 are not interpreted by \fBboot\fR.
753 .RE

755 .sp
756 .ne 2
757 .na
758 \fB\fIfile\fR\fR
759 .ad
760 .sp .6
761 .RS 4n
762 Name of a standalone program to \fBboot\fR. If a filename is not explicitly
763 specified, either on the \fBboot\fR command line or in the \fIboot-file\fR
764 NVRAM variable, \fBboot\fR chooses an appropriate default filename.
765 .RE

767 .sp
768 .ne 2
769 .na
770 \fB\fIOBP\fR \fInames\fR\fR
771 .ad
772 .sp .6
773 .RS 4n
774 Specify the open boot prom designations. For example, on Desktop SPARC based
775 systems, the designation \fB/sbus/esp@0,800000/sd@3,0:a\fR indicates a
776 \fBSCSI\fR disk (sd) at target 3, lun0 on the \fBSCSI\fR bus, with the esp host
777 adapter plugged into slot 0.
778 .RE

780 .sp
781 .ne 2
782 .na
783 \fB\fB-Z\fR \fIdataset\fR\fR
```

```
.ad
.sp .6
.RS 4n
Boot from the root file system in the specified ZFS dataset.
.RE

.SS "x86"
.LP
The following x86 options are supported:
.sp
.ne 2
.na
\fB\fB-B\fR \fIprop\fR=\fIval\fR...\fR
.ad
.sp .6
.RS 4n
One or more property-value pairs to be passed to the kernel. Multiple
property-value pairs must be separated by a comma. Use of this option is the
equivalent of the command: \fBeeprom\fR \fIprop\fR=\fIval\fR. See
\fBeeprom\fR(1M) for available properties and valid values.
.RE

.sp
.ne 2
.na
\fB\fIboot-flags\fR\fR
.ad
.sp .6
.RS 4n
The boot program passes all \fIboot-flags\fR to \fBfile\fR. They are not
interpreted by \fBboot\fR. See \fBkernel\fR(1M) and \fBkmdb\fR(1) for
information about the options available with the kernel.
.RE

.SH X86 BOOT SEQUENCE DETAILS
.LP
After a PC-compatible machine is turned on, the system firmware in the \fBBIOS
ROM\fR executes a power-on self test (POST), runs \fBBIOS\fR extensions in
peripheral board \fBROMs,\fR and invokes software interrupt INT 19h, Bootstrap.
The INT 19h handler typically performs the standard PC-compatible boot, which
consists of trying to read the first physical sector from the first diskette
drive, or, if that fails, from the first hard disk. The processor then jumps to
the first byte of the sector image in memory.
.SH X86 PRIMARY BOOT
.LP
The first sector on a hard disk contains the master boot block (first stage of
the boot program), which contains the master boot program and the Master Boot
Record (\fBMBR\fR) table. The master boot program has recorded the location of
the secondary stage of the boot program and using this location, master boot
will load and start the secondary stage of the boot program.

To support booting multiple operating systems, the master boot program is also
installed as the first sector of the partition with the illumos root file
system. This will allow configuring third party boot programs to use the
chainload technique to boot illumos system.

If the first stage is installed on the master boot block (see the \fB-m\fR
option of \fBinstallboot\fR(1M)), then \fBstage2\fR is loaded directly
from the Solaris partition regardless of the active partition.
.sp
.LP
A similar sequence occurs for DVD or CD boot, but the master boot block location
and contents are dictated by the El Torito specification. The El Torito boot
will then continue in the same way as with the hard disk.
.sp
.LP
```



```
Floppy booting is not longer supported. Booting from USB devices follows the
same procedure as with hard disks.
.sp
.LP
An x86 \fBMBR\fR partition for the Solaris software begins with a
one-cylinder boot slice, which contains the boot loader \fBstage1\fR in the
first sector, the standard Solaris disk label and volume table of contents
(VTOC) in the second and third sectors, and in case the UFS file system is
used for the root file system, \fBstage2\fR in the fiftieth and subsequent
sectors.

If the zfs boot is used, \fBstage2\fR is always stored in the zfs pool
boot program area.
.sp
.LP
The behavior is slightly different when a disk is using \fBEFI\fR
partitioning.

To support a UFS root file system in the \fBEFI\fR partition, the \fBstage2\fR
must be stored on separate dedicated partition, as there is no space in UFS
file system boot program area to store the current \fBstage2\fR. This separate
dedicated partition is used as raw disk space, and must have enough space
for both \fBstage1\fR and \fBstage2\fR. The type (tag) of this partition
must be \fBboot\fR, \fBEFI\fR UUID:
.sp
.in +2
.nf
\fB6a82cb45-1dd2-11b2-99a6-080020736631\fR
.fi
.in -2
.sp
For the UUID reference, please see \fB/usr/include/sys/efi_partition.h\fR.

In case of a whole disk zfs pool configuration, the \fBstage1\fR is always
installed in the first sector of the disk, and it always loads \fBstage2\fR
from the partition specified at the boot loader installation time.
.sp
.LP
Once \fBstage2\fR is running, it will load and start the third stage boot
program from root file system. Boot loader supports loading from the ZFS,
UFS and PCFS file systems. The stage3 boot program defaults to be
\fB/boot/loader\fR, and implements a user interface to load and boot the
unix kernel.
.sp
.LP
For network booting, the supported method is Intel’s Preboot eXecution
Environment (PXE) standard. When booting from the network using PXE, the system
or network adapter BIOS uses DHCP to locate a network bootstrap program
(\fBpxeboot\fR) on a boot server and reads it using Trivial File Transfer
Protocol (TFTP). The BIOS executes the \fBpxeboot\fR by jumping to its first
byte in memory. The \fBpxeboot\fR program is combined stage2 and stage2 boot
program and implements user interface to load and boot unix kernel.
.SH X86 KERNEL STARTUP
.LP
The kernel startup process is independent of the kernel loading process. During
kernel startup, console I/O goes to the device specified by the \fBconsole\fR
property.
.sp
.LP
When booting from UFS, the root device is specified by the \fBbootpath\fR
property, and the root file system type is specified by the \fBfstype\fR
property. These properties should be setup by the Solaris Install/Upgrade
process in \fB/boot/solaris/bootenv.rc\fR and can be overridden with the
\fB-B\fR option, described above (see the \fBeeprom\fR(1M) man page).
.sp
.LP
```

```
916 When booting from ZFS, the root device is automatically passed by the boot
917 loader to the kernel as a boot parameter \fB-B\fR \fBzfs-bootfs\fR. The actual
918 value used by the boot loader can be observed with the \fBeeprom bootcmd\fR
919 command.
920 .sp
921 .LP
922 If the console properties are not present, console I/O defaults to \fBscreen\fR
923 and \fBkeyboard\fR. The root device defaults to \fBramdisk\fR and the file
924 system defaults to \fBufs\fR.
925 .SH EXAMPLES
926 .SS "SPARC"
927 .LP
928 \fBExample 1 \fRTo Boot the Default Kernel In Single-User Interactive Mode
929 .sp
930 .LP
931 To boot the default kernel in single-user interactive mode, respond to the
932 \fBok\fR prompt with one of the following:

934 .sp
935 .in +2
936 .nf
937 \fBboot\fR \fB\fR\fB-as\fR

939 \fBboot\fR \fBdisk3\fR \fB-as\fR
940 .fi
941 .in -2
942 .sp

944 .LP
945 \fBExample 2 \fRNetwork Booting
946 .sp
947 .LP
948 To illustrate some of the subtle repercussions of various boot command line
949 invocations, assume that the \fBnetwork-boot-arguments\fR are set and that
950 \fBnet\fR is devaliased as shown in the commands below.

952 .sp
953 .LP
954 In the following command, device arguments in the device alias are processed by
955 the device driver. The network boot support package processes arguments in
956 \fBnetwork-boot-arguments\fR.

958 .sp
959 .in +2
960 .nf
961 \fBboot net\fR
962 .fi
963 .in -2
964 .sp

966 .sp
967 .LP
968 The command below results in no device arguments. The network boot support
969 package processes arguments in \fBnetwork-boot-arguments\fR.

971 .sp
972 .in +2
973 .nf
974 \fBboot net:\fR
975 .fi
976 .in -2
977 .sp

979 .sp
980 .LP
981 The command below results in no device arguments. \fBrarp\fR is the only
```



```
982 network boot support package argument. \fBnetwork-boot-arguments\fR is ignored.

984 .sp
985 .in +2
986 .nf
987 \fBboot net:rarp\fR
988 .fi
989 .in -2
990 .sp

992 .sp
993 .LP
994 In the command below, the specified device arguments are honored. The network
995 boot support package processes arguments in \fBnetwork-boot-arguments\fR.

997 .sp
998 .in +2
999 .nf
1000 \fBboot net:speed=100,duplex=full\fR
1001 .fi
1002 .in -2
1003 .sp

1005 .SS "x86"
1006 .LP
1007 \fBExample 3 \fRTo Boot the Default Kernel In 64-bit Single-User Interactive
1008 Mode
1009 .sp
1010 .LP
1011 To boot the default kernel in single-user interactive mode, press the ESC key
1012 to get the boot loader \fBok\fR prompt and enter:

1014 .sp
1015 .in +2
1016 .nf
1017 boot -as
1018 .fi
1019 .in -2

1021 .SH FILES
1022 .ne 2
1023 .na
1024 \fB\fB/etc/inittab\fR\fR
1025 .ad
1026 .sp .6
1027 .RS 4n
1028 Table in which the \fBinitdefault\fR state is specified
1029 .RE

1031 .sp
1032 .ne 2
1033 .na
1034 \fB\fB/sbin/init\fR\fR
1035 .ad
1036 .sp .6
1037 .RS 4n
1038 Program that brings the system to the \fBinitdefault\fR state
1039 .RE

1041 .SS "64-bit SPARC Only"
1042 .ne 2
1043 .na
1044 \fB\fB/platform/\fR\fIplatform-name\fR\fB/kernel/sparcv9/unix\fR\fR
1045 .ad
1046 .sp .6
1047 .RS 4n
```

```
1048 Default program to boot system.
1049 .RE

1051 .SS "x86 Only"
1052 .ne 2
1053 .na
1054 \fB\fB/boot\fR\fR
1055 .ad
1056 .sp .6
1057 .RS 4n
1058 Directory containing boot-related files.
1059 .RE

1061 .sp
1062 .ne 2
1063 .na
1064 \fB\fB/rpool/boot/menu.lst\fR\fR
1065 .ad
1066 .sp .6
1067 .RS 4n
1068 Menu index file of bootable operating systems displayed by the boot loader.
1069 .sp
1070 \fBNote:\fR this file is located on the root ZFS pool. While many installs
1071 often name their root zpool 'rpool', this is not required and the
1072 /rpool in the path above should be substituted with the name of
1073 the root pool of your current system.
1074 .RE

1076 .sp
1077 .ne 2
1078 .na
1079 \fB\fB/platform/i86pc/kernel/unix\fR\fR
1080 .ad
1081 .sp .6
1082 .RS 4n
1083 32-bit kernel.
1084 .RE

1086 .SS "64-bit x86 Only"
1087 .ne 2
1088 .na
1089 \fB\fB/platform/i86pc/kernel/amd64/unix\fR\fR
1090 .ad
1091 .sp .6
1092 .RS 4n
1093 64-bit kernel.
1094 .RE

1096 .SH SEE ALSO
1097 .LP
1098 \fBkmdb\fR(1), \fBuname\fR(1), \fBbootadm\fR(1M), \fBeeprom\fR(1M),
1099 \fBinit\fR(1M), \fBinstallboot\fR(1M), \fBkernel\fR(1M), \fBmonitor\fR(1M),
1100 \fBshutdown\fR(1M), \fBsvcadm\fR(1M), \fBumountall\fR(1M), \fBzpool\fR(1M),
1101 \fBuadmin\fR(2), \fBbootparams\fR(4), \fBinittab\fR(4), \fBvfstab\fR(4),
1102 \fBfilesystem\fR(5)
1103 .sp
1104 .LP
1105 RFC 903, \fIA Reverse Address Resolution Protocol\fR,
1106 \fBhttp://www.ietf.org/rfc/rfc903.txt\fR
1107 .sp
1108 .LP
1109 RFC 2131, \fIDynamic Host Configuration Protocol\fR,
1110 \fBhttp://www.ietf.org/rfc/rfc2131.txt\fR
1111 .sp
1112 .LP
1113 RFC 2132, \fIDHCP Options and BOOTP Vendor Extensions\fR,
```



```
1114 \fBhttp://www.ietf.org/rfc/rfc2132.txt\fR
1115 .sp
1116 .LP
1117 RFC 2396, \fIUniform Resource Identifiers (URI): Generic Syntax\fR,
1118 \fBhttp://www.ietf.org/rfc/rfc2396.txt\fR
1119 .sp
1120 .LP
1121 \fI\fR
1122 .sp
1123 .LP
1124 \fISun Hardware Platform Guide\fR
1125 .sp
1126 .LP
1127 \fIOpenBoot Command Reference Manual\fR
1128 .SH WARNINGS
1129 .LP
1130 The \fBboot\fR utility is unable to determine which files can be used as
1131 bootable programs. If the booting of a file that is not bootable is requested,
1132 the \fBboot\fR utility loads it and branches to it. What happens after that is
1133 unpredictable.
1134 .SH NOTES
1135 .LP
1136 \fIplatform-name\fR can be found using the \fB-i\fR option of \fBuname\fR(1).
1137 \fIhardware-class-name\fR can be found using the \fB-m\fR option of
1138 \fBuname\fR(1).
1139 .sp
1140 .LP
1141 The current release of the Solaris operating system does not support machines
1142 running an UltraSPARC-I CPU.
```

```
**********************************************************
    5031 Fri Jan 11 21:14:01 2019
new/usr/src/man/man1m/ypmap2src.1m
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
```

```
'\" te
.\" Copyright (C) 2003, 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 YPMAP2SRC 1M "Apr 10, 2003"
.SH NAME
ypmap2src \- convert NIS maps to NIS source files
.SH SYNOPSIS
.LP
.nf
\fB/usr/lib/netsvc/yp/ypmap2src\fR [\fB-t\fR]
     [ [\fB-c\fR \fIcustom-map-name\fR]]... [\fB-d\fR \fIdomain\fR] \fB-o\fR \fI
     [ [ \fIsource-file\fR]]...
.fi

.SH DESCRIPTION
.sp
.LP
Use the \fBypmap2src\fR utility to convert standard NIS maps to approximations
of the equivalent NIS source files. This utility functions like the reverse of
\fBypmake\fR(1M).
.sp
.LP
The primary use for \fBypmap2src\fR is to convert from a NIS server that uses
the NIS to LDAP(N2L) transition mechanism, which does not use NIS source files,
to traditional NIS, where source files are required. The \fBypmap2src\fR
utility is also used by NIS administrators who wish to discover the contents of
NIS maps for which the sources are not available.
.sp
.LP
Generally, this operation is not necessary. More often, administrators will
switch from traditional NIS to N2L in anticipation of the eventual transition
to LDAP naming. When this switch is made, authoritative information is moved
into the LDAP DIT, and the NIS sources have no further role. N2L supports NIS
clients until such time as they can be converted to LDAP, and the NIS service
suspended.
.sp
.LP
The \fBypmap2src\fR utility does not guarantee that the files that are
generated are identical to the original NIS source files. Some information
might have been thrown away by \fBypmake\fR and cannot be recovered. N2L also
might have updated the maps to reflect changes made by LDAP clients. It is
essential that the sources generated are checked to confirm no problems have
occurred.
.sp
.LP
Per entry comment fields, from existing source files, are not merged into
source files generated by \fBypmap2src\fR. If a user wishes N2L to maintain
comment information, then the \fBNISLDAPmapping\fR configuration file should be
modified so that the comment fields are mapped into  LDAP. This will ensure
that the comments are visible to native LDAP clients and present in the N2L map
files.
.sp
.LP
When \fBypmap2src\fR is run, it will take up-to-date comments from the map file
and insert them into the NIS source file generated.
.SS "Handling Custom Maps"
```



```
.sp
.LP
\fBypmap2src\fR only knows about the standard NIS maps and standard source to
map conversion. If an advanced user has changed these, that is, the user has
modified the NIS makefile, the equivalent changes must also be made to the
\fBypmap2src\fR script.
.SH OPTIONS
.sp
.LP
\fBypmap2src\fR supports the following options:
.sp
.ne 2
.na
\fB\fB-c\fR\fR
.ad
.RS 18n
Specifies that \fIcustom-map-name\fR should be converted to a source
file by running \fBmakedbm\fR \fB-u\fR on it. This is a short cut so that
simple custom maps can be handled without editing \fBypmap2src\fR.
.RE

.sp
.ne 2
.na
\fB\fB-d\fR \fIdomain-name\fR\fR
.ad
.RS 18n
 Specifies the domain to convert. The \fIdomain-name\fR can be a fully
qualified file path, such as \fB/var/yp/a.b.c\fR, or just a domain name,
a.b.c\fB\fR. In the latter case, \fBypmaptosrc\fR looks in \fB/var/yp\fR for
the domain directory.
.RE

.sp
.ne 2
.na
\fB\fB-o\fR \fIdest\fR\fR
.ad
.RS 18n
 Specifies the destination directory for the converted files. A directory other
than \fB/etc\fR should be specified. The maps generated are copied to the
correct location, \fB/etc\fR, \fB/etc/security\fR or other source directory, as
appropriate.
.RE

.sp
.ne 2
.na
\fB\fB-t\fR\fR
.ad
.RS 18n
Specifies that traditional NIS maps, without N2L's \fBLDAP_\fR prefix, should
be converted. By default, maps with the \fBLDAP_\fR prefix are converted.
.RE

.SH OPERANDS
.sp
.LP
\fBypmap2src\fR supports the following operands:
.sp
.ne 2
.na
\fB\fIsource-file\fR\fR
.ad
.RS 15n
Lists the standard source files to convert. If this option is not given, then
```

```
121 all the standard source files, plus any custom files specified by the \fB-c\fR
125 all the standard source files, plus any custom files pecified by the \fB-c\fR
122 option, are converted.
123 .RE

125 .SH ATTRIBUTES
130 .sp
126 .LP
127 See \fBattributes\fR(5) for descriptions of the following attributes:
128 .sp

130 .sp
131 .TS
132 box;
133 c | c
134 l | l .
135 ATTRIBUTE TYPE  ATTRIBUTE VALUE
136 _
137 Interface Stability     Obsolete
138 .TE

140 .SH SEE ALSO
146 .sp
141 .LP
142 \fBypmake\fR(1M), \fBypserv\fR(1M), \fBNISLDAPmapping\fR(4),
143 \fBattributes\fR(5)
144 .sp
145 .LP
146 \fI\fR
```

```
**********************************************************
  115315 Fri Jan 11 21:14:01 2019
new/usr/src/man/man1m/zfs.1m
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
```

```
.\"
.\" 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) 2009 Sun Microsystems, Inc. All Rights Reserved.
.\" Copyright 2011 Joshua M. Clulow <josh@sysmgr.org>
.\" Copyright (c) 2011, 2016 by Delphix. All rights reserved.
.\" Copyright (c) 2013 by Saso Kiselkov. All rights reserved.
.\" Copyright (c) 2014, Joyent, Inc. All rights reserved.
.\" Copyright (c) 2014 by Adam Stevko. All rights reserved.
.\" Copyright (c) 2014 Integros [integros.com]
.\" Copyright 2018 Nexenta Systems, Inc.
.\" Copyright 2018 Joyent, Inc.
.\" Copyright (c) 2018 Datto Inc.
.\"
.Dd Feb 10, 2018
.Dt ZFS 1M
.Os
.Sh NAME
.Nm zfs
.Nd configures ZFS file systems
.Sh SYNOPSIS
.Nm
.Op Fl \?
.Nm
.Cm create
.Op Fl p
.Oo Fl o Ar property Ns = Ns Ar value Oc Ns ...
.Ar filesystem
.Nm
.Cm create
.Op Fl ps
.Op Fl b Ar blocksize
.Oo Fl o Ar property Ns = Ns Ar value Oc Ns ...
.Fl V Ar size Ar volume
.Nm
.Cm destroy
.Op Fl Rfnprv
.Ar filesystem Ns | Ns Ar volume
.Nm
.Cm destroy
```



```
.Op Fl Rdnprv
.Ar filesystem Ns | Ns Ar volume Ns @ Ns Ar snap Ns
.Oo % Ns Ar snap Ns Oo , Ns Ar snap Ns Oo % Ns Ar snap Oc Oc Oc Ns ...
.Nm
.Cm destroy
.Ar filesystem Ns | Ns Ar volume Ns # Ns Ar bookmark
.Nm
.Cm snapshot
.Op Fl r
.Oo Fl o Ar property Ns = Ns value Oc Ns ...
.Ar filesystem Ns @ Ns Ar snapname Ns | Ns Ar volume Ns @ Ns Ar snapname Ns ...
.Nm
.Cm rollback
.Op Fl Rfr
.Ar snapshot
.Nm
.Cm clone
.Op Fl p
.Oo Fl o Ar property Ns = Ns Ar value Oc Ns ...
.Ar snapshot Ar filesystem Ns | Ns Ar volume
.Nm
.Cm promote
.Ar clone-filesystem
.Nm
.Cm rename
.Op Fl f
.Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot
.Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot
.Nm
.Cm rename
.Op Fl fp
.Ar filesystem Ns | Ns Ar volume
.Ar filesystem Ns | Ns Ar volume
.Nm
.Cm rename
.Fl r
.Ar snapshot Ar snapshot
.Nm
.Cm list
.Op Fl r Ns | Ns Fl d Ar depth
.Op Fl Hp
.Oo Fl o Ar property Ns Oo , Ns Ar property Oc Ns ... Oc
.Oo Fl s Ar property Oc Ns ...
.Oo Fl S Ar property Oc Ns ...
.Oo Fl t Ar type Ns Oo , Ns Ar type Oc Ns ... Oc
.Oo Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot Oc Ns ...
.Nm
.Cm remap
.Ar filesystem Ns | Ns Ar volume
.Nm
.Cm set
.Ar property Ns = Ns Ar value Oo Ar property Ns = Ns Ar value Oc Ns ...
.Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot Ns ...
.Nm
.Cm get
.Op Fl r Ns | Ns Fl d Ar depth
.Op Fl Hp
.Oo Fl o Ar field Ns Oo , Ns Ar field Oc Ns ... Oc
.Oo Fl s Ar source Ns Oo , Ns Ar source Oc Ns ... Oc
.Oo Fl t Ar type Ns Oo , Ns Ar type Oc Ns ... Oc
.Cm all | Ar property Ns Oo , Ns Ar property Oc Ns ...
.Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot Ns | Ns Ar bookmark Ns ...
.Nm
.Cm inherit
.Op Fl rS
.Ar property Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot Ns ...
```

```
.Nm
.Cm upgrade
.Nm
.Cm upgrade
.Fl v
.Nm
.Cm upgrade
.Op Fl r
.Op Fl V Ar version
.Fl a | Ar filesystem
.Nm
.Cm userspace
.Op Fl Hinp
.Oo Fl o Ar field Ns Oo , Ns Ar field Oc Ns ... Oc
.Oo Fl s Ar field Oc Ns ...
.Oo Fl S Ar field Oc Ns ...
.Oo Fl t Ar type Ns Oo , Ns Ar type Oc Ns ... Oc
.Ar filesystem Ns | Ns Ar snapshot
.Nm
.Cm groupspace
.Op Fl Hinp
.Oo Fl o Ar field Ns Oo , Ns Ar field Oc Ns ... Oc
.Oo Fl s Ar field Oc Ns ...
.Oo Fl S Ar field Oc Ns ...
.Oo Fl t Ar type Ns Oo , Ns Ar type Oc Ns ... Oc
.Ar filesystem Ns | Ns Ar snapshot
.Nm
.Cm mount
.Nm
.Cm mount
.Op Fl Ov
.Op Fl o Ar options
.Fl a | Ar filesystem
.Nm
.Cm unmount
.Op Fl f
.Fl a | Ar filesystem Ns | Ns Ar mountpoint
.Nm
.Cm share
.Fl a | Ar filesystem
.Nm
.Cm unshare
.Fl a | Ar filesystem Ns | Ns Ar mountpoint
.Nm
.Cm bookmark
.Ar snapshot bookmark
.Nm
.Cm send
.Op Fl DLPRcenpv
.Op Oo Fl I Ns | Ns Fl i Oc Ar snapshot
.Ar snapshot
.Nm
.Cm send
.Op Fl Lce
.Op Fl i Ar snapshot Ns | Ns Ar bookmark
.Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot
.Nm
.Cm send
.Op Fl Penv
.Fl t Ar receive_resume_token
.Nm
.Cm receive
.Op Fl Fnsuv
.Op Fl o Sy origin Ns = Ns Ar snapshot
.Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot
.Nm
```



```
.Cm receive
.Op Fl Fnsuv
.Op Fl d Ns | Ns Fl e
.Op Fl o Sy origin Ns = Ns Ar snapshot
.Ar filesystem
.Nm
.Cm receive
.Fl A
.Ar filesystem Ns | Ns Ar volume
.Nm
.Cm allow
.Ar filesystem Ns | Ns Ar volume
.Nm
.Cm allow
.Op Fl dglu
.Ar user Ns | Ns Ar group Ns Oo , Ns Ar user Ns | Ns Ar group Oc Ns ...
.Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
.Ar setname Oc Ns ...
.Ar filesystem Ns | Ns Ar volume
.Nm
.Cm allow
.Op Fl dl
.Fl e Ns | Ns Sy everyone
.Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
.Ar setname Oc Ns ...
.Ar filesystem Ns | Ns Ar volume
.Nm
.Cm allow
.Fl c
.Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
.Ar setname Oc Ns ...
.Ar filesystem Ns | Ns Ar volume
.Nm
.Cm allow
.Fl s No @ Ns Ar setname
.Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
.Ar setname Oc Ns ...
.Ar filesystem Ns | Ns Ar volume
.Nm
.Cm unallow
.Op Fl dglru
.Ar user Ns | Ns Ar group Ns Oo , Ns Ar user Ns | Ns Ar group Oc Ns ...
.Oo Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
.Ar setname Oc Ns ... Oc
.Ar filesystem Ns | Ns Ar volume
.Nm
.Cm unallow
.Op Fl dlr
.Fl e Ns | Ns Sy everyone
.Oo Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
.Ar setname Oc Ns ... Oc
.Ar filesystem Ns | Ns Ar volume
.Nm
.Cm unallow
.Op Fl r
.Fl c
.Oo Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
.Ar setname Oc Ns ... Oc
.Ar filesystem Ns | Ns Ar volume
.Nm
.Cm unallow
.Op Fl r
.Fl s @ Ns Ar setname
.Oo Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
.Ar setname Oc Ns ... Oc
.Ar filesystem Ns | Ns Ar volume
```

```
.Nm
.Cm hold
.Op Fl r
.Ar tag Ar snapshot Ns ...
.Nm
.Cm holds
.Op Fl r
.Ar snapshot Ns ...
.Nm
.Cm release
.Op Fl r
.Ar tag Ar snapshot Ns ...
.Nm
.Cm diff
.Op Fl FHt
.Ar snapshot Ar snapshot Ns | Ns Ar filesystem
.Nm
.Cm program
.Op Fl jn
.Op Fl t Ar timeout
.Op Fl m Ar memory_limit
.Ar pool script
.Op Ar arg1 No ...
.Sh DESCRIPTION
The
.Nm
command configures ZFS datasets within a ZFS storage pool, as described in
.Xr zpool 1M .
A dataset is identified by a unique path within the ZFS namespace.
For example:
.Bd -literal
pool/{filesystem,volume,snapshot}
.Ed
.Pp
where the maximum length of a dataset name is
.Dv MAXNAMELEN
.Pq 256 bytes
and the maximum amount of nesting allowed in a path is 50 levels deep.
.Pp
A dataset can be one of the following:
.Bl -tag -width "file system"
.It Sy file system
A ZFS dataset of type
.Sy filesystem
can be mounted within the standard system namespace and behaves like other file
systems.
While ZFS file systems are designed to be POSIX compliant, known issues exist
that prevent compliance in some cases.
Applications that depend on standards conformance might fail due to non-standard
behavior when checking file system free space.
.It Sy volume
A logical volume exported as a raw or block device.
This type of dataset should only be used under special circumstances.
File systems are typically used in most environments.
.It Sy snapshot
A read-only version of a file system or volume at a given point in time.
It is specified as
.Ar filesystem Ns @ Ns Ar name
or
.Ar volume Ns @ Ns Ar name .
.El
.Ss ZFS File System Hierarchy
A ZFS storage pool is a logical collection of devices that provide space for
datasets.
A storage pool is also the root of the ZFS file system hierarchy.
.Pp
```



```
The root of the pool can be accessed as a file system, such as mounting and
unmounting, taking snapshots, and setting properties.
The physical storage characteristics, however, are managed by the
.Xr zpool 1M
command.
.Pp
See
.Xr zpool 1M
for more information on creating and administering pools.
.Ss Snapshots
A snapshot is a read-only copy of a file system or volume.
Snapshots can be created extremely quickly, and initially consume no additional
space within the pool.
As data within the active dataset changes, the snapshot consumes more data than
would otherwise be shared with the active dataset.
.Pp
Snapshots can have arbitrary names.
Snapshots of volumes can be cloned or rolled back, but cannot be accessed
independently.
.Pp
File system snapshots can be accessed under the
.Pa .zfs/snapshot
directory in the root of the file system.
Snapshots are automatically mounted on demand and may be unmounted at regular
intervals.
The visibility of the
.Pa .zfs
directory can be controlled by the
.Sy snapdir
property.
.Ss Clones
A clone is a writable volume or file system whose initial contents are the same
as another dataset.
As with snapshots, creating a clone is nearly instantaneous, and initially
consumes no additional space.
.Pp
Clones can only be created from a snapshot.
When a snapshot is cloned, it creates an implicit dependency between the parent
and child.
Even though the clone is created somewhere else in the dataset hierarchy, the
original snapshot cannot be destroyed as long as a clone exists.
The
.Sy origin
property exposes this dependency, and the
.Cm destroy
command lists any such dependencies, if they exist.
.Pp
The clone parent-child dependency relationship can be reversed by using the
.Cm promote
subcommand.
This causes the
.Qq origin
file system to become a clone of the specified file system, which makes it
possible to destroy the file system that the clone was created from.
.Ss "Mount Points"
Creating a ZFS file system is a simple operation, so the number of file systems
per system is likely to be numerous.
To cope with this, ZFS automatically manages mounting and unmounting file
systems without the need to edit the
.Pa /etc/vfstab
file.
All automatically managed file systems are mounted by ZFS at boot time.
.Pp
By default, file systems are mounted under
.Pa /path ,
where
```

```
.Ar path
is the name of the file system in the ZFS namespace.
Directories are created and destroyed as needed.
.Pp
A file system can also have a mount point set in the
.Sy mountpoint
property.
This directory is created as needed, and ZFS automatically mounts the file
system when the
.Nm zfs Cm mount Fl a
command is invoked
.Po without editing
.Pa /etc/vfstab
.Pc .
The
.Sy mountpoint
property can be inherited, so if
.Em pool/home
has a mount point of
.Pa /export/stuff ,
then
.Em pool/home/user
automatically inherits a mount point of
.Pa /export/stuff/user .
.Pp
A file system
.Sy mountpoint
property of
.Sy none
prevents the file system from being mounted.
.Pp
If needed, ZFS file systems can also be managed with traditional tools
.Po
.Nm mount ,
.Nm umount ,
.Pa /etc/vfstab
.Pc .
If a file system's mount point is set to
.Sy legacy ,
ZFS makes no attempt to manage the file system, and the administrator is
responsible for mounting and unmounting the file system.
.Ss "Zones"
A ZFS file system can be added to a non-global zone by using the
.Nm zonecfg Cm add Sy fs
subcommand.
A ZFS file system that is added to a non-global zone must have its
.Sy mountpoint
property set to
.Sy legacy .
.Pp
The physical properties of an added file system are controlled by the global
administrator.
However, the zone administrator can create, modify, or destroy files within the
added file system, depending on how the file system is mounted.
.Pp
A dataset can also be delegated to a non-global zone by using the
.Nm zonecfg Cm add Sy dataset
subcommand.
You cannot delegate a dataset to one zone and the children of the same dataset
to another zone.
The zone administrator can change properties of the dataset or any of its
children.
However, the
.Sy quota ,
.Sy filesystem_limit
and
```



```
.Sy snapshot_limit
properties of the delegated dataset can be modified only by the global
administrator.
.Pp
A ZFS volume can be added as a device to a non-global zone by using the
.Nm zonecfg Cm add Sy device
subcommand.
However, its physical properties can be modified only by the global
administrator.
.Pp
For more information about
.Nm zonecfg
syntax, see
.Xr zonecfg 1M .
.Pp
After a dataset is delegated to a non-global zone, the
.Sy zoned
property is automatically set.
A zoned file system cannot be mounted in the global zone, since the zone
administrator might have to set the mount point to an unacceptable value.
.Pp
The global administrator can forcibly clear the
.Sy zoned
property, though this should be done with extreme care.
The global administrator should verify that all the mount points are acceptable
before clearing the property.
.Ss Native Properties
Properties are divided into two types, native properties and user-defined
.Po or
.Qq user
.Pc
properties.
Native properties either export internal statistics or control ZFS behavior.
In addition, native properties are either editable or read-only.
User properties have no effect on ZFS behavior, but you can use them to annotate
datasets in a way that is meaningful in your environment.
For more information about user properties, see the
.Sx User Properties
section, below.
.Pp
Every dataset has a set of properties that export statistics about the dataset
as well as control various behaviors.
Properties are inherited from the parent unless overridden by the child.
Some properties apply only to certain types of datasets
.Pq file systems, volumes, or snapshots .
.Pp
The values of numeric properties can be specified using human-readable suffixes
.Po for example,
.Sy k ,
.Sy KB ,
.Sy M ,
.Sy Gb ,
and so forth, up to
.Sy Z
for zettabyte
.Pc .
The following are all valid
.Pq and equal
specifications:
.Li 1536M, 1.5g, 1.50GB .
.Pp
The values of non-numeric properties are case sensitive and must be lowercase,
except for
.Sy mountpoint ,
.Sy sharenfs ,
and
```

```
.Sy sharesmb .
.Pp
The following native properties consist of read-only statistics about the
dataset.
These properties can be neither set, nor inherited.
Native properties apply to all dataset types unless otherwise noted.
.Bl -tag -width "usedbyrefreservation"
.It Sy available
The amount of space available to the dataset and all its children, assuming that
there is no other activity in the pool.
Because space is shared within a pool, availability can be limited by any number
of factors, including physical pool size, quotas, reservations, or other
datasets within the pool.
.Pp
This property can also be referred to by its shortened column name,
.Sy avail .
.It Sy compressratio
For non-snapshots, the compression ratio achieved for the
.Sy used
space of this dataset, expressed as a multiplier.
The
.Sy used
property includes descendant datasets, and, for clones, does not include the
space shared with the origin snapshot.
For snapshots, the
.Sy compressratio
is the same as the
.Sy refcompressratio
property.
Compression can be turned on by running:
.Nm zfs Cm set Sy compression Ns = Ns Sy on Ar dataset .
The default value is
.Sy off .
.It Sy createtxg
The transaction group (txg) in which the dataset was created.
Bookmarks have the same
.Sy createtxg
as the snapshot they are initially tied to.
This property is suitable for ordering a list of snapshots,
e.g. for incremental send and receive.
.It Sy creation
The time this dataset was created.
.It Sy clones
For snapshots, this property is a comma-separated list of filesystems or volumes
which are clones of this snapshot.
The clones'
.Sy origin
property is this snapshot.
If the
.Sy clones
property is not empty, then this snapshot can not be destroyed
.Po even with the
.Fl r
or
.Fl f
options
.Pc .
.It Sy defer_destroy
This property is
.Sy on
if the snapshot has been marked for deferred destroy by using the
.Nm zfs Cm destroy Fl d
command.
Otherwise, the property is
.Sy off .
.It Sy filesystem_count
```



```
The total number of filesystems and volumes that exist under this location in
the dataset tree.
This value is only available when a
.Sy filesystem_limit
has been set somewhere in the tree under which the dataset resides.
.It Sy guid
The 64 bit GUID of this dataset or bookmark which does not change over its
entire lifetime.
When a snapshot is sent to another pool, the received snapshot has the same
GUID.
Thus, the
.Sy guid
is suitable to identify a snapshot across pools.
.It Sy logicalreferenced
The amount of space that is
.Qq logically
accessible by this dataset.
See the
.Sy referenced
property.
The logical space ignores the effect of the
.Sy compression
and
.Sy copies
properties, giving a quantity closer to the amount of data that applications
see.
However, it does include space consumed by metadata.
.Pp
This property can also be referred to by its shortened column name,
.Sy lrefer .
.It Sy logicalused
The amount of space that is
.Qq logically
consumed by this dataset and all its descendents.
See the
.Sy used
property.
The logical space ignores the effect of the
.Sy compression
and
.Sy copies
properties, giving a quantity closer to the amount of data that applications
see.
However, it does include space consumed by metadata.
.Pp
This property can also be referred to by its shortened column name,
.Sy lused .
.It Sy mounted
For file systems, indicates whether the file system is currently mounted.
This property can be either
.Sy yes
or
.Sy no .
.It Sy origin
For cloned file systems or volumes, the snapshot from which the clone was
created.
See also the
.Sy clones
property.
.It Sy receive_resume_token
For filesystems or volumes which have saved partially-completed state from
.Sy zfs receive -s ,
this opaque token can be provided to
.Sy zfs send -t
to resume and complete the
.Sy zfs receive .
```

```
.It Sy referenced
The amount of data that is accessible by this dataset, which may or may not be
shared with other datasets in the pool.
When a snapshot or clone is created, it initially references the same amount of
space as the file system or snapshot it was created from, since its contents are
identical.
.Pp
This property can also be referred to by its shortened column name,
.Sy refer .
.It Sy refcompressratio
The compression ratio achieved for the
.Sy referenced
space of this dataset, expressed as a multiplier.
See also the
.Sy compressratio
property.
.It Sy snapshot_count
The total number of snapshots that exist under this location in the dataset
tree.
This value is only available when a
.Sy snapshot_limit
has been set somewhere in the tree under which the dataset resides.
.It Sy type
The type of dataset:
.Sy filesystem ,
.Sy volume ,
or
.Sy snapshot .
.It Sy used
The amount of space consumed by this dataset and all its descendents.
This is the value that is checked against this dataset's quota and reservation.
The space used does not include this dataset's reservation, but does take into
account the reservations of any descendent datasets.
The amount of space that a dataset consumes from its parent, as well as the
amount of space that is freed if this dataset is recursively destroyed, is the
greater of its space used and its reservation.
.Pp
The used space of a snapshot
.Po see the
.Sx Snapshots
section
.Pc
is space that is referenced exclusively by this snapshot.
If this snapshot is destroyed, the amount of
.Sy used
space will be freed.
Space that is shared by multiple snapshots isn't accounted for in this metric.
When a snapshot is destroyed, space that was previously shared with this
snapshot can become unique to snapshots adjacent to it, thus changing the used
space of those snapshots.
The used space of the latest snapshot can also be affected by changes in the
file system.
Note that the
.Sy used
space of a snapshot is a subset of the
.Sy written
space of the snapshot.
.Pp
The amount of space used, available, or referenced does not take into account
pending changes.
Pending changes are generally accounted for within a few seconds.
Committing a change to a disk using
.Xr fsync 3C
or
.Dv O_SYNC
does not necessarily guarantee that the space usage information is updated
```



```
immediately.
.It Sy usedby*
The
.Sy usedby*
properties decompose the
.Sy used
properties into the various reasons that space is used.
Specifically,
.Sy used No =
.Sy usedbychildren No +
.Sy usedbydataset No +
.Sy usedbyrefreservation No +
.Sy usedbysnapshots .
These properties are only available for datasets created on
.Nm zpool
.Qo version 13 Qc
pools.
.It Sy usedbychildren
The amount of space used by children of this dataset, which would be freed if
all the dataset's children were destroyed.
.It Sy usedbydataset
The amount of space used by this dataset itself, which would be freed if the
dataset were destroyed
.Po after first removing any
.Sy refreservation
and destroying any necessary snapshots or descendents
.Pc .
.It Sy usedbyrefreservation
The amount of space used by a
.Sy refreservation
set on this dataset, which would be freed if the
.Sy refreservation
was removed.
.It Sy usedbysnapshots
The amount of space consumed by snapshots of this dataset.
In particular, it is the amount of space that would be freed if all of this
dataset's snapshots were destroyed.
Note that this is not simply the sum of the snapshots'
.Sy used
properties because space can be shared by multiple snapshots.
.It Sy userused Ns @ Ns Em user
The amount of space consumed by the specified user in this dataset.
Space is charged to the owner of each file, as displayed by
.Nm ls Fl l .
The amount of space charged is displayed by
.Nm du
and
.Nm ls Fl s .
See the
.Nm zfs Cm userspace
subcommand for more information.
.Pp
Unprivileged users can access only their own space usage.
The root user, or a user who has been granted the
.Sy userused
privilege with
.Nm zfs Cm allow ,
can access everyone's usage.
.Pp
The
.Sy userused Ns @ Ns Em ...
properties are not displayed by
.Nm zfs Cm get Sy all .
The user's name must be appended after the @ symbol, using one of the following
forms:
.Bl -bullet -width ""
```

```
.It
.Em POSIX name
.Po for example,
.Sy joe
.Pc
.It
.Em POSIX numeric ID
.Po for example,
.Sy 789
.Pc
.It
.Em SID name
.Po for example,
.Sy joe.smith@mydomain
.Pc
.It
.Em SID numeric ID
.Po for example,
.Sy S-1-123-456-789
.Pc
.El
.It Sy userrefs
This property is set to the number of user holds on this snapshot.
User holds are set by using the
.Nm zfs Cm hold
command.
.It Sy groupused Ns @ Ns Em group
The amount of space consumed by the specified group in this dataset.
Space is charged to the group of each file, as displayed by
.Nm ls Fl l .
See the
.Sy userused Ns @ Ns Em user
property for more information.
.Pp
Unprivileged users can only access their own groups' space usage.
The root user, or a user who has been granted the
.Sy groupused
privilege with
.Nm zfs Cm allow ,
can access all groups' usage.
.It Sy volblocksize
For volumes, specifies the block size of the volume.
The
.Sy blocksize
cannot be changed once the volume has been written, so it should be set at
volume creation time.
The default
.Sy blocksize
for volumes is 8 Kbytes.
Any power of 2 from 512 bytes to 128 Kbytes is valid.
.Pp
This property can also be referred to by its shortened column name,
.Sy volblock .
.It Sy written
The amount of space
.Sy referenced
by this dataset, that was written since the previous snapshot
.Pq i.e. that is not referenced by the previous snapshot .
.It Sy written Ns @ Ns Em snapshot
The amount of
.Sy referenced
space written to this dataset since the specified snapshot.
This is the space that is referenced by this dataset but was not referenced by
the specified snapshot.
.Pp
The
```



```
.Em snapshot
may be specified as a short snapshot name
.Po just the part after the
.Sy @
.Pc ,
in which case it will be interpreted as a snapshot in the same filesystem as
this dataset.
The
.Em snapshot
may be a full snapshot name
.Po Em filesystem Ns @ Ns Em snapshot Pc ,
which for clones may be a snapshot in the origin's filesystem
.Pq or the origin of the origin's filesystem, etc.
.El
.Pp
The following native properties can be used to change the behavior of a ZFS
dataset.
.Bl -tag -width ""
.It Xo
.Sy aclinherit Ns = Ns Sy discard Ns | Ns Sy noallow Ns | Ns
.Sy restricted Ns | Ns Sy passthrough Ns | Ns Sy passthrough-x
.Xc
Controls how ACEs are inherited when files and directories are created.
.Bl -tag -width "passthrough-x"
.It Sy discard
does not inherit any ACEs.
.It Sy noallow
only inherits inheritable ACEs that specify
.Qq deny
permissions.
.It Sy restricted
default, removes the
.Sy write_acl
and
.Sy write_owner
permissions when the ACE is inherited.
.It Sy passthrough
inherits all inheritable ACEs without any modifications.
.It Sy passthrough-x
same meaning as
.Sy passthrough ,
except that the
.Sy owner@ ,
.Sy group@ ,
and
.Sy everyone@
ACEs inherit the execute permission only if the file creation mode also requests
the execute bit.
.El
.Pp
When the property value is set to
.Sy passthrough ,
files are created with a mode determined by the inheritable ACEs.
If no inheritable ACEs exist that affect the mode, then the mode is set in
accordance to the requested mode from the application.
.It Xo
.Sy aclmode Ns = Ns Sy discard Ns | Ns Sy groupmask Ns | Ns
.Sy passthrough Ns | Ns Sy restricted
.Xc
Controls how an ACL is modified during
.Xr chmod 2
and how inherited ACEs are modified by the file creation mode.
.Bl -tag -width "passthrough"
.It Sy discard
default, deletes all ACEs except for those representing the mode of the file or
directory requested by
```

```
.Xr chmod 2 .
.It Sy groupmask
reduces permissions granted by all
.Sy ALLOW
entries found in the ACL such that they are no greater than the group
permissions specified by the mode.
.It Sy passthrough
indicates that no changes are made to the ACL other than creating or updating
the necessary ACEs to represent the new mode of the file or directory.
.It Sy restricted
causes the
.Xr chmod 2
operation to return an error when used on any file or directory which has a
non-trivial ACL, with entries in addition to those that represent the mode.
.El
.Pp
.Xr chmod 2
is required to change the set user ID, set group ID, or sticky bit on a file or
directory, as they do not have equivalent ACEs.
In order to use
.Xr chmod 2
on a file or directory with a non-trivial ACL when
.Sy aclmode
is set to
.Sy restricted ,
you must first remove all ACEs except for those that represent the current mode.
.It Sy atime Ns = Ns Sy on Ns | Ns Sy off
Controls whether the access time for files is updated when they are read.
Turning this property off avoids producing write traffic when reading files and
can result in significant performance gains, though it might confuse mailers
and other similar utilities.
The default value is
.Sy on .
.It Sy canmount Ns = Ns Sy on Ns | Ns Sy off Ns | Ns Sy noauto
If this property is set to
.Sy off ,
the file system cannot be mounted, and is ignored by
.Nm zfs Cm mount Fl a .
Setting this property to
.Sy off
is similar to setting the
.Sy mountpoint
property to
.Sy none ,
except that the dataset still has a normal
.Sy mountpoint
property, which can be inherited.
Setting this property to
.Sy off
allows datasets to be used solely as a mechanism to inherit properties.
One example of setting
.Sy canmount Ns = Ns Sy off
is to have two datasets with the same
.Sy mountpoint ,
so that the children of both datasets appear in the same directory, but might
have different inherited characteristics.
.Pp
When set to
.Sy noauto ,
a dataset can only be mounted and unmounted explicitly.
The dataset is not mounted automatically when the dataset is created or
imported, nor is it mounted by the
.Nm zfs Cm mount Fl a
command or unmounted by the
.Nm zfs Cm unmount Fl a
command.
```



```
.Pp
This property is not inherited.
.It Xo
.Sy checksum Ns = Ns Sy on Ns | Ns Sy off Ns | Ns Sy fletcher2 Ns | Ns
.Sy fletcher4 Ns | Ns Sy sha256 Ns | Ns Sy noparity Ns | Ns
.Sy sha512 Ns | Ns Sy skein Ns | Ns Sy edonr
.Xc
Controls the checksum used to verify data integrity.
The default value is
.Sy on ,
which automatically selects an appropriate algorithm
.Po currently,
.Sy fletcher4 ,
but this may change in future releases
.Pc .
The value
.Sy off
disables integrity checking on user data.
The value
.Sy noparity
not only disables integrity but also disables maintaining parity for user data.
This setting is used internally by a dump device residing on a RAID-Z pool and
should not be used by any other dataset.
Disabling checksums is
.Sy NOT
a recommended practice.
.Pp
The
.Sy sha512 ,
.Sy skein ,
and
.Sy edonr
checksum algorithms require enabling the appropriate features on the pool.
Please see
.Xr zpool-features 5
for more information on these algorithms.
.Pp
Changing this property affects only newly-written data.
.It Xo
.Sy compression Ns = Ns Sy on Ns | Ns Sy off Ns | Ns Sy gzip Ns | Ns
.Sy gzip- Ns Em N Ns | Ns Sy lz4 Ns | Ns Sy lzjb Ns | Ns Sy zle
.Xc
Controls the compression algorithm used for this dataset.
.Pp
Setting compression to
.Sy on
indicates that the current default compression algorithm should be used.
The default balances compression and decompression speed, with compression ratio
and is expected to work well on a wide variety of workloads.
Unlike all other settings for this property,
.Sy on
does not select a fixed compression type.
As new compression algorithms are added to ZFS and enabled on a pool, the
default compression algorithm may change.
The current default compression algorithm is either
.Sy lzjb
or, if the
.Sy lz4_compress
feature is enabled,
.Sy lz4 .
.Pp
The
.Sy lz4
compression algorithm is a high-performance replacement for the
.Sy lzjb
algorithm.
```

```
1049 It features significantly faster compression and decompression, as well as a
1050 moderately higher compression ratio than
1051 .Sy lzjb ,
1052 but can only be used on pools with the
1053 .Sy lz4_compress
1054 feature set to
1055 .Sy enabled .
1056 See
1057 .Xr zpool-features 5
1058 for details on ZFS feature flags and the
1059 .Sy lz4_compress
1060 feature.
1061 .Pp
1062 The
1063 .Sy lzjb
1064 compression algorithm is optimized for performance while providing decent data
1065 compression.
1066 .Pp
1067 The
1068 .Sy gzip
1069 compression algorithm uses the same compression as the
1070 .Xr gzip 1
1071 command.
1072 You can specify the
1073 .Sy gzip
1074 level by using the value
1075 .Sy gzip- Ns Em N ,
1076 where
1077 .Em N
1078 is an integer from 1
1079 .Pq fastest
1080 to 9
1081 .Pq best compression ratio .
1082 Currently,
1083 .Sy gzip
1084 is equivalent to
1085 .Sy gzip-6
1086 .Po which is also the default for
1087 .Xr gzip 1
1088 .Pc .
1089 .Pp
1090 The
1091 .Sy zle
1092 compression algorithm compresses runs of zeros.
1093 .Pp
1094 This property can also be referred to by its shortened column name
1095 .Sy compress .
1096 Changing this property affects only newly-written data.
1097 .It Sy copies Ns = Ns Sy 1 Ns | Ns Sy 2 Ns | Ns Sy 3
1098 Controls the number of copies of data stored for this dataset.
1099 These copies are in addition to any redundancy provided by the pool, for
1100 example, mirroring or RAID-Z.
1101 The copies are stored on different disks, if possible.
1102 The space used by multiple copies is charged to the associated file and dataset,
1103 changing the
1104 .Sy used
1105 property and counting against quotas and reservations.
1106 .Pp
1107 Changing this property only affects newly-written data.
1108 Therefore, set this property at file system creation time by using the
1109 .Fl o Sy copies Ns = Ns Ar N
1110 option.
1111 .It Sy devices Ns = Ns Sy on Ns | Ns Sy off
1112 Controls whether device nodes can be opened on this file system.
1113 The default value is
1114 .Sy on .
```



```
1115 .It Sy exec Ns = Ns Sy on Ns | Ns Sy off
1116 Controls whether processes can be executed from within this file system.
1117 The default value is
1118 .Sy on .
1119 .It Sy filesystem_limit Ns = Ns Em count Ns | Ns Sy none
1120 Limits the number of filesystems and volumes that can exist under this point in
1121 the dataset tree.
1122 The limit is not enforced if the user is allowed to change the limit.
1123 Setting a
1124 .Sy filesystem_limit
1125 to
1126 .Sy on
1127 a descendent of a filesystem that already has a
1128 .Sy filesystem_limit
1129 does not override the ancestor's
1130 .Sy filesystem_limit ,
1131 but rather imposes an additional limit.
1132 This feature must be enabled to be used
1133 .Po see
1134 .Xr zpool-features 5
1135 .Pc .
1136 .It Sy mountpoint Ns = Ns Pa path Ns | Ns Sy none Ns | Ns Sy legacy
1137 Controls the mount point used for this file system.
1138 See the
1139 .Sx Mount Points
1140 section for more information on how this property is used.
1141 .Pp
1142 When the
1143 .Sy mountpoint
1144 property is changed for a file system, the file system and any children that
1145 inherit the mount point are unmounted.
1146 If the new value is
1147 .Sy legacy ,
1148 then they remain unmounted.
1149 Otherwise, they are automatically remounted in the new location if the property
1150 was previously
1151 .Sy legacy
1152 or
1153 .Sy none ,
1154 or if they were mounted before the property was changed.
1155 In addition, any shared file systems are unshared and shared in the new
1156 location.
1157 .It Sy nbmand Ns = Ns Sy on Ns | Ns Sy off
1158 Controls whether the file system should be mounted with
1159 .Sy nbmand
1160 .Pq Non Blocking mandatory locks .
1161 This is used for SMB clients.
1162 Changes to this property only take effect when the file system is umounted and
1163 remounted.
1164 See
1165 .Xr mount 1M
1166 for more information on
1167 .Sy nbmand
1168 mounts.
1169 .It Sy primarycache Ns = Ns Sy all Ns | Ns Sy none Ns | Ns Sy metadata
1170 Controls what is cached in the primary cache
1171 .Pq ARC .
1172 If this property is set to
1173 .Sy all ,
1174 then both user data and metadata is cached.
1175 If this property is set to
1176 .Sy none ,
1177 then neither user data nor metadata is cached.
1178 If this property is set to
1179 .Sy metadata ,
1180 then only metadata is cached.
```

```
1181 The default value is
1182 .Sy all .
1183 .It Sy quota Ns = Ns Em size Ns | Ns Sy none
1184 Limits the amount of space a dataset and its descendents can consume.
1185 This property enforces a hard limit on the amount of space used.
1186 This includes all space consumed by descendents, including file systems and
1187 snapshots.
1188 Setting a quota on a descendent of a dataset that already has a quota does not
1189 override the ancestor's quota, but rather imposes an additional limit.
1190 .Pp
1191 Quotas cannot be set on volumes, as the
1192 .Sy volsize
1193 property acts as an implicit quota.
1194 .It Sy snapshot_limit Ns = Ns Em count Ns | Ns Sy none
1195 Limits the number of snapshots that can be created on a dataset and its
1196 descendents.
1197 Setting a
1198 .Sy snapshot_limit
1199 on a descendent of a dataset that already has a
1200 .Sy snapshot_limit
1201 does not override the ancestor's
1202 .Sy snapshot_limit ,
1203 but rather imposes an additional limit.
1204 The limit is not enforced if the user is allowed to change the limit.
1205 For example, this means that recursive snapshots taken from the global zone are
1206 counted against each delegated dataset within a zone.
1207 This feature must be enabled to be used
1208 .Po see
1209 .Xr zpool-features 5
1210 .Pc .
1211 .It Sy userquota@ Ns Em user Ns = Ns Em size Ns | Ns Sy none
1212 Limits the amount of space consumed by the specified user.
1213 User space consumption is identified by the
1214 .Sy userspace@ Ns Em user
1215 property.
1216 .Pp
1217 Enforcement of user quotas may be delayed by several seconds.
1218 This delay means that a user might exceed their quota before the system notices
1219 that they are over quota and begins to refuse additional writes with the
1220 .Er EDQUOT
1221 error message.
1222 See the
1223 .Nm zfs Cm userspace
1224 subcommand for more information.
1225 .Pp
1226 Unprivileged users can only access their own groups' space usage.
1227 The root user, or a user who has been granted the
1228 .Sy userquota
1229 privilege with
1230 .Nm zfs Cm allow ,
1231 can get and set everyone's quota.
1232 .Pp
1233 This property is not available on volumes, on file systems before version 4, or
1234 on pools before version 15.
1235 The
1236 .Sy userquota@ Ns Em ...
1237 properties are not displayed by
1238 .Nm zfs Cm get Sy all .
1239 The user's name must be appended after the
1240 .Sy @
1241 symbol, using one of the following forms:
1242 .Bl -bullet
1243 .It
1244 .Em POSIX name
1245 .Po for example,
1246 .Sy joe
```



```
1247 .Pc
1248 .It
1249 .Em POSIX numeric ID
1250 .Po for example,
1251 .Sy 789
1252 .Pc
1253 .It
1254 .Em SID name
1255 .Po for example,
1256 .Sy joe.smith@mydomain
1257 .Pc
1258 .It
1259 .Em SID numeric ID
1260 .Po for example,
1261 .Sy S-1-123-456-789
1262 .Pc
1263 .El
1264 .It Sy groupquota@ Ns Em group Ns = Ns Em size Ns | Ns Sy none
1265 Limits the amount of space consumed by the specified group.
1266 Group space consumption is identified by the
1267 .Sy groupused@ Ns Em group
1268 property.
1269 .Pp
1270 Unprivileged users can access only their own groups' space usage.
1271 The root user, or a user who has been granted the
1272 .Sy groupquota
1273 privilege with
1274 .Nm zfs Cm allow ,
1275 can get and set all groups' quotas.
1276 .It Sy readonly Ns = Ns Sy on Ns | Ns Sy off
1277 Controls whether this dataset can be modified.
1278 The default value is
1279 .Sy off .
1280 .Pp
1281 This property can also be referred to by its shortened column name,
1282 .Sy rdonly .
1283 .It Sy recordsize Ns = Ns Em size
1284 Specifies a suggested block size for files in the file system.
1285 This property is designed solely for use with database workloads that access
1286 files in fixed-size records.
1287 ZFS automatically tunes block sizes according to internal algorithms optimized
1288 for typical access patterns.
1289 .Pp
1290 For databases that create very large files but access them in small random
1291 chunks, these algorithms may be suboptimal.
1292 Specifying a
1293 .Sy recordsize
1294 greater than or equal to the record size of the database can result in
1295 significant performance gains.
1296 Use of this property for general purpose file systems is strongly discouraged,
1297 and may adversely affect performance.
1298 .Pp
1299 The size specified must be a power of two greater than or equal to 512 and less
1300 than or equal to 128 Kbytes.
1301 If the
1302 .Sy large_blocks
1303 feature is enabled on the pool, the size may be up to 1 Mbyte.
1304 See
1305 .Xr zpool-features 5
1306 for details on ZFS feature flags.
1307 .Pp
1308 Changing the file system's
1309 .Sy recordsize
1310 affects only files created afterward; existing files are unaffected.
1311 .Pp
1312 This property can also be referred to by its shortened column name,
```

.Sy recsize .
.It Sy redundant_metadata Ns = Ns Sy all Ns | Ns Sy most
Controls what types of metadata are stored redundantly.
ZFS stores an extra copy of metadata, so that if a single block is corrupted,
the amount of user data lost is limited.
This extra copy is in addition to any redundancy provided at the pool level
.Pq e.g. by mirroring or RAID-Z ,
and is in addition to an extra copy specified by the
.Sy copies
property
.Pq up to a total of 3 copies .
For example if the pool is mirrored,
.Sy copies Ns = Ns 2 ,
and
.Sy redundant_metadata Ns = Ns Sy most ,
then ZFS stores 6 copies of most metadata, and 4 copies of data and some
metadata.
.Pp
When set to
.Sy all ,
ZFS stores an extra copy of all metadata.
If a single on-disk block is corrupt, at worst a single block of user data
.Po which is
.Sy recordsize
bytes long
.Pc
can be lost.
.Pp
When set to
.Sy most ,
ZFS stores an extra copy of most types of metadata.
This can improve performance of random writes, because less metadata must be
written.
In practice, at worst about 100 blocks
.Po of
.Sy recordsize
bytes each
.Pc
of user data can be lost if a single on-disk block is corrupt.
The exact behavior of which metadata blocks are stored redundantly may change in
future releases.
.Pp
The default value is
.Sy all .
.It Sy refquota Ns = Ns Em size Ns | Ns Sy none
Limits the amount of space a dataset can consume.
This property enforces a hard limit on the amount of space used.
This hard limit does not include space used by descendents, including file
systems and snapshots.
.It Sy refreservation Ns = Ns Em size Ns | Ns Sy none Ns | Ns Sy auto
The minimum amount of space guaranteed to a dataset, not including its
descendents.
When the amount of space used is below this value, the dataset is treated as if
it were taking up the amount of space specified by
.Sy refreservation .
The
.Sy refreservation
reservation is accounted for in the parent datasets' space used, and counts
against the parent datasets' quotas and reservations.
.Pp
If
.Sy refreservation
is set, a snapshot is only allowed if there is enough free pool space outside of
this reservation to accommodate the current number of
.Qq referenced
bytes in the dataset.

.Pp
If
.Sy refreservation
is set to
.Sy auto ,
a volume is thick provisioned
.Po or
.Qq not sparse
.Pc .
.Sy refreservation Ns = Ns Sy auto
is only supported on volumes.
See
.Sy volsize
in the
.Sx Native Properties
section for more information about sparse volumes.
.Pp
This property can also be referred to by its shortened column name,
.Sy refreserv .
.It Sy reservation Ns = Ns Em size Ns | Ns Sy none
The minimum amount of space guaranteed to a dataset and its descendants.
When the amount of space used is below this value, the dataset is treated as if
it were taking up the amount of space specified by its reservation.
Reservations are accounted for in the parent datasets' space used, and count
against the parent datasets' quotas and reservations.
.Pp
This property can also be referred to by its shortened column name,
.Sy reserv .
.It Sy secondarycache Ns = Ns Sy all Ns | Ns Sy none Ns | Ns Sy metadata
Controls what is cached in the secondary cache
.Pq L2ARC .
If this property is set to
.Sy all ,
then both user data and metadata is cached.
If this property is set to
.Sy none ,
then neither user data nor metadata is cached.
If this property is set to
.Sy metadata ,
then only metadata is cached.
The default value is
.Sy all .
.It Sy setuid Ns = Ns Sy on Ns | Ns Sy off
Controls whether the setuid bit is respected for the file system.
The default value is
.Sy on .
.It Sy sharesmb Ns = Ns Sy on Ns | Ns Sy off Ns | Ns Em opts
Controls whether the file system is shared via SMB, and what options are to be
used.
A file system with the
.Sy sharesmb
property set to
.Sy off
is managed through traditional tools such as
.Xr sharemgr 1M .
Otherwise, the file system is automatically shared and unshared with the
.Nm zfs Cm share
and
.Nm zfs Cm unshare
commands.
If the property is set to
.Sy on ,
the
.Xr sharemgr 1M
command is invoked with no options.
Otherwise, the

```
.Xr sharemgr 1M
command is invoked with options equivalent to the contents of this property.
.Pp
Because SMB shares requires a resource name, a unique resource name is
constructed from the dataset name.
The constructed name is a copy of the dataset name except that the characters in
the dataset name, which would be invalid in the resource name, are replaced with
underscore
.Pq Sy _
characters.
A pseudo property
.Qq name
is also supported that allows you to replace the data set name with a specified
name.
The specified name is then used to replace the prefix dataset in the case of
inheritance.
For example, if the dataset
.Em data/home/john
is set to
.Sy name Ns = Ns Sy john ,
then
.Em data/home/john
has a resource name of
.Sy john .
If a child dataset
.Em data/home/john/backups
is shared, it has a resource name of
.Sy john_backups .
.Pp
When SMB shares are created, the SMB share name appears as an entry in the
.Pa .zfs/shares
directory.
You can use the
.Nm ls
or
.Nm chmod
command to display the share-level ACLs on the entries in this directory.
.Pp
When the
.Sy sharesmb
property is changed for a dataset, the dataset and any children inheriting the
property are re-shared with the new options, only if the property was previously
set to
.Sy off ,
or if they were shared before the property was changed.
If the new property is set to
.Sy off ,
the file systems are unshared.
.It Sy sharenfs Ns = Ns Sy on Ns | Ns Sy off Ns | Ns Em opts
Controls whether the file system is shared via NFS, and what options are to be
used.
A file system with a
.Sy sharenfs
property of
.Sy off
is managed through traditional tools such as
.Xr share 1M ,
.Xr unshare 1M ,
and
.Xr dfstab 4 .
Otherwise, the file system is automatically shared and unshared with the
.Nm zfs Cm share
and
.Nm zfs Cm unshare
commands.
If the property is set to
```



```
.Sy on ,
.Xr share 1M
command is invoked with no options.
Otherwise, the
.Xr share 1M
command is invoked with options equivalent to the contents of this property.
.Pp
When the
.Sy sharenfs
property is changed for a dataset, the dataset and any children inheriting the
property are re-shared with the new options, only if the property was previously
.Sy off ,
or if they were shared before the property was changed.
If the new property is
.Sy off ,
the file systems are unshared.
.It Sy logbias Ns = Ns Sy latency Ns | Ns Sy throughput
Provide a hint to ZFS about handling of synchronous requests in this dataset.
If
.Sy logbias
is set to
.Sy latency
.Pq the default ,
ZFS will use pool log devices
.Pq if configured
to handle the requests at low latency.
If
.Sy logbias
is set to
.Sy throughput ,
ZFS will not use configured pool log devices.
ZFS will instead optimize synchronous operations for global pool throughput and
efficient use of resources.
.It Sy snapdir Ns = Ns Sy hidden Ns | Ns Sy visible
Controls whether the
.Pa .zfs
directory is hidden or visible in the root of the file system as discussed in
the
.Sx Snapshots
section.
The default value is
.Sy hidden .
.It Sy sync Ns = Ns Sy standard Ns | Ns Sy always Ns | Ns Sy disabled
Controls the behavior of synchronous requests
.Pq e.g. fsync, O_DSYNC .
.Sy standard
is the
.Tn POSIX
specified behavior of ensuring all synchronous requests are written to stable
storage and all devices are flushed to ensure data is not cached by device
controllers
.Pq this is the default .
.Sy always
causes every file system transaction to be written and flushed before its
system call returns.
This has a large performance penalty.
.Sy disabled
disables synchronous requests.
File system transactions are only committed to stable storage periodically.
This option will give the highest performance.
However, it is very dangerous as ZFS would be ignoring the synchronous
transaction demands of applications such as databases or NFS.
Administrators should only use this option when the risks are understood.
.It Sy version Ns = Ns Em N Ns | Ns Sy current
The on-disk version of this file system, which is independent of the pool
version.
```

```
1577 This property can only be set to later supported versions.
1578 See the
1579 .Nm zfs Cm upgrade
1580 command.
1581 .It Sy volsize Ns = Ns Em size
1582 For volumes, specifies the logical size of the volume.
1583 By default, creating a volume establishes a reservation of equal size.
1584 For storage pools with a version number of 9 or higher, a
1585 .Sy refreservation
1586 is set instead.
1587 Any changes to
1588 .Sy volsize
1589 are reflected in an equivalent change to the reservation
1590 .Po or
1591 .Sy refreservation
1592 .Pc .
1593 The
1594 .Sy volsize
1595 can only be set to a multiple of
1596 .Sy volblocksize ,
1597 and cannot be zero.
1598 .Pp
1599 The reservation is kept equal to the volume's logical size to prevent unexpected
1600 behavior for consumers.
1601 Without the reservation, the volume could run out of space, resulting in
1602 undefined behavior or data corruption, depending on how the volume is used.
1603 These effects can also occur when the volume size is changed while it is in use
1604 .Pq particularly when shrinking the size .
1605 Extreme care should be used when adjusting the volume size.
1606 .Pp
1607 Though not recommended, a
1608 .Qq sparse volume
1609 .Po also known as
1610 .Qq thin provisioned
1611 .Pc
1612 can be created by specifying the
1613 .Fl s
1614 option to the
1615 .Nm zfs Cm create Fl V
1616 command, or by changing the value of the
1617 .Sy refreservation
1618 property
1619 .Po or
1620 .Sy reservation
1621 property on pool version 8 or earlier
1622 .Pc
1623 after the volume has been created.
1624 A
1625 .Qq sparse volume
1626 is a volume where the value of
1627 .Sy refreservation
1628 is less than the size of the volume plus the space required to store its
1629 metadata.
1630 Consequently, writes to a sparse volume can fail with
1631 .Er ENOSPC
1632 when the pool is low on space.
1633 For a sparse volume, changes to
1634 .Sy volsize
1635 are not reflected in the
1636 .Sy refreservation.
1637 A volume that is not sparse is said to be
1638 .Qq thick provisioned .
1639 A sparse volume can become thick provisioned by setting
1640 .Sy refreservation
1641 to
1642 .Sy auto .
```



```
1643 .It Sy vscan Ns = Ns Sy on Ns | Ns Sy off
1644 Controls whether regular files should be scanned for viruses when a file is
1645 opened and closed.
1646 In addition to enabling this property, the virus scan service must also be
1647 enabled for virus scanning to occur.
1648 The default value is
1649 .Sy off .
1650 .It Sy xattr Ns = Ns Sy on Ns | Ns Sy off
1651 Controls whether extended attributes are enabled for this file system.
1652 The default value is
1653 .Sy on .
1654 .It Sy zoned Ns = Ns Sy on Ns | Ns Sy off
1655 Controls whether the dataset is managed from a non-global zone.
1656 See the
1657 .Sx Zones
1658 section for more information.
1659 The default value is
1660 .Sy off .
1661 .El
1662 .Pp
1663 The following three properties cannot be changed after the file system is
1664 created, and therefore, should be set when the file system is created.
1665 If the properties are not set with the
1666 .Nm zfs Cm create
1667 or
1668 .Nm zpool Cm create
1669 commands, these properties are inherited from the parent dataset.
1670 If the parent dataset lacks these properties due to having been created prior to
1671 these features being supported, the new file system will have the default values
1672 for these properties.
1673 .Bl -tag -width ""
1674 .It Xo
1675 .Sy casesensitivity Ns = Ns Sy sensitive Ns | Ns
1676 .Sy insensitive Ns | Ns Sy mixed
1677 .Xc
1678 Indicates whether the file name matching algorithm used by the file system
1679 should be case-sensitive, case-insensitive, or allow a combination of both
1680 styles of matching.
1681 The default value for the
1682 .Sy casesensitivity
1683 property is
1684 .Sy sensitive .
1685 Traditionally,
1686 .Ux
1687 and
1688 .Tn POSIX
1689 file systems have case-sensitive file names.
1690 .Pp
1691 The
1692 .Sy mixed
1693 value for the
1694 .Sy casesensitivity
1695 property indicates that the file system can support requests for both
1696 case-sensitive and case-insensitive matching behavior.
1697 Currently, case-insensitive matching behavior on a file system that supports
1698 mixed behavior is limited to the SMB server product.
1699 For more information about the
1700 .Sy mixed
1701 value behavior, see the "ZFS Administration Guide".
1702 .It Xo
1703 .Sy normalization Ns = Ns Sy none Ns | Ns Sy formC Ns | Ns
1704 .Sy formD Ns | Ns Sy formKC Ns | Ns Sy formKD
1705 .Xc
1706 Indicates whether the file system should perform a
1707 .Sy unicode
1708 normalization of file names whenever two file names are compared, and which
```

normalization algorithm should be used.
File names are always stored unmodified, names are normalized as part of any
comparison process.
If this property is set to a legal value other than
.Sy none ,
and the
.Sy utf8only
property was left unspecified, the
.Sy utf8only
property is automatically set to
.Sy on .
The default value of the
.Sy normalization
property is
.Sy none .
This property cannot be changed after the file system is created.
.It Sy utf8only Ns = Ns Sy on Ns | Ns Sy off
Indicates whether the file system should reject file names that include
characters that are not present in the
.Sy UTF-8
character code set.
If this property is explicitly set to
.Sy off ,
the normalization property must either not be explicitly set or be set to
.Sy none .
The default value for the
.Sy utf8only
property is
.Sy off .
This property cannot be changed after the file system is created.
.El
.Pp
The
.Sy casesensitivity ,
.Sy normalization ,
and
.Sy utf8only
properties are also new permissions that can be assigned to non-privileged users
by using the ZFS delegated administration feature.
.Ss "Temporary Mount Point Properties"
When a file system is mounted, either through
.Xr mount 1M
for legacy mounts or the
.Nm zfs Cm mount
command for normal file systems, its mount options are set according to its
properties.
The correlation between properties and mount options is as follows:
.Bd -literal
    PROPERTY   MOUNT OPTION
    devices   devices/nodevices
    exec   exec/noexec
    readonly   ro/rw
    setuid   setuid/nosetuid
    xattr   xattr/noxattr
.Ed
.Pp
In addition, these options can be set on a per-mount basis using the
.Fl o
option, without affecting the property that is stored on disk.
The values specified on the command line override the values stored in the
dataset.
The
.Sy nosuid
option is an alias for
.Sy nodevices Ns \&, Ns Sy nosetuid .
These properties are reported as

.Qq temporary
by the
.Nm zfs Cm get
command.
If the properties are changed while the dataset is mounted, the new setting
overrides any temporary settings.
.Ss "User Properties"
In addition to the standard native properties, ZFS supports arbitrary user
properties.
User properties have no effect on ZFS behavior, but applications or
administrators can use them to annotate datasets
.Pq file systems, volumes, and snapshots .
.Pp
User property names must contain a colon
.Pq Qq Sy \&:
character to distinguish them from native properties.
They may contain lowercase letters, numbers, and the following punctuation
characters: colon
.Pq Qq Sy \&: ,
dash
.Pq Qq Sy - ,
period
.Pq Qq Sy \&. ,
and underscore
.Pq Qq Sy _ .
The expected convention is that the property name is divided into two portions
such as
.Em module Ns \&: Ns Em property ,
but this namespace is not enforced by ZFS.
User property names can be at most 256 characters, and cannot begin with a dash
.Pq Qq Sy - .
.Pp
When making programmatic use of user properties, it is strongly suggested to use
a reversed
.Sy DNS
domain name for the
.Em module
component of property names to reduce the chance that two
independently-developed packages use the same property name for different
purposes.
.Pp
The values of user properties are arbitrary strings, are always inherited, and
are never validated.
All of the commands that operate on properties
.Po Nm zfs Cm list ,
.Nm zfs Cm get ,
.Nm zfs Cm set ,
and so forth
.Pc
can be used to manipulate both native properties and user properties.
Use the
.Nm zfs Cm inherit
command to clear a user property.
If the property is not defined in any parent dataset, it is removed entirely.
Property values are limited to 8192 bytes.
.Ss ZFS Volumes as Swap or Dump Devices
During an initial installation a swap device and dump device are created on ZFS
volumes in the ZFS root pool.
By default, the swap area size is based on 1/2 the size of physical memory up to
2 Gbytes.
The size of the dump device depends on the kernel’s requirements at installation
time.
Separate ZFS volumes must be used for the swap area and dump devices.
Do not swap to a file on a ZFS file system.
A ZFS swap file configuration is not supported.
.Pp

```
1841 If you need to change your swap area or dump device after the system is
1842 installed or upgraded, use the
1843 .Xr swap 1M
1844 and
1845 .Xr dumpadm 1M
1846 commands.
1847 .Sh SUBCOMMANDS
1848 All subcommands that modify state are logged persistently to the pool in their
1849 original form.
1850 .Bl -tag -width ""
1851 .It Nm Fl \?
1852 Displays a help message.
1853 .It Xo
1854 .Nm
1855 .Cm create
1856 .Op Fl p
1857 .Oo Fl o Ar property Ns = Ns Ar value Oc Ns ...
1858 .Ar filesystem
1859 .Xc
1860 Creates a new ZFS file system.
1861 The file system is automatically mounted according to the
1862 .Sy mountpoint
1863 property inherited from the parent.
1864 .Bl -tag -width "-o"
1865 .It Fl o Ar property Ns = Ns Ar value
1866 Sets the specified property as if the command
1867 .Nm zfs Cm set Ar property Ns = Ns Ar value
1868 was invoked at the same time the dataset was created.
1869 Any editable ZFS property can also be set at creation time.
1870 Multiple
1871 .Fl o
1872 options can be specified.
1873 An error results if the same property is specified in multiple
1874 .Fl o
1875 options.
1876 .It Fl p
1877 Creates all the non-existing parent datasets.
1878 Datasets created in this manner are automatically mounted according to the
1879 .Sy mountpoint
1880 property inherited from their parent.
1881 Any property specified on the command line using the
1882 .Fl o
1883 option is ignored.
1884 If the target filesystem already exists, the operation completes successfully.
1885 .El
1886 .It Xo
1887 .Nm
1888 .Cm create
1889 .Op Fl ps
1890 .Op Fl b Ar blocksize
1891 .Oo Fl o Ar property Ns = Ns Ar value Oc Ns ...
1892 .Fl V Ar size Ar volume
1893 .Xc
1894 Creates a volume of the given size.
1895 The volume is exported as a block device in
1896 .Pa /dev/zvol/{dsk,rdsk}/path ,
1897 where
1898 .Em path
1899 is the name of the volume in the ZFS namespace.
1900 The size represents the logical size as exported by the device.
1901 By default, a reservation of equal size is created.
1902 .Pp
1903 .Ar size
1904 is automatically rounded up to the nearest 128 Kbytes to ensure that the volume
1905 has an integral number of blocks regardless of
1906 .Sy blocksize .
```



```
1907 .Bl -tag -width "-b"
1908 .It Fl b Ar blocksize
1909 Equivalent to
1910 .Fl o Sy volblocksize Ns = Ns Ar blocksize .
1911 If this option is specified in conjunction with
1912 .Fl o Sy volblocksize ,
1913 the resulting behavior is undefined.
1914 .It Fl o Ar property Ns = Ns Ar value
1915 Sets the specified property as if the
1916 .Nm zfs Cm set Ar property Ns = Ns Ar value
1917 command was invoked at the same time the dataset was created.
1918 Any editable ZFS property can also be set at creation time.
1919 Multiple
1920 .Fl o
1921 options can be specified.
1922 An error results if the same property is specified in multiple
1923 .Fl o
1924 options.
1925 .It Fl p
1926 Creates all the non-existing parent datasets.
1927 Datasets created in this manner are automatically mounted according to the
1928 .Sy mountpoint
1929 property inherited from their parent.
1930 Any property specified on the command line using the
1931 .Fl o
1932 option is ignored.
1933 If the target filesystem already exists, the operation completes successfully.
1934 .It Fl s
1935 Creates a sparse volume with no reservation.
1936 See
1937 .Sy volsize
1938 in the
1939 .Sx Native Properties
1940 section for more information about sparse volumes.
1941 .El
1942 .It Xo
1943 .Nm
1944 .Cm destroy
1945 .Op Fl Rfnprv
1946 .Ar filesystem Ns | Ns Ar volume
1947 .Xc
1948 Destroys the given dataset.
1949 By default, the command unshares any file systems that are currently shared,
1950 unmounts any file systems that are currently mounted, and refuses to destroy a
1951 dataset that has active dependents
1952 .Pq children or clones .
1953 .Bl -tag -width "-R"
1954 .It Fl R
1955 Recursively destroy all dependents, including cloned file systems outside the
1956 target hierarchy.
1957 .It Fl f
1958 Force an unmount of any file systems using the
1959 .Nm unmount Fl f
1960 command.
1961 This option has no effect on non-file systems or unmounted file systems.
1962 .It Fl n
1963 Do a dry-run
1964 .Pq Qq No-op
1965 deletion.
1966 No data will be deleted.
1967 This is useful in conjunction with the
1968 .Fl v
1969 or
1970 .Fl p
1971 flags to determine what data would be deleted.
1972 .It Fl p
```

```
1973 Print machine-parsable verbose information about the deleted data.
1974 .It Fl r
1975 Recursively destroy all children.
1976 .It Fl v
1977 Print verbose information about the deleted data.
1978 .El
1979 .Pp
1980 Extreme care should be taken when applying either the
1981 .Fl r
1982 or the
1983 .Fl R
1984 options, as they can destroy large portions of a pool and cause unexpected
1985 behavior for mounted file systems in use.
1986 .It Xo
1987 .Nm
1988 .Cm destroy
1989 .Op Fl Rdnprv
1990 .Ar filesystem Ns | Ns Ar volume Ns @ Ns Ar snap Ns
1991 .Oo % Ns Ar snap Ns Oo , Ns Ar snap Ns Oo % Ns Ar snap Oc Oc Oc Ns ...
1992 .Xc
1993 The given snapshots are destroyed immediately if and only if the
1994 .Nm zfs Cm destroy
1995 command without the
1996 .Fl d
1997 option would have destroyed it.
1998 Such immediate destruction would occur, for example, if the snapshot had no
1999 clones and the user-initiated reference count were zero.
2000 .Pp
2001 If a snapshot does not qualify for immediate destruction, it is marked for
2002 deferred deletion.
2003 In this state, it exists as a usable, visible snapshot until both of the
2004 preconditions listed above are met, at which point it is destroyed.
2005 .Pp
2006 An inclusive range of snapshots may be specified by separating the first and
2007 last snapshots with a percent sign.
2008 The first and/or last snapshots may be left blank, in which case the
2009 filesystem's oldest or newest snapshot will be implied.
2010 .Pp
2011 Multiple snapshots
2012 .Pq or ranges of snapshots
2013 of the same filesystem or volume may be specified in a comma-separated list of
2014 snapshots.
2015 Only the snapshot's short name
2016 .Po the part after the
2017 .Sy @
2018 .Pc
2019 should be specified when using a range or comma-separated list to identify
2020 multiple snapshots.
2021 .Bl -tag -width "-R"
2022 .It Fl R
2023 Recursively destroy all clones of these snapshots, including the clones,
2024 snapshots, and children.
2025 If this flag is specified, the
2026 .Fl d
2027 flag will have no effect.
2028 .It Fl d
2029 Defer snapshot deletion.
2030 .It Fl n
2031 Do a dry-run
2032 .Pq Qq No-op
2033 deletion.
2034 No data will be deleted.
2035 This is useful in conjunction with the
2036 .Fl p
2037 or
2038 .Fl v
```



```
2039 flags to determine what data would be deleted.
2040 .It Fl p
2041 Print machine-parsable verbose information about the deleted data.
2042 .It Fl r
2043 Destroy
2044 .Pq or mark for deferred deletion
2045 all snapshots with this name in descendent file systems.
2046 .It Fl v
2047 Print verbose information about the deleted data.
2048 .Pp
2049 Extreme care should be taken when applying either the
2050 .Fl r
2051 or the
2052 .Fl R
2053 options, as they can destroy large portions of a pool and cause unexpected
2054 behavior for mounted file systems in use.
2055 .El
2056 .It Xo
2057 .Nm
2058 .Cm destroy
2059 .Ar filesystem Ns | Ns Ar volume Ns # Ns Ar bookmark
2060 .Xc
2061 The given bookmark is destroyed.
2062 .It Xo
2063 .Nm
2064 .Cm snapshot
2065 .Op Fl r
2066 .Oo Fl o Ar property Ns = Ns value Oc Ns ...
2067 .Ar filesystem Ns @ Ns Ar snapname Ns | Ns Ar volume Ns @ Ns Ar snapname Ns ...
2068 .Xc
2069 Creates snapshots with the given names.
2070 All previous modifications by successful system calls to the file system are
2071 part of the snapshots.
2072 Snapshots are taken atomically, so that all snapshots correspond to the same
2073 moment in time.
2074 See the
2075 .Sx Snapshots
2076 section for details.
2077 .Bl -tag -width "-o"
2078 .It Fl o Ar property Ns = Ns Ar value
2079 Sets the specified property; see
2080 .Nm zfs Cm create
2081 for details.
2082 .It Fl r
2083 Recursively create snapshots of all descendent datasets
2084 .El
2085 .It Xo
2086 .Nm
2087 .Cm rollback
2088 .Op Fl Rfr
2089 .Ar snapshot
2090 .Xc
2091 Roll back the given dataset to a previous snapshot.
2092 When a dataset is rolled back, all data that has changed since the snapshot is
2093 discarded, and the dataset reverts to the state at the time of the snapshot.
2094 By default, the command refuses to roll back to a snapshot other than the most
2095 recent one.
2096 In order to do so, all intermediate snapshots and bookmarks must be destroyed by
2097 specifying the
2098 .Fl r
2099 option.
2100 .Pp
2101 The
2102 .Fl rR
2103 options do not recursively destroy the child snapshots of a recursive snapshot.
2104 Only direct snapshots of the specified filesystem are destroyed by either of
```

```
2105 these options.
2106 To completely roll back a recursive snapshot, you must rollback the individual
2107 child snapshots.
2108 .Bl -tag -width "-R"
2109 .It Fl R
2110 Destroy any more recent snapshots and bookmarks, as well as any clones of those
2111 snapshots.
2112 .It Fl f
2113 Used with the
2114 .Fl R
2115 option to force an unmount of any clone file systems that are to be destroyed.
2116 .It Fl r
2117 Destroy any snapshots and bookmarks more recent than the one specified.
2118 .El
2119 .It Xo
2120 .Nm
2121 .Cm clone
2122 .Op Fl p
2123 .Oo Fl o Ar property Ns = Ns Ar value Oc Ns ...
2124 .Ar snapshot Ar filesystem Ns | Ns Ar volume
2125 .Xc
2126 Creates a clone of the given snapshot.
2127 See the
2128 .Sx Clones
2129 section for details.
2130 The target dataset can be located anywhere in the ZFS hierarchy, and is created
2131 as the same type as the original.
2132 .Bl -tag -width "-o"
2133 .It Fl o Ar property Ns = Ns Ar value
2134 Sets the specified property; see
2135 .Nm zfs Cm create
2136 for details.
2137 .It Fl p
2138 Creates all the non-existing parent datasets.
2139 Datasets created in this manner are automatically mounted according to the
2140 .Sy mountpoint
2141 property inherited from their parent.
2142 If the target filesystem or volume already exists, the operation completes
2143 successfully.
2144 .El
2145 .It Xo
2146 .Nm
2147 .Cm promote
2148 .Ar clone-filesystem
2149 .Xc
2150 Promotes a clone file system to no longer be dependent on its
2151 .Qq origin
2152 snapshot.
2153 This makes it possible to destroy the file system that the clone was created
2154 from.
2155 The clone parent-child dependency relationship is reversed, so that the origin
2156 file system becomes a clone of the specified file system.
2157 .Pp
2158 The snapshot that was cloned, and any snapshots previous to this snapshot, are
2159 now owned by the promoted clone.
2160 The space they use moves from the origin file system to the promoted clone, so
2161 enough space must be available to accommodate these snapshots.
2162 No new space is consumed by this operation, but the space accounting is
2163 adjusted.
2164 The promoted clone must not have any conflicting snapshot names of its own.
2165 The
2166 .Cm rename
2167 subcommand can be used to rename any conflicting snapshots.
2168 .It Xo
2169 .Nm
2170 .Cm rename
```



```
2171 .Op Fl f
2172 .Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot
2173 .Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot
2174 .Xc
2175 .It Xo
2176 .Nm
2177 .Cm rename
2178 .Op Fl fp
2179 .Ar filesystem Ns | Ns Ar volume
2180 .Ar filesystem Ns | Ns Ar volume
2181 .Xc
2182 Renames the given dataset.
2183 The new target can be located anywhere in the ZFS hierarchy, with the exception
2184 of snapshots.
2185 Snapshots can only be renamed within the parent file system or volume.
2186 When renaming a snapshot, the parent file system of the snapshot does not need
2187 to be specified as part of the second argument.
2188 Renamed file systems can inherit new mount points, in which case they are
2189 unmounted and remounted at the new mount point.
2190 .Bl -tag -width "-a"
2191 .It Fl f
2192 Force unmount any filesystems that need to be unmounted in the process.
2193 .It Fl p
2194 Creates all the nonexistent parent datasets.
2195 Datasets created in this manner are automatically mounted according to the
2196 .Sy mountpoint
2197 property inherited from their parent.
2198 .El
2199 .It Xo
2200 .Nm
2201 .Cm rename
2202 .Fl r
2203 .Ar snapshot Ar snapshot
2204 .Xc
2205 Recursively rename the snapshots of all descendent datasets.
2206 Snapshots are the only dataset that can be renamed recursively.
2207 .It Xo
2208 .Nm
2209 .Cm list
2210 .Op Fl r Ns | Ns Fl d Ar depth
2211 .Op Fl Hp
2212 .Oo Fl o Ar property Ns Oo , Ns Ar property Oc Ns ... Oc
2213 .Oo Fl s Ar property Oc Ns ...
2214 .Oo Fl S Ar property Oc Ns ...
2215 .Oo Fl t Ar type Ns Oo , Ns Ar type Oc Ns ... Oc
2216 .Oo Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot Oc Ns ...
2217 .Xc
2218 Lists the property information for the given datasets in tabular form.
2219 If specified, you can list property information by the absolute pathname or the
2220 relative pathname.
2221 By default, all file systems and volumes are displayed.
2222 Snapshots are displayed if the
2223 .Sy listsnaps
2224 property is
2225 .Sy on
2226 .Po the default is
2227 .Sy off
2228 .Pc .
2229 The following fields are displayed,
2230 .Sy name Ns \&, Ns Sy used Ns \&, Ns Sy available Ns \&, Ns Sy referenced Ns \&,
2231 .Sy mountpoint .
2232 .Bl -tag -width "-H"
2233 .It Fl H
2234 Used for scripting mode.
2235 Do not print headers and separate fields by a single tab instead of arbitrary
2236 white space.
```

```
2237 .It Fl S Ar property
2238 Same as the
2239 .Fl s
2240 option, but sorts by property in descending order.
2241 .It Fl d Ar depth
2242 Recursively display any children of the dataset, limiting the recursion to
2243 .Ar depth .
2244 A
2245 .Ar depth
2246 of
2247 .Sy 1
2248 will display only the dataset and its direct children.
2249 .It Fl o Ar property
2250 A comma-separated list of properties to display.
2251 The property must be:
2252 .Bl -bullet
2253 .It
2254 One of the properties described in the
2255 .Sx Native Properties
2256 section
2257 .It
2258 A user property
2259 .It
2260 The value
2261 .Sy name
2262 to display the dataset name
2263 .It
2264 The value
2265 .Sy space
2266 to display space usage properties on file systems and volumes.
2267 This is a shortcut for specifying
2268 .Fl o Sy name Ns \&, Ns Sy avail Ns \&, Ns Sy used Ns \&, Ns Sy usedsnap Ns \&,
2269 .Sy usedds Ns \&, Ns Sy usedrefreserv Ns \&, Ns Sy usedchild Fl t
2270 .Sy filesystem Ns \&, Ns Sy volume
2271 syntax.
2272 .El
2273 .It Fl p
2274 Display numbers in parsable
2275 .Pq exact
2276 values.
2277 .It Fl r
2278 Recursively display any children of the dataset on the command line.
2279 .It Fl s Ar property
2280 A property for sorting the output by column in ascending order based on the
2281 value of the property.
2282 The property must be one of the properties described in the
2283 .Sx Properties
2284 section, or the special value
2285 .Sy name
2286 to sort by the dataset name.
2287 Multiple properties can be specified at one time using multiple
2288 .Fl s
2289 property options.
2290 Multiple
2291 .Fl s
2292 options are evaluated from left to right in decreasing order of importance.
2293 The following is a list of sorting criteria:
2294 .Bl -bullet
2295 .It
2296 Numeric types sort in numeric order.
2297 .It
2298 String types sort in alphabetical order.
2299 .It
2300 Types inappropriate for a row sort that row to the literal bottom, regardless of
2301 the specified ordering.
2302 .El
```



```
2303 .Pp
2304 If no sorting options are specified the existing behavior of
2305 .Nm zfs Cm list
2306 is preserved.
2307 .It Fl t Ar type
2308 A comma-separated list of types to display, where
2309 .Ar type
2310 is one of
2311 .Sy filesystem ,
2312 .Sy snapshot ,
2313 .Sy volume ,
2314 .Sy bookmark ,
2315 or
2316 .Sy all .
2317 For example, specifying
2318 .Fl t Sy snapshot
2319 displays only snapshots.
2320 .El
2321 .It Xo
2322 .Nm
2323 .Cm set
2324 .Ar property Ns = Ns Ar value Oo Ar property Ns = Ns Ar value Oc Ns ...
2325 .Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot Ns ...
2326 .Xc
2327 Sets the property or list of properties to the given value(s) for each dataset.
2328 Only some properties can be edited.
2329 See the
2330 .Sx Properties
2331 section for more information on what properties can be set and acceptable
2332 values.
2333 Numeric values can be specified as exact values, or in a human-readable form
2334 with a suffix of
2335 .Sy B , K , M , G , T , P , E , Z
2336 .Po for bytes, kilobytes, megabytes, gigabytes, terabytes, petabytes, exabytes,
2337 or zettabytes, respectively
2338 .Pc .
2339 User properties can be set on snapshots.
2340 For more information, see the
2341 .Sx User Properties
2342 section.
2343 .It Xo
2344 .Nm
2345 .Cm get
2346 .Op Fl r Ns | Ns Fl d Ar depth
2347 .Op Fl Hp
2348 .Oo Fl o Ar field Ns Oo , Ns Ar field Oc Ns ... Oc
2349 .Oo Fl s Ar source Ns Oo , Ns Ar source Oc Ns ... Oc
2350 .Oo Fl t Ar type Ns Oo , Ns Ar type Oc Ns ... Oc
2351 .Cm all | Ar property Ns Oo , Ns Ar property Oc Ns ...
2352 .Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot Ns | Ns Ar bookmark Ns ...
2353 .Xc
2354 Displays properties for the given datasets.
2355 If no datasets are specified, then the command displays properties for all
2356 datasets on the system.
2357 For each property, the following columns are displayed:
2358 .Bd -literal
2359     name      Dataset name
2360     property  Property name
2361     value     Property value
2362     source    Property source.  Can either be local, default,
2363               temporary, inherited, or none (-).
2364 .Ed
2365 .Pp
2366 All columns are displayed by default, though this can be controlled by using the
2367 .Fl o
2368 option.
```

This command takes a comma-separated list of properties as described in the
.Sx Native Properties
and
.Sx User Properties
sections.
.Pp
The special value
.Sy all
can be used to display all properties that apply to the given dataset’s type
.Pq filesystem, volume, snapshot, or bookmark .
.Bl -tag -width "-H"
.It Fl H
Display output in a form more easily parsed by scripts.
Any headers are omitted, and fields are explicitly separated by a single tab
instead of an arbitrary amount of space.
.It Fl d Ar depth
Recursively display any children of the dataset, limiting the recursion to
.Ar depth .
A depth of
.Sy 1
will display only the dataset and its direct children.
.It Fl o Ar field
A comma-separated list of columns to display.
.Sy name Ns \&, Ns Sy property Ns \&, Ns Sy value Ns \&, Ns Sy source
is the default value.
.It Fl p
Display numbers in parsable
.Pq exact
values.
.It Fl r
Recursively display properties for any children.
.It Fl s Ar source
A comma-separated list of sources to display.
Those properties coming from a source other than those in this list are ignored.
Each source must be one of the following:
.Sy local ,
.Sy default ,
.Sy inherited ,
.Sy temporary ,
and
.Sy none .
The default value is all sources.
.It Fl t Ar type
A comma-separated list of types to display, where
.Ar type
is one of
.Sy filesystem ,
.Sy snapshot ,
.Sy volume ,
.Sy bookmark ,
or
.Sy all .
.El
.It Xo
.Nm
.Cm inherit
.Op Fl rS
.Ar property Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot Ns ...
.Xc
Clears the specified property, causing it to be inherited from an ancestor,
restored to default if no ancestor has the property set, or with the
.Fl S
option reverted to the received value if one exists.
See the
.Sx Properties
section for a listing of default values, and details on which properties can be

inherited.
.Bl -tag -width "-r"
.It Fl r
Recursively inherit the given property for all children.
.It Fl S
Revert the property to the received value if one exists; otherwise operate as
if the
.Fl S
option was not specified.
.El
.It Xo
.Nm
.Cm remap
.Ar filesystem Ns | Ns Ar volume
.Xc
Remap the indirect blocks in the given filesystem or volume so that they no
Remap the indirect blocks in the given fileystem or volume so that they no
longer reference blocks on previously removed vdevs and we can eventually
shrink the size of the indirect mapping objects for the previously removed
vdevs. Note that remapping all blocks might not be possible and that
references from snapshots will still exist and cannot be remapped.
.It Xo
.Nm
.Cm upgrade
.Xc
Displays a list of file systems that are not the most recent version.
.It Xo
.Nm
.Cm upgrade
.Fl v
.Xc
Displays a list of currently supported file system versions.
.It Xo
.Nm
.Cm upgrade
.Op Fl r
.Op Fl V Ar version
.Fl a | Ar filesystem
.Xc
Upgrades file systems to a new on-disk version.
Once this is done, the file systems will no longer be accessible on systems
running older versions of the software.
.Nm zfs Cm send
streams generated from new snapshots of these file systems cannot be accessed on
systems running older versions of the software.
.Pp
In general, the file system version is independent of the pool version.
See
.Xr zpool 1M
for information on the
.Nm zpool Cm upgrade
command.
.Pp
In some cases, the file system version and the pool version are interrelated and
the pool version must be upgraded before the file system version can be
upgraded.
.Bl -tag -width "-V"
.It Fl V Ar version
Upgrade to the specified
.Ar version .
If the
.Fl V
flag is not specified, this command upgrades to the most recent version.
This
option can only be used to increase the version number, and only up to the most
recent version supported by this software.

```
.It Fl a
Upgrade all file systems on all imported pools.
.It Ar filesystem
Upgrade the specified file system.
.It Fl r
Upgrade the specified file system and all descendent file systems.
.El
.It Xo
.Nm
.Cm userspace
.Op Fl Hinp
.Oo Fl o Ar field Ns Oo , Ns Ar field Oc Ns ... Oc
.Oo Fl s Ar field Oc Ns ...
.Oo Fl S Ar field Oc Ns ...
.Oo Fl t Ar type Ns Oo , Ns Ar type Oc Ns ... Oc
.Ar filesystem Ns | Ns Ar snapshot
.Xc
Displays space consumed by, and quotas on, each user in the specified filesystem
or snapshot.
This corresponds to the
.Sy userused@ Ns Em user
and
.Sy userquota@ Ns Em user
properties.
.Bl -tag -width "-H"
.It Fl H
Do not print headers, use tab-delimited output.
.It Fl S Ar field
Sort by this field in reverse order.
See
.Fl s .
.It Fl i
Translate SID to POSIX ID.
The POSIX ID may be ephemeral if no mapping exists.
Normal POSIX interfaces
.Po for example,
.Xr stat 2 ,
.Nm ls Fl l
.Pc
perform this translation, so the
.Fl i
option allows the output from
.Nm zfs Cm userspace
to be compared directly with those utilities.
However,
.Fl i
may lead to confusion if some files were created by an SMB user before a
SMB-to-POSIX name mapping was established.
In such a case, some files will be owned by the SMB entity and some by the POSIX
entity.
However, the
.Fl i
option will report that the POSIX entity has the total usage and quota for both.
.It Fl n
Print numeric ID instead of user/group name.
.It Fl o Ar field Ns Oo , Ns Ar field Oc Ns ...
Display only the specified fields from the following set:
.Sy type ,
.Sy name ,
.Sy used ,
.Sy quota .
The default is to display all fields.
.It Fl p
Use exact
.Pq parsable
numeric output.
```



```
.It Fl s Ar field
Sort output by this field.
The
.Fl s
and
.Fl S
flags may be specified multiple times to sort first by one field, then by
another.
The default is
.Fl s Sy type Fl s Sy name .
.It Fl t Ar type Ns Oo , Ns Ar type Oc Ns ...
Print only the specified types from the following set:
.Sy all ,
.Sy posixuser ,
.Sy smbuser ,
.Sy posixgroup ,
.Sy smbgroup .
The default is
.Fl t Sy posixuser Ns \&, Ns Sy smbuser .
The default can be changed to include group types.
.El
.It Xo
.Nm
.Cm groupspace
.Op Fl Hinp
.Oo Fl o Ar field Ns Oo , Ns Ar field Oc Ns ... Oc
.Oo Fl s Ar field Oc Ns ...
.Oo Fl S Ar field Oc Ns ...
.Oo Fl t Ar type Ns Oo , Ns Ar type Oc Ns ... Oc
.Ar filesystem Ns | Ns Ar snapshot
.Xc
Displays space consumed by, and quotas on, each group in the specified
filesystem or snapshot.
This subcommand is identical to
.Nm zfs Cm userspace ,
except that the default types to display are
.Fl t Sy posixgroup Ns \&, Ns Sy smbgroup .
.It Xo
.Nm
.Cm mount
.Xc
Displays all ZFS file systems currently mounted.
.It Xo
.Nm
.Cm mount
.Op Fl Ov
.Op Fl o Ar options
.Fl a | Ar filesystem
.Xc
Mounts ZFS file systems.
.Bl -tag -width "-O"
.It Fl O
Perform an overlay mount.
See
.Xr mount 1M
for more information.
.It Fl a
Mount all available ZFS file systems.
Invoked automatically as part of the boot process.
.It Ar filesystem
Mount the specified filesystem.
.It Fl o Ar options
An optional, comma-separated list of mount options to use temporarily for the
duration of the mount.
See the
.Sx Temporary Mount Point Properties
```

```
2632 section for details.
2633 .It Fl v
2634 Report mount progress.
2635 .El
2636 .It Xo
2637 .Nm
2638 .Cm unmount
2639 .Op Fl f
2640 .Fl a | Ar filesystem Ns | Ns Ar mountpoint
2641 .Xc
2642 Unmounts currently mounted ZFS file systems.
2643 .Bl -tag -width "-a"
2644 .It Fl a
2645 Unmount all available ZFS file systems.
2646 Invoked automatically as part of the shutdown process.
2647 .It Ar filesystem Ns | Ns Ar mountpoint
2648 Unmount the specified filesystem.
2649 The command can also be given a path to a ZFS file system mount point on the
2650 system.
2651 .It Fl f
2652 Forcefully unmount the file system, even if it is currently in use.
2653 .El
2654 .It Xo
2655 .Nm
2656 .Cm share
2657 .Fl a | Ar filesystem
2658 .Xc
2659 Shares available ZFS file systems.
2660 .Bl -tag -width "-a"
2661 .It Fl a
2662 Share all available ZFS file systems.
2663 Invoked automatically as part of the boot process.
2664 .It Ar filesystem
2665 Share the specified filesystem according to the
2666 .Sy sharenfs
2667 and
2668 .Sy sharesmb
2669 properties.
2670 File systems are shared when the
2671 .Sy sharenfs
2672 or
2673 .Sy sharesmb
2674 property is set.
2675 .El
2676 .It Xo
2677 .Nm
2678 .Cm unshare
2679 .Fl a | Ar filesystem Ns | Ns Ar mountpoint
2680 .Xc
2681 Unshares currently shared ZFS file systems.
2682 .Bl -tag -width "-a"
2683 .It Fl a
2684 Unshare all available ZFS file systems.
2685 Invoked automatically as part of the shutdown process.
2686 .It Ar filesystem Ns | Ns Ar mountpoint
2687 Unshare the specified filesystem.
2688 The command can also be given a path to a ZFS file system shared on the system.
2689 .El
2690 .It Xo
2691 .Nm
2692 .Cm bookmark
2693 .Ar snapshot bookmark
2694 .Xc
2695 Creates a bookmark of the given snapshot.
2696 Bookmarks mark the point in time when the snapshot was created, and can be used
2697 as the incremental source for a
```



```
2698 .Nm zfs Cm send
2699 command.
2700 .Pp
2701 This feature must be enabled to be used.
2702 See
2703 .Xr zpool-features 5
2704 for details on ZFS feature flags and the
2705 .Sy bookmarks
2706 feature.
2707 .It Xo
2708 .Nm
2709 .Cm send
2710 .Op Fl DLPRcenpv
2711 .Op Oo Fl I Ns | Ns Fl i Oc Ar snapshot
2712 .Ar snapshot
2713 .Xc
2714 Creates a stream representation of the second
2715 .Ar snapshot ,
2716 which is written to standard output.
2717 The output can be redirected to a file or to a different system
2718 .Po for example, using
2719 .Xr ssh 1
2720 .Pc .
2721 By default, a full stream is generated.
2722 .Bl -tag -width "-D"
2723 .It Fl D, -dedup
2724 Generate a deduplicated stream.
2725 Blocks which would have been sent multiple times in the send stream will only be
2726 sent once.
2727 The receiving system must also support this feature to receive a deduplicated
2728 stream.
2729 This flag can be used regardless of the dataset’s
2730 .Sy dedup
2731 property, but performance will be much better if the filesystem uses a
2732 dedup-capable checksum
2733 .Po for example,
2734 .Sy sha256
2735 .Pc .
2736 .It Fl I Ar snapshot
2737 Generate a stream package that sends all intermediary snapshots from the first
2738 snapshot to the second snapshot.
2739 For example,
2740 .Fl I Em @a Em fs@d
2741 is similar to
2742 .Fl i Em @a Em fs@b Ns \&; Fl i Em @b Em fs@c Ns \&; Fl i Em @c Em fs@d .
2743 The incremental source may be specified as with the
2744 .Fl i
2745 option.
2746 .It Fl L, -large-block
2747 Generate a stream which may contain blocks larger than 128KB.
2748 This flag has no effect if the
2749 .Sy large_blocks
2750 pool feature is disabled, or if the
2751 .Sy recordsize
2752 property of this filesystem has never been set above 128KB.
2753 The receiving system must have the
2754 .Sy large_blocks
2755 pool feature enabled as well.
2756 See
2757 .Xr zpool-features 5
2758 for details on ZFS feature flags and the
2759 .Sy large_blocks
2760 feature.
2761 .It Fl P, -parsable
2762 Print machine-parsable verbose information about the stream package generated.
2763 .It Fl R, -replicate
```

Generate a replication stream package, which will replicate the specified
file system, and all descendent file systems, up to the named snapshot.
When received, all properties, snapshots, descendent file systems, and clones
are preserved.
.Pp
If the
.Fl i
or
.Fl I
flags are used in conjunction with the
.Fl R
flag, an incremental replication stream is generated.
The current values of properties, and current snapshot and file system names are
set when the stream is received.
If the
.Fl F
flag is specified when this stream is received, snapshots and file systems that
do not exist on the sending side are destroyed.
.It Fl e, -embed
Generate a more compact stream by using
.Sy WRITE_EMBEDDED
records for blocks which are stored more compactly on disk by the
.Sy embedded_data
pool feature.
This flag has no effect if the
.Sy embedded_data
feature is disabled.
The receiving system must have the
.Sy embedded_data
feature enabled.
If the
.Sy lz4_compress
feature is active on the sending system, then the receiving system must have
that feature enabled as well.
See
.Xr zpool-features 5
for details on ZFS feature flags and the
.Sy embedded_data
feature.
.It Fl c, -compressed
Generate a more compact stream by using compressed WRITE records for blocks
which are compressed on disk and in memory
.Po see the
.Sy compression
property for details
.Pc .
If the
.Sy lz4_compress
feature is active on the sending system, then the receiving system must have
that feature enabled as well.
If the
.Sy large_blocks
feature is enabled on the sending system but the
.Fl L
option is not supplied in conjunction with
.Fl c ,
then the data will be decompressed before sending so it can be split into
smaller block sizes.
.It Fl i Ar snapshot
Generate an incremental stream from the first
.Ar snapshot
.Pq the incremental source
to the second
.Ar snapshot
.Pq the incremental target .
The incremental source can be specified as the last component of the snapshot

name
.Po the
.Sy @
character and following
.Pc
and it is assumed to be from the same file system as the incremental target.
.Pp
If the destination is a clone, the source may be the origin snapshot, which must
be fully specified
.Po for example,
.Em pool/fs@origin ,
not just
.Em @origin
.Pc .
.It Fl n, -dryrun
Do a dry-run
.Pq Qq No-op
send.
Do not generate any actual send data.
This is useful in conjunction with the
.Fl v
or
.Fl P
flags to determine what data will be sent.
In this case, the verbose output will be written to standard output
.Po contrast with a non-dry-run, where the stream is written to standard output
and the verbose output goes to standard error
.Pc .
.It Fl p, -props
Include the dataset’s properties in the stream.
This flag is implicit when
.Fl R
is specified.
The receiving system must also support this feature.
.It Fl v, -verbose
Print verbose information about the stream package generated.
This information includes a per-second report of how much data has been sent.
.Pp
The format of the stream is committed.
You will be able to receive your streams on future versions of ZFS .
.El
.It Xo
.Nm
.Cm send
.Op Fl Lce
.Op Fl i Ar snapshot Ns | Ns Ar bookmark
.Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot
.Xc
Generate a send stream, which may be of a filesystem, and may be incremental
from a bookmark.
If the destination is a filesystem or volume, the pool must be read-only, or the
filesystem must not be mounted.
When the stream generated from a filesystem or volume is received, the default
snapshot name will be
.Qq --head-- .
.Bl -tag -width "-L"
.It Fl L, -large-block
Generate a stream which may contain blocks larger than 128KB.
This flag has no effect if the
.Sy large_blocks
pool feature is disabled, or if the
.Sy recordsize
property of this filesystem has never been set above 128KB.
The receiving system must have the
.Sy large_blocks
pool feature enabled as well.

```
2896 See
2897 .Xr zpool-features 5
2898 for details on ZFS feature flags and the
2899 .Sy large_blocks
2900 feature.
2901 .It Fl c, -compressed
2902 Generate a more compact stream by using compressed WRITE records for blocks
2903 which are compressed on disk and in memory
2904 .Po see the
2905 .Sy compression
2906 property for details
2907 .Pc .
2908 If the
2909 .Sy lz4_compress
2910 feature is active on the sending system, then the receiving system must have
2911 that feature enabled as well.
2912 If the
2913 .Sy large_blocks
2914 feature is enabled on the sending system but the
2915 .Fl L
2916 option is not supplied in conjunction with
2917 .Fl c ,
2918 then the data will be decompressed before sending so it can be split into
2919 smaller block sizes.
2920 .It Fl e, -embed
2921 Generate a more compact stream by using
2922 .Sy WRITE_EMBEDDED
2923 records for blocks which are stored more compactly on disk by the
2924 .Sy embedded_data
2925 pool feature.
2926 This flag has no effect if the
2927 .Sy embedded_data
2928 feature is disabled.
2929 The receiving system must have the
2930 .Sy embedded_data
2931 feature enabled.
2932 If the
2933 .Sy lz4_compress
2934 feature is active on the sending system, then the receiving system must have
2935 that feature enabled as well.
2936 See
2937 .Xr zpool-features 5
2938 for details on ZFS feature flags and the
2939 .Sy embedded_data
2940 feature.
2941 .It Fl i Ar snapshot Ns | Ns Ar bookmark
2942 Generate an incremental send stream.
2943 The incremental source must be an earlier snapshot in the destination's history.
2944 It will commonly be an earlier snapshot in the destination's file system, in
2945 which case it can be specified as the last component of the name
2946 .Po the
2947 .Sy #
2948 or
2949 .Sy @
2950 character and following
2951 .Pc .
2952 .Pp
2953 If the incremental target is a clone, the incremental source can be the origin
2954 snapshot, or an earlier snapshot in the origin's filesystem, or the origin's
2955 origin, etc.
2956 .El
2957 .It Xo
2958 .Nm
2959 .Cm send
2960 .Op Fl Penv
2961 .Fl t
```



```
2962 .Ar receive_resume_token
2963 .Xc
2964 Creates a send stream which resumes an interrupted receive.
2965 The
2966 .Ar receive_resume_token
2967 is the value of this property on the filesystem or volume that was being
2968 received into.
2969 See the documentation for
2970 .Sy zfs receive -s
2971 for more details.
2972 .It Xo
2973 .Nm
2974 .Cm receive
2975 .Op Fl Fnsuv
2976 .Op Fl o Sy origin Ns = Ns Ar snapshot
2977 .Ar filesystem Ns | Ns Ar volume Ns | Ns Ar snapshot
2978 .Xc
2979 .It Xo
2980 .Nm
2981 .Cm receive
2982 .Op Fl Fnsuv
2983 .Op Fl d Ns | Ns Fl e
2984 .Op Fl o Sy origin Ns = Ns Ar snapshot
2985 .Ar filesystem
2986 .Xc
2987 Creates a snapshot whose contents are as specified in the stream provided on
2988 standard input.
2989 If a full stream is received, then a new file system is created as well.
2990 Streams are created using the
2991 .Nm zfs Cm send
2992 subcommand, which by default creates a full stream.
2993 .Nm zfs Cm recv
2994 can be used as an alias for
2995 .Nm zfs Cm receive.
2996 .Pp
2997 If an incremental stream is received, then the destination file system must
2998 already exist, and its most recent snapshot must match the incremental stream's
2999 source.
3000 For
3001 .Sy zvols ,
3002 the destination device link is destroyed and recreated, which means the
3003 .Sy zvol
3004 cannot be accessed during the
3005 .Cm receive
3006 operation.
3007 .Pp
3008 When a snapshot replication package stream that is generated by using the
3009 .Nm zfs Cm send Fl R
3010 command is received, any snapshots that do not exist on the sending location are
3011 destroyed by using the
3012 .Nm zfs Cm destroy Fl d
3013 command.
3014 .Pp
3015 The name of the snapshot
3016 .Pq and file system, if a full stream is received
3017 that this subcommand creates depends on the argument type and the use of the
3018 .Fl d
3019 or
3020 .Fl e
3021 options.
3022 .Pp
3023 If the argument is a snapshot name, the specified
3024 .Ar snapshot
3025 is created.
3026 If the argument is a file system or volume name, a snapshot with the same name
3027 as the sent snapshot is created within the specified
```

```
.Ar filesystem
or
.Ar volume .
If neither of the
.Fl d
or
.Fl e
options are specified, the provided target snapshot name is used exactly as
provided.
.Pp
The
.Fl d
and
.Fl e
options cause the file system name of the target snapshot to be determined by
appending a portion of the sent snapshot’s name to the specified target
.Ar filesystem .
If the
.Fl d
option is specified, all but the first element of the sent snapshot’s file
system path
.Pq usually the pool name
is used and any required intermediate file systems within the specified one are
created.
If the
.Fl e
option is specified, then only the last element of the sent snapshot’s file
system name
.Pq i.e. the name of the source file system itself
is used as the target file system name.
.Bl -tag -width "-F"
.It Fl F
Force a rollback of the file system to the most recent snapshot before
performing the receive operation.
If receiving an incremental replication stream
.Po for example, one generated by
.Nm zfs Cm send Fl R Op Fl i Ns | Ns Fl I
.Pc ,
destroy snapshots and file systems that do not exist on the sending side.
.It Fl d
Discard the first element of the sent snapshot’s file system name, using the
remaining elements to determine the name of the target file system for the new
snapshot as described in the paragraph above.
.It Fl e
Discard all but the last element of the sent snapshot’s file system name, using
that element to determine the name of the target file system for the new
snapshot as described in the paragraph above.
.It Fl n
Do not actually receive the stream.
This can be useful in conjunction with the
.Fl v
option to verify the name the receive operation would use.
.It Fl o Sy origin Ns = Ns Ar snapshot
Forces the stream to be received as a clone of the given snapshot.
If the stream is a full send stream, this will create the filesystem
described by the stream as a clone of the specified snapshot.
Which snapshot was specified will not affect the success or failure of the
receive, as long as the snapshot does exist.
If the stream is an incremental send stream, all the normal verification will be
performed.
.It Fl u
File system that is associated with the received stream is not mounted.
.It Fl v
Print verbose information about the stream and the time required to perform the
receive operation.
.It Fl s
```



```
If the receive is interrupted, save the partially received state, rather
than deleting it.
Interruption may be due to premature termination of the stream
.Po e.g. due to network failure or failure of the remote system
if the stream is being read over a network connection
.Pc ,
a checksum error in the stream, termination of the
.Nm zfs Cm receive
process, or unclean shutdown of the system.
.Pp
The receive can be resumed with a stream generated by
.Nm zfs Cm send Fl t Ar token ,
where the
.Ar token
is the value of the
.Sy receive_resume_token
property of the filesystem or volume which is received into.
.Pp
To use this flag, the storage pool must have the
.Sy extensible_dataset
feature enabled.
See
.Xr zpool-features 5
for details on ZFS feature flags.
.El
.It Xo
.Nm
.Cm receive
.Fl A
.Ar filesystem Ns | Ns Ar volume
.Xc
Abort an interrupted
.Nm zfs Cm receive Fl s ,
deleting its saved partially received state.
.It Xo
.Nm
.Cm allow
.Ar filesystem Ns | Ns Ar volume
.Xc
Displays permissions that have been delegated on the specified filesystem or
volume.
See the other forms of
.Nm zfs Cm allow
for more information.
.It Xo
.Nm
.Cm allow
.Op Fl dglu
.Ar user Ns | Ns Ar group Ns Oo , Ns Ar user Ns | Ns Ar group Oc Ns ...
.Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
.Ar setname Oc Ns ...
.Ar filesystem Ns | Ns Ar volume
.Xc
.It Xo
.Nm
.Cm allow
.Op Fl dl
.Fl e Ns | Ns Sy everyone
.Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
.Ar setname Oc Ns ...
.Ar filesystem Ns | Ns Ar volume
.Xc
Delegates ZFS administration permission for the file systems to non-privileged
users.
.Bl -tag -width "-d"
.It Fl d
```

Allow only for the descendent file systems.
.It Fl e Ns | Ns Sy everyone
Specifies that the permissions be delegated to everyone.
.It Fl g Ar group Ns Oo , Ns Ar group Oc Ns ...
Explicitly specify that permissions are delegated to the group.
.It Fl l
Allow
.Qq locally
only for the specified file system.
.It Fl u Ar user Ns Oo , Ns Ar user Oc Ns ...
Explicitly specify that permissions are delegated to the user.
.It Ar user Ns | Ns Ar group Ns Oo , Ns Ar user Ns | Ns Ar group Oc Ns ...
Specifies to whom the permissions are delegated.
Multiple entities can be specified as a comma-separated list.
If neither of the
.Fl gu
options are specified, then the argument is interpreted preferentially as the
keyword
.Sy everyone ,
then as a user name, and lastly as a group name.
To specify a user or group named
.Qq everyone ,
use the
.Fl g
or
.Fl u
options.
To specify a group with the same name as a user, use the
.Fl g
options.
.It Xo
.Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
.Ar setname Oc Ns ...
.Xc
The permissions to delegate.
Multiple permissions may be specified as a comma-separated list.
Permission names are the same as ZFS subcommand and property names.
See the property list below.
Property set names, which begin with
.Sy @ ,
may be specified.
See the
.Fl s
form below for details.
.El
.Pp
If neither of the
.Fl dl
options are specified, or both are, then the permissions are allowed for the
file system or volume, and all of its descendents.
.Pp
Permissions are generally the ability to use a ZFS subcommand or change a ZFS
property.
The following permissions are available:
.Bd -literal
NAME   TYPE   NOTES
allow   subcommand   Must also have the permission that is
                               being allowed
clone   subcommand   Must also have the 'create' ability and
                               'mount' ability in the origin file system
create   subcommand   Must also have the 'mount' ability
destroy   subcommand   Must also have the 'mount' ability
diff   subcommand   Allows lookup of paths within a dataset
                               given an object number, and the ability
                               to create snapshots necessary to
                               'zfs diff'.

mount   subcommand   Allows mount/umount of ZFS datasets
promote   subcommand   Must also have the 'mount' and 'promote'
                               ability in the origin file system
receive   subcommand   Must also have the 'mount' and 'create'
                               ability
rename   subcommand   Must also have the 'mount' and 'create'
                               ability in the new parent
rollback   subcommand   Must also have the 'mount' ability
send   subcommand
share   subcommand   Allows sharing file systems over NFS
                               or SMB protocols
snapshot   subcommand   Must also have the 'mount' ability

groupquota   other   Allows accessing any groupquota@...
                               property
groupused   other   Allows reading any groupused@... property
userprop   other   Allows changing any user property
userquota   other   Allows accessing any userquota@...
                               property
userused   other   Allows reading any userused@... property

aclinherit   property
aclmode   property
atime   property
canmount   property
casesensitivity property
checksum   property
compression   property
copies   property
devices   property
exec   property
filesystem_limit property
mountpoint   property
nbmand   property
normalization   property
primarycache   property
quota   property
readonly   property
recordsize   property
refquota   property
refreservation property
reservation   property
secondarycache property
setuid   property
sharenfs   property
sharesmb   property
snapdir   property
snapshot_limit property
utf8only   property
version   property
volblocksize   property
volsize   property
vscan   property
xattr   property
zoned   property
.Ed
.It Xo
.Nm
.Cm allow
.Fl c
.Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
.Ar setname Oc Ns ...
.Ar filesystem Ns | Ns Ar volume
.Xc
Sets
.Qq create time

```
3292 permissions.
3293 These permissions are granted
3294 .Pq locally
3295 to the creator of any newly-created descendent file system.
3296 .It Xo
3297 .Nm
3298 .Cm allow
3299 .Fl s No @ Ns Ar setname
3300 .Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
3301 .Ar setname Oc Ns ...
3302 .Ar filesystem Ns | Ns Ar volume
3303 .Xc
3304 Defines or adds permissions to a permission set.
3305 The set can be used by other
3306 .Nm zfs Cm allow
3307 commands for the specified file system and its descendents.
3308 Sets are evaluated dynamically, so changes to a set are immediately reflected.
3309 Permission sets follow the same naming restrictions as ZFS file systems, but the
3310 name must begin with
3311 .Sy @ ,
3312 and can be no more than 64 characters long.
3313 .It Xo
3314 .Nm
3315 .Cm unallow
3316 .Op Fl dglru
3317 .Ar user Ns | Ns Ar group Ns Oo , Ns Ar user Ns | Ns Ar group Oc Ns ...
3318 .Oo Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
3319 .Ar setname Oc Ns ... Oc
3320 .Ar filesystem Ns | Ns Ar volume
3321 .Xc
3322 .It Xo
3323 .Nm
3324 .Cm unallow
3325 .Op Fl dlr
3326 .Fl e Ns | Ns Sy everyone
3327 .Oo Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
3328 .Ar setname Oc Ns ... Oc
3329 .Ar filesystem Ns | Ns Ar volume
3330 .Xc
3331 .It Xo
3332 .Nm
3333 .Cm unallow
3334 .Op Fl r
3335 .Fl c
3336 .Oo Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
3337 .Ar setname Oc Ns ... Oc
3338 .Ar filesystem Ns | Ns Ar volume
3339 .Xc
3340 Removes permissions that were granted with the
3341 .Nm zfs Cm allow
3342 command.
3343 No permissions are explicitly denied, so other permissions granted are still in
3344 effect.
3345 For example, if the permission is granted by an ancestor.
3346 If no permissions are specified, then all permissions for the specified
3347 .Ar user ,
3348 .Ar group ,
3349 or
3350 .Sy everyone
3351 are removed.
3352 Specifying
3353 .Sy everyone
3354 .Po or using the
3355 .Fl e
3356 option
3357 .Pc
```



```
3358 only removes the permissions that were granted to everyone, not all permissions
3359 for every user and group.
3360 See the
3361 .Nm zfs Cm allow
3362 command for a description of the
3363 .Fl ldugec
3364 options.
3365 .Bl -tag -width "-r"
3366 .It Fl r
3367 Recursively remove the permissions from this file system and all descendents.
3368 .El
3369 .It Xo
3370 .Nm
3371 .Cm unallow
3372 .Op Fl r
3373 .Fl s No @ Ns Ar setname
3374 .Oo Ar perm Ns | Ns @ Ns Ar setname Ns Oo , Ns Ar perm Ns | Ns @ Ns
3375 .Ar setname Oc Ns ... Oc
3376 .Ar filesystem Ns | Ns Ar volume
3377 .Xc
3378 Removes permissions from a permission set.
3379 If no permissions are specified, then all permissions are removed, thus removing
3380 the set entirely.
3381 .It Xo
3382 .Nm
3383 .Cm hold
3384 .Op Fl r
3385 .Ar tag Ar snapshot Ns ...
3386 .Xc
3387 Adds a single reference, named with the
3388 .Ar tag
3389 argument, to the specified snapshot or snapshots.
3390 Each snapshot has its own tag namespace, and tags must be unique within that
3391 space.
3392 .Pp
3393 If a hold exists on a snapshot, attempts to destroy that snapshot by using the
3394 .Nm zfs Cm destroy
3395 command return
3396 .Er EBUSY .
3397 .Bl -tag -width "-r"
3398 .It Fl r
3399 Specifies that a hold with the given tag is applied recursively to the snapshots
3400 of all descendent file systems.
3401 .El
3402 .It Xo
3403 .Nm
3404 .Cm holds
3405 .Op Fl r
3406 .Ar snapshot Ns ...
3407 .Xc
3408 Lists all existing user references for the given snapshot or snapshots.
3409 .Bl -tag -width "-r"
3410 .It Fl r
3411 Lists the holds that are set on the named descendent snapshots, in addition to
3412 listing the holds on the named snapshot.
3413 .El
3414 .It Xo
3415 .Nm
3416 .Cm release
3417 .Op Fl r
3418 .Ar tag Ar snapshot Ns ...
3419 .Xc
3420 Removes a single reference, named with the
3421 .Ar tag
3422 argument, from the specified snapshot or snapshots.
3423 The tag must already exist for each snapshot.
```

```
3424 If a hold exists on a snapshot, attempts to destroy that snapshot by using the
3425 .Nm zfs Cm destroy
3426 command return
3427 .Er EBUSY .
3428 .Bl -tag -width "-r"
3429 .It Fl r
3430 Recursively releases a hold with the given tag on the snapshots of all
3431 descendent file systems.
3432 .El
3433 .It Xo
3434 .Nm
3435 .Cm diff
3436 .Op Fl FHt
3437 .Ar snapshot Ar snapshot Ns | Ns Ar filesystem
3438 .Xc
3439 Display the difference between a snapshot of a given filesystem and another
3440 snapshot of that filesystem from a later time or the current contents of the
3441 filesystem.
3442 The first column is a character indicating the type of change, the other columns
3443 indicate pathname, new pathname
3444 .Pq in case of rename ,
3445 change in link count, and optionally file type and/or change time.
3446 The types of change are:
3447 .Bd -literal
3448 -       The path has been removed
3449 +       The path has been created
3450 M       The path has been modified
3451 R       The path has been renamed
3452 .Ed
3453 .Bl -tag -width "-F"
3454 .It Fl F
3455 Display an indication of the type of file, in a manner similar to the
3456 .Fl
3457 option of
3458 .Xr ls 1 .
3459 .Bd -literal
3460 B       Block device
3461 C       Character device
3462 /       Directory
3463 >       Door
3464 |       Named pipe
3465 @       Symbolic link
3466 P       Event port
3467 =       Socket
3468 F       Regular file
3469 .Ed
3470 .It Fl H
3471 Give more parsable tab-separated output, without header lines and without
3472 arrows.
3473 .It Fl t
3474 Display the path's inode change time as the first column of output.
3475 .El
3476 .It Xo
3477 .Nm
3478 .Cm program
3479 .Op Fl jn
3480 .Op Fl t Ar timeout
3481 .Op Fl m Ar memory_limit
3482 .Ar pool script
3483 .Op Ar arg1 No ...
3484 .Xc
3485 Executes
3486 .Ar script
3487 as a ZFS channel program on
3488 .Ar pool .
3489 The ZFS channel
```



```
3490 program interface allows ZFS administrative operations to be run
3491 programmatically via a Lua script.
3492 The entire script is executed atomically, with no other administrative
3493 operations taking effect concurrently.
3494 A library of ZFS calls is made available to channel program scripts.
3495 Channel programs may only be run with root privileges.
3496 .sp
3497 For full documentation of the ZFS channel program interface, see the manual
3498 page for
3499 .Bl -tag -width ""
3500 .It Fl j
3501 Display channel program output in JSON format.
3502 When this flag is specified and standard output is empty -
3503 channel program encountered an error.
3504 The details of such an error will be printed to standard error in plain text.
3505 .It Fl n
3506 Executes a read-only channel program, which runs faster.
3507 The program cannot change on-disk state by calling functions from
3508 the zfs.sync submodule.
3509 The program can be used to gather information such as properties and
3510 determining if changes would succeed (zfs.check.*).
3511 Without this flag, all pending changes must be synced to disk before
3512 a channel program can complete.
3513 .It Fl t Ar timeout
3514 Execution time limit, in milliseconds.
3515 If a channel program executes for longer than the provided timeout, it will
3516 be stopped and an error will be returned.
3517 The default timeout is 1000 ms, and can be set to a maximum of 10000 ms.
3518 .It Fl m Ar memory-limit
3519 Memory limit, in bytes.
3520 If a channel program attempts to allocate more memory than the given limit,
3521 it will be stopped and an error returned.
3522 The default memory limit is 10 MB, and can be set to a maximum of 100 MB.
3523 .sp
3524 All remaining argument strings are passed directly to the channel program as
3525 arguments.
3526 See
3527 .Xr zfs-program 1M
3528 for more information.
3529 .El
3530 .El
3531 .Sh EXIT STATUS
3532 The
3533 .Nm
3534 utility exits 0 on success, 1 if an error occurs, and 2 if invalid command line
3535 options were specified.
3536 .Sh EXAMPLES
3537 .Bl -tag -width ""
3538 .It Sy Example 1 No Creating a ZFS File System Hierarchy
3539 The following commands create a file system named
3540 .Em pool/home
3541 and a file system named
3542 .Em pool/home/bob .
3543 The mount point
3544 .Pa /export/home
3545 is set for the parent file system, and is automatically inherited by the child
3546 file system.
3547 .Bd -literal
3548 # zfs create pool/home
3549 # zfs set mountpoint=/export/home pool/home
3550 # zfs create pool/home/bob
3551 .Ed
3552 .It Sy Example 2 No Creating a ZFS Snapshot
3553 The following command creates a snapshot named
3554 .Sy yesterday .
3555 This snapshot is mounted on demand in the
```

```
.Pa .zfs/snapshot
directory at the root of the
.Em pool/home/bob
file system.
.Bd -literal
# zfs snapshot pool/home/bob@yesterday
.Ed
.It Sy Example 3 No Creating and Destroying Multiple Snapshots
The following command creates snapshots named
.Sy yesterday
of
.Em pool/home
and all of its descendent file systems.
Each snapshot is mounted on demand in the
.Pa .zfs/snapshot
directory at the root of its file system.
The second command destroys the newly created snapshots.
.Bd -literal
# zfs snapshot -r pool/home@yesterday
# zfs destroy -r pool/home@yesterday
.Ed
.It Sy Example 4 No Disabling and Enabling File System Compression
The following command disables the
.Sy compression
property for all file systems under
.Em pool/home .
The next command explicitly enables
.Sy compression
for
.Em pool/home/anne .
.Bd -literal
# zfs set compression=off pool/home
# zfs set compression=on pool/home/anne
.Ed
.It Sy Example 5 No Listing ZFS Datasets
The following command lists all active file systems and volumes in the system.
Snapshots are displayed if the
.Sy listsnaps
property is
.Sy on .
The default is
.Sy off .
See
.Xr zpool 1M
for more information on pool properties.
.Bd -literal
# zfs list
NAME                     USED  AVAIL  REFER  MOUNTPOINT
pool                     450K   457G    18K  /pool
pool/home                315K   457G    21K  /export/home
pool/home/anne            18K   457G    18K  /export/home/anne
pool/home/bob            276K   457G   276K  /export/home/bob
.Ed
.It Sy Example 6 No Setting a Quota on a ZFS File System
The following command sets a quota of 50 Gbytes for
.Em pool/home/bob .
.Bd -literal
# zfs set quota=50G pool/home/bob
.Ed
.It Sy Example 7 No Listing ZFS Properties
The following command lists all properties for
.Em pool/home/bob .
.Bd -literal
# zfs get all pool/home/bob
NAME           PROPERTY              VALUE                  SOURCE
pool/home/bob  type                  filesystem             -
```



```
pool/home/bob  creation              Tue Jul 21 15:53 2009  -
pool/home/bob  used                  21K                    -
pool/home/bob  available             20.0G                  -
pool/home/bob  referenced            21K                    -
pool/home/bob  compressratio         1.00x                  -
pool/home/bob  mounted               yes                    -
pool/home/bob  quota                 20G                    local
pool/home/bob  reservation           none                   default
pool/home/bob  recordsize            128K                   default
pool/home/bob  mountpoint            /pool/home/bob         default
pool/home/bob  sharenfs              off                    default
pool/home/bob  checksum              on                     default
pool/home/bob  compression           on                     local
pool/home/bob  atime                 on                     default
pool/home/bob  devices               on                     default
pool/home/bob  exec                  on                     default
pool/home/bob  setuid                on                     default
pool/home/bob  readonly              off                    default
pool/home/bob  zoned                 off                    default
pool/home/bob  snapdir               hidden                 default
pool/home/bob  aclmode               discard                default
pool/home/bob  aclinherit            restricted             default
pool/home/bob  canmount              on                     default
pool/home/bob  xattr                 on                     default
pool/home/bob  copies                1                      default
pool/home/bob  version               4                      -
pool/home/bob  utf8only              off                    -
pool/home/bob  normalization         none                   -
pool/home/bob  casesensitivity       sensitive              -
pool/home/bob  vscan                 off                    default
pool/home/bob  nbmand                off                    default
pool/home/bob  sharesmb              off                    default
pool/home/bob  refquota              none                   default
pool/home/bob  refreservation        none                   default
pool/home/bob  primarycache          all                    default
pool/home/bob  secondarycache        all                    default
pool/home/bob  usedbysnapshots       0                      -
pool/home/bob  usedbydataset         21K                    -
pool/home/bob  usedbychildren        0                      -
pool/home/bob  usedbyrefreservation  0                      -
.Ed
.Pp
The following command gets a single property value.
.Bd -literal
# zfs get -H -o value compression pool/home/bob
on
.Ed
The following command lists all properties with local settings for
.Em pool/home/bob .
.Bd -literal
# zfs get -r -s local -o name,property,value all pool/home/bob
NAME           PROPERTY              VALUE
pool/home/bob  quota                 20G
pool/home/bob  compression           on
.Ed
.It Sy Example 8 No Rolling Back a ZFS File System
The following command reverts the contents of
.Em pool/home/anne
to the snapshot named
.Sy yesterday ,
deleting all intermediate snapshots.
.Bd -literal
# zfs rollback -r pool/home/anne@yesterday
.Ed
.It Sy Example 9 No Creating a ZFS Clone
The following command creates a writable file system whose initial contents are
```

```
the same as
.Em pool/home/bob@yesterday .
.Bd -literal
# zfs clone pool/home/bob@yesterday pool/clone
.Ed
.It Sy Example 10 No Promoting a ZFS Clone
The following commands illustrate how to test out changes to a file system, and
then replace the original file system with the changed one, using clones, clone
promotion, and renaming:
.Bd -literal
# zfs create pool/project/production
  populate /pool/project/production with data
# zfs snapshot pool/project/production@today
# zfs clone pool/project/production@today pool/project/beta
  make changes to /pool/project/beta and test them
# zfs promote pool/project/beta
# zfs rename pool/project/production pool/project/legacy
# zfs rename pool/project/beta pool/project/production
  once the legacy version is no longer needed, it can be destroyed
# zfs destroy pool/project/legacy
.Ed
.It Sy Example 11 No Inheriting ZFS Properties
The following command causes
.Em pool/home/bob
and
.Em pool/home/anne
to inherit the
.Sy checksum
property from their parent.
.Bd -literal
# zfs inherit checksum pool/home/bob pool/home/anne
.Ed
.It Sy Example 12 No Remotely Replicating ZFS Data
The following commands send a full stream and then an incremental stream to a
remote machine, restoring them into
.Em poolB/received/fs@a
and
.Em poolB/received/fs@b ,
respectively.
.Em poolB
must contain the file system
.Em poolB/received ,
and must not initially contain
.Em poolB/received/fs .
.Bd -literal
# zfs send pool/fs@a | \e
  ssh host zfs receive poolB/received/fs@a
# zfs send -i a pool/fs@b | \e
  ssh host zfs receive poolB/received/fs
.Ed
.It Sy Example 13 No Using the zfs receive -d Option
The following command sends a full stream of
.Em poolA/fsA/fsB@snap
to a remote machine, receiving it into
.Em poolB/received/fsA/fsB@snap .
The
.Em fsA/fsB@snap
portion of the received snapshot's name is determined from the name of the sent
snapshot.
.Em poolB
must contain the file system
.Em poolB/received .
If
.Em poolB/received/fsA
does not exist, it is created as an empty file system.
.Bd -literal
```



```
# zfs send poolA/fsA/fsB@snap | \e
  ssh host zfs receive -d poolB/received
.Ed
.It Sy Example 14 No Setting User Properties
The following example sets the user-defined
.Sy com.example:department
property for a dataset.
.Bd -literal
# zfs set com.example:department=12345 tank/accounting
.Ed
.It Sy Example 15 No Performing a Rolling Snapshot
The following example shows how to maintain a history of snapshots with a
consistent naming scheme.
To keep a week's worth of snapshots, the user destroys the oldest snapshot,
renames the remaining snapshots, and then creates a new snapshot, as follows:
.Bd -literal
# zfs destroy -r pool/users@7daysago
# zfs rename -r pool/users@6daysago @7daysago
# zfs rename -r pool/users@5daysago @6daysago
# zfs rename -r pool/users@yesterday @5daysago
# zfs rename -r pool/users@yesterday @4daysago
# zfs rename -r pool/users@yesterday @3daysago
# zfs rename -r pool/users@yesterday @2daysago
# zfs rename -r pool/users@today @yesterday
# zfs snapshot -r pool/users@today
.Ed
.It Sy Example 16 No Setting sharenfs Property Options on a ZFS File System
The following commands show how to set
.Sy sharenfs
property options to enable
.Sy rw
access for a set of
.Sy IP
addresses and to enable root access for system
.Sy neo
on the
.Em tank/home
file system.
.Bd -literal
# zfs set sharenfs='rw=@123.123.0.0/16,root=neo' tank/home
.Ed
.Pp
If you are using
.Sy DNS
for host name resolution, specify the fully qualified hostname.
.It Sy Example 17 No Delegating ZFS Administration Permissions on a ZFS Dataset
The following example shows how to set permissions so that user
.Sy cindys
can create, destroy, mount, and take snapshots on
.Em tank/cindys .
The permissions on
.Em tank/cindys
are also displayed.
.Bd -literal
# zfs allow cindys create,destroy,mount,snapshot tank/cindys
# zfs allow tank/cindys
---- Permissions on tank/cindys --------------------------------------
Local+Descendent permissions:
        user cindys create,destroy,mount,snapshot
.Ed
.Pp
Because the
.Em tank/cindys
mount point permission is set to 755 by default, user
.Sy cindys
will be unable to mount file systems under
```

```
.Em tank/cindys .
Add an ACE similar to the following syntax to provide mount point access:
.Bd -literal
# chmod A+user:cindys:add_subdirectory:allow /tank/cindys
.Ed
.It Sy Example 18 No Delegating Create Time Permissions on a ZFS Dataset
The following example shows how to grant anyone in the group
.Sy staff
to create file systems in
.Em tank/users .
This syntax also allows staff members to destroy their own file systems, but not
destroy anyone else's file system.
The permissions on
.Em tank/users
are also displayed.
.Bd -literal
# zfs allow staff create,mount tank/users
# zfs allow -c destroy tank/users
# zfs allow tank/users
---- Permissions on tank/users --------------------------------------
Permission sets:
        destroy
Local+Descendent permissions:
        group staff create,mount
.Ed
.It Sy Example 19 No Defining and Granting a Permission Set on a ZFS Dataset
The following example shows how to define and grant a permission set on the
.Em tank/users
file system.
The permissions on
.Em tank/users
are also displayed.
.Bd -literal
# zfs allow -s @pset create,destroy,snapshot,mount tank/users
# zfs allow staff @pset tank/users
# zfs allow tank/users
---- Permissions on tank/users --------------------------------------
Permission sets:
        @pset create,destroy,mount,snapshot
Local+Descendent permissions:
        group staff @pset
.Ed
.It Sy Example 20 No Delegating Property Permissions on a ZFS Dataset
The following example shows to grant the ability to set quotas and reservations
on the
.Em users/home
file system.
The permissions on
.Em users/home
are also displayed.
.Bd -literal
# zfs allow cindys quota,reservation users/home
# zfs allow users/home
---- Permissions on users/home --------------------------------------
Local+Descendent permissions:
        user cindys quota,reservation
cindys% zfs set quota=10G users/home/marks
cindys% zfs get quota users/home/marks
NAME              PROPERTY  VALUE  SOURCE
users/home/marks  quota     10G    local
.Ed
.It Sy Example 21 No Removing ZFS Delegated Permissions on a ZFS Dataset
The following example shows how to remove the snapshot permission from the
.Sy staff
group on the
.Em tank/users
```



```
file system.
The permissions on
.Em tank/users
are also displayed.
.Bd -literal
# zfs unallow staff snapshot tank/users
# zfs allow tank/users
---- Permissions on tank/users --------------------------------------
Permission sets:
        @pset create,destroy,mount,snapshot
Local+Descendent permissions:
        group staff @pset
.Ed
.It Sy Example 22 No Showing the differences between a snapshot and a ZFS Datase
The following example shows how to see what has changed between a prior
snapshot of a ZFS dataset and its current state.
The
.Fl F
option is used to indicate type information for the files affected.
.Bd -literal
# zfs diff -F tank/test@before tank/test
M       /       /tank/test/
M       F       /tank/test/linked      (+1)
R       F       /tank/test/oldname -> /tank/test/newname
-       F       /tank/test/deleted
+       F       /tank/test/created
M       F       /tank/test/modified
.Ed
.El
.Sh INTERFACE STABILITY
.Sy Committed .
.Sh SEE ALSO
.Xr gzip 1 ,
.Xr ssh 1 ,
.Xr mount 1M ,
.Xr share 1M ,
.Xr sharemgr 1M ,
.Xr unshare 1M ,
.Xr zonecfg 1M ,
.Xr zpool 1M ,
.Xr chmod 2 ,
.Xr stat 2 ,
.Xr write 2 ,
.Xr fsync 3C ,
.Xr dfstab 4 ,
.Xr acl 5 ,
.Xr attributes 5
```

```
**********************************************************
   10587 Fri Jan 11 21:14:02 2019
new/usr/src/man/man2/sysinfo.2
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
```

```
'\" te
.\" Copyright (c) 2009, Sun Microsystems, Inc. All Rights Reserved.
.\" Copyright 1989 AT&T
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH SYSINFO 2 "Sep 7, 2015"
.SH NAME
sysinfo \- get and set system information strings
.SH SYNOPSIS
.LP
.nf
#include <sys/systeminfo.h>

\fBint\fR \fBsysinfo\fR(\fBint\fR \fIcommand\fR, \fBchar *\fR\fIbuf\fR, \fBlong\
.fi

.SH DESCRIPTION
.LP
The \fBsysinfo()\fR function copies information relating to the operating
system on which the process is executing into the buffer pointed to by
\fIbuf\fR. It can also set certain information where appropriate commands are
available. The \fIcount\fR parameter indicates the size of the buffer.
.sp
.LP
The POSIX P1003.1 interface (see \fBstandards\fR(5)) \fBsysconf\fR(3C) provides
a similar class of configuration information, but returns an integer rather
than a string.
.sp
.LP
The values for \fIcommand\fR are as follows:
.sp
.ne 2
.na
\fB\fBSI_SYSNAME\fR\fR
.ad
.sp .6
.RS 4n
Copy into the array pointed to by \fIbuf\fR the string that would be returned
by \fBuname\fR(2) in the \fIsysname\fR field. This is the name of the
implementation of the operating system, for example, \fBSunOS\fR or \fBUTS\fR.
.RE

.sp
.ne 2
.na
\fB\fBSI_HOSTNAME\fR\fR
.ad
.sp .6
.RS 4n
Copy into the array pointed to by \fIbuf\fR a string that names the present
host machine. This is the string that would be returned by \fBuname()\fR in the
\fInodename\fR field. This hostname or nodename is often the name the machine is
known by locally. The \fIhostname\fR is the name of this machine as a node in
some network. Different networks might have different names for the node, but
presenting the nodename to the appropriate network directory or name-to-address
mapping service should produce a transport end point address. The name might
not be fully qualified. Internet host names can be up to \fB256\fR bytes in
```



```
length (plus the terminating null).
.RE

.sp
.ne 2
.na
\fB\fBSI_SET_HOSTNAME\fR\fR
.ad
.sp .6
.RS 4n
Copy the null-terminated contents of the array pointed to by \fIbuf\fR into the
string maintained by the kernel whose value will be returned by succeeding
calls to \fBsysinfo()\fR with the command \fBSI_HOSTNAME\fR. This command
requires that {\fBPRIV_SYS_ADMIN\fR} is asserted in the effective set of the
calling process.
.RE

.sp
.ne 2
.na
\fB\fBSI_RELEASE\fR\fR
.ad
.sp .6
.RS 4n
Copy into the array pointed to by \fIbuf\fR the string that would be returned
by \fBuname\fR(2) in the \fIrelease\fR field. Typical values might be \fB5.2\fR
or \fB4.1\fR.
.RE

.sp
.ne 2
.na
\fB\fBSI_VERSION\fR\fR
.ad
.sp .6
.RS 4n
Copy into the array pointed to by \fIbuf\fR the string that would be returned
by \fBuname\fR(2) in the \fIversion\fR field. The syntax and semantics of this
string are defined by the system provider.
.RE

.sp
.ne 2
.na
\fB\fBSI_MACHINE\fR\fR
.ad
.sp .6
.RS 4n
Copy into the array pointed to by \fIbuf\fR the string that would be returned
by \fBuname\fR(2) in the \fImachine\fR field, for example, \fBsun4u\fR.
.RE

.sp
.ne 2
.na
\fB\fBSI_ARCHITECTURE\fR\fR
.ad
.sp .6
.RS 4n
Copy into the array pointed to by \fIbuf\fR a string describing the basic
instruction set architecture of the current system, for example, \fBsparc\fR,
\fBmc68030\fR, \fBm32100\fR, or \fBi386\fR. These names might not match
predefined names in the C language compilation system.
.RE

.sp
```

```
.ne 2
.na
\fB\fBSI_ARCHITECTURE_64\fR\fR
.ad
.sp .6
.RS 4n
Copy into the array pointed to by \fIbuf\fR a string describing the 64-bit
instruction set architecture of the current system, for example, \fBsparcv9\fR
or \fBamd64\fR.  These names might not match predefined names in the C language
compilation system.  This subcode is not recognized on systems that do not
allow a 64-bit application to run.
.RE

.sp
.ne 2
.na
\fB\fBSI_ARCHITECTURE_32\fR\fR
.ad
.sp .6
.RS 4n
Copy into the array pointed to by \fIbuf\fR a string describing the 32-bit
instruction set architecture of the current system, for example, \fBsparc\fR or
\fBi386\fR.  These names might not match predefined names in the C language
compilation system.
.RE

.sp
.ne 2
.na
\fB\fBSI_ARCHITECTURE_K\fR\fR
.ad
.sp .6
.RS 4n
Copy into the array pointed to by \fIbuf\fR a string describing the kernel
instruction set architecture of the current system for example \fBsparcv9\fR or
\fBi386\fR.  These names might not match predefined names in the C language
compilation system.
.RE

.sp
.ne 2
.na
\fB\fBSI_ARCHITECTURE_NATIVE\fR\fR
.ad
.sp .6
.RS 4n
Copy into the array pointed to by \fIbuf\fR a string describing the native
instruction set architecture of the current system, for example \fBsparcv9\fR
or \fBi386\fR.  These names might not match predefined names in the C language
compilation system.
.RE

.sp
.ne 2
.na
\fB\fBSI_ISALIST\fR\fR
.ad
.sp .6
.RS 4n
Copy into the array pointed to by \fIbuf\fR the names of the variant
instruction set architectures executable on the current system.
.sp
The names are space-separated and are ordered in the sense of best performance.
That is, earlier-named instruction sets might contain more instructions than
later-named instruction sets; a program that is compiled for an earlier-named
instruction set will most likely run faster on this machine than the same
```



```
program compiled for a later-named instruction set.
.sp
Programs compiled for an instruction set that does not appear in the list will
most likely experience performance degradation or not run at all on this
machine.
.sp
The instruction set names known to the system are listed in \fBisalist\fR(5);
these names might not match predefined names or compiler options in the C
language compilation system.
.sp
This command is obsolete and might be removed in a future release. See
\fBgetisax\fR(2) and the \fILinker and Libraries Guide\fR for a better way to
handle instruction set extensions.
.RE

.sp
.ne 2
.na
\fB\fBSI_PLATFORM\fR\fR
.ad
.sp .6
.RS 4n
Copy into the array pointed to by \fIbuf\fR a string describing the specific
model of the hardware platform, for example, \fBSUNW,Sun-Blade-1500\fR,
\fBSUNW,Sun-Fire-T200\fR, or \fBi86pc\fR.
.RE

.sp
.ne 2
.na
\fB\fBSI_HW_PROVIDER\fR\fR
.ad
.sp .6
.RS 4n
Copies the name of the hardware manufacturer into the array pointed to by
\fIbuf\fR.
.RE

.sp
.ne 2
.na
\fB\fBSI_HW_SERIAL\fR\fR
.ad
.sp .6
.RS 4n
Copy into the array pointed to by \fIbuf\fR a string which is the ASCII
representation of the hardware-specific serial number of the physical machine
on which the function is executed. This might be implemented in Read-Only
Memory, using software constants set when building the operating system, or by
other means, and might contain non-numeric characters. If the function is
executed within a non-global zone that emulates a host identifier, then the
ASCII representation of the zone’s host identifier is copied into the array
pointed to by \fIbuf\fR. It is anticipated that manufacturers will not issue
the same "serial number" to more than one physical machine. The pair of strings
returned by \fBSI_HW_PROVIDER\fR and \fBSI_HW_SERIAL\fR is not guaranteed to be
unique across all vendor’s SVR4 implementations and could change over the
lifetime of a given system.
.RE

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

Copies the Secure Remote Procedure Call domain name into the array pointed to by \fIbuf\fR.
.RE

.sp
.ne 2
.na
\fB\fBSI_SET_SRPC_DOMAIN\fR\fR
.ad
.sp .6
.RS 4n
Set the string to be returned by \fBsysinfo()\fR with the \fBSI_SRPC_DOMAIN\fR command to the value contained in the array pointed to by \fIbuf\fR. This command requires that {\fBPRIV_SYS_ADMIN\fR} is asserted in the effective set of the calling process.
.RE

.sp
.ne 2
.na
\fB\fBSI_DHCP_CACHE\fR\fR
.ad
.sp .6
.RS 4n
Copy into the array pointed to by \fIbuf\fR an ASCII string consisting of the
ASCII hexadecimal encoding of the name of the interface configured by
ASCII hexidecimal encoding of the name of the interface configured by
\fBboot\fR(1M) followed by the DHCPACK reply from the server. This command is intended for use only by the \fBdhcpagent\fR(1M) DHCP client daemon for the purpose of adopting the DHCP maintenance of the interface configured by \fBboot\fR.
.RE

.SH RETURN VALUES
.LP
Upon successful completion, the value returned indicates the buffer size in bytes required to hold the complete value and the terminating null character. If this value is no greater than the value passed in \fIcount\fR, the entire string was copied. If this value is greater than \fIcount\fR, the string copied into \fIbuf\fR has been truncated to \fIcount\fR\(mi1 bytes plus a terminating null character.
.sp
.LP
Otherwise, \(mi1 is returned and \fBerrno\fR is set to indicate the error.
.SH ERRORS
.LP
The \fBsysinfo()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBEFAULT\fR\fR
.ad
.RS 10n
The \fIbuf\fR argument does not point to a valid address.
.RE

.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 10n
The \fIcount\fR argument for a non-SET command is less than 0 or the data for a SET command exceeds the limits established by the implementation.
.RE

.sp
.ne 2
.na
\fB\fBEPERM\fR\fR
.ad
.RS 10n
The {\fBPRIV_SYS_ADMIN\fR} was not asserted in the effective set of the calling process.
.RE

.SH USAGE
.LP
In many cases there is no corresponding programming interface to set these values; such strings are typically settable only by the system administrator modifying entries in the \fB/etc/system\fR directory or the code provided by the particular OEM reading a serial number or code out of read-only memory, or hard-coded in the version of the operating system.
.sp
.LP
A good estimation for \fIcount\fR is 257, which is likely to cover all strings returned by this interface in typical installations.
.SH SEE ALSO
.LP
\fBboot\fR(1M), \fBdhcpagent\fR(1M), \fBgetisax\fR(2), \fBuname\fR(2), \fBgethostid\fR(3C), \fBgethostname\fR(3C), \fBsysconf\fR(3C), \fBisalist\fR(5), \fBprivileges\fR(5), \fBstandards\fR(5), \fBzones\fR(5)
.sp
.LP
\fILinker and Libraries Guide\fR

```
**********************************************************
    9133 Fri Jan 11 21:14:02 2019
new/usr/src/man/man2/utimes.2
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
.\"
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for
.\" permission to reproduce portions of its copyrighted documentation.
.\" Original documentation from The Open Group can be obtained online at
.\" http://www.opengroup.org/bookstore/.
.\"
.\" The Institute of Electrical and Electronics Engineers and The Open
.\" Group, have given us permission to reprint portions of their
.\" documentation.
.\"
.\" In the following statement, the phrase ‘‘this text’’ refers to portions
.\" of the system documentation.
.\"
.\" Portions of this text are reprinted and reproduced in electronic form
.\" in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition,
.\" Standard for Information Technology -- Portable Operating System
.\" Interface (POSIX), The Open Group Base Specifications Issue 6,
.\" Copyright (C) 2001-2004 by the Institute of Electrical and Electronics
.\" Engineers, Inc and The Open Group.  In the event of any discrepancy
.\" between these versions and the original IEEE and The Open Group
.\" Standard, the original IEEE and The Open Group Standard is the referee
.\" document.  The original Standard can be obtained online at
.\" http://www.opengroup.org/unix/online.html.
.\"
.\" This notice shall appear on any product containing this material.
.\"
.\" The contents of this file are subject to the terms of the
.\" Common Development and Distribution License (the "License").
.\" You may not use this file except in compliance with the License.
.\"
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
.\" or http://www.opensolaris.org/os/licensing.
.\" See the License for the specific language governing permissions
.\" and limitations under the License.
.\"
.\" When distributing Covered Code, include this CDDL HEADER in each
.\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
.\" If applicable, add the following below this CDDL HEADER, with the
.\" fields enclosed by brackets "[]" replaced with your own identifying
.\" information: Portions Copyright [yyyy] [name of copyright owner]
.\"
.\"
.\" Portions Copyright (c) 1992, X/Open Company Limited.  All Rights Reserved.
.\" Copyright (c) 2009, Sun Microsystems, Inc.  All Rights Reserved.
.\" Copyright (c) 2014, Joyent, Inc.
.\"
.TH UTIMES 2 "Dec 20, 2014"
.SH NAME
utimes, futimesat \- set file access and modification times
.SH SYNOPSIS
.LP
.nf
#include <sys/time.h>

\fBint\fR \fButimes\fR(\fBconst char *\fR\fIpath\fR, \fBconst struct timeval\fR
.fi

.LP
```



```
.nf
\fBint\fR \fBfutimesat\fR(\fBint\fR \fIfildes\fR, \fBconst char *\fR\fIpath\fR,
    \fBconst struct timeval\fR \fItimes\fR[2]);
.fi

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

\fBint\fR \fBfutimens\fR(\fBint\fR \fIfiledes\fR, \fBconst timespec_t\fR \fInsti

\fBint\fR \fButimensat\fR(\fBint\fR \fIfiledes\fR, \fBconst char *\fR\fIpath\fR,
    \fBconst timespec_t\fR \fInstimes[2]\fR, \fBint\fR \fIflag\fR);

.SH DESCRIPTION
.LP
The \fButimes()\fR function sets the access and modification times of the file
pointed to by the \fIpath\fR argument to the value of the \fItimes\fR argument.
It allows time specifications accurate to the microsecond.
.sp
.LP
The \fBfutimesat()\fR function also sets access and modification times.  See
\fBfsattr\fR(5). If \fIpath\fR is a relative path name, however,
\fBfutimesat()\fR resolves the path relative to the \fIfildes\fR argument
rather than the current working directory.  If \fIfildes\fR is set to
\fBAT_FDCWD\fR, defined in <\fBfcntl.h\fR>, \fBfutimesat()\fR resolves the path
relative to the current working directory.  If \fIpath\fR is a null pointer,
\fBfutimesat()\fR sets the access and modification times on the file referenced
by \fIfildes\fR. The \fIfildes\fR argument is ignored even when
\fBfutimesat()\fR is provided with an absolute path.
.sp
.LP
The \fItimes\fR argument is an array of \fBtimeval\fR structures. The first
array member represents the date and time of last access, and the second member
represents the date and time of last modification.  The times in the
\fBtimeval\fR structure are measured in seconds and microseconds since the
Epoch, although rounding toward the nearest second may occur.
.sp
.LP
If the \fItimes\fR argument is a null pointer, the access and modification
times of the file are set to the current time.  The effective user \fBID\fR of
the process must be the same as the owner of the file, or must have write
access to the file or the {\fBPRIV_FILE_OWNER\fR} privilege to use this call in
this manner. Upon completion, \fButimes()\fR will mark the time of the last
file status change, \fBst_ctime\fR, for update.
.sp
.LP
The \fBfutimens()\fR and \fButimensat()\fR functions also set access and
modification times; however, instead of taking \fBstruct timeval\fR, they take
\fBtimespec_t\fR which allows for nanosecond granularity. The \fBfutimens()\fR
function sets the access and modification times on the file descriptor
referenced by \fIfiledes\fR.
.sp
.LP
The \fButimensat()\fR function determines the file to set the access and
modification times in an similar way to \fBfutemsat()\fR. If the argument
\fIpath\fR is an absolute path, then the argument \fIfiledes\fR is ignored;
otherwise, \fIpath\fR is interpreted as a path relative to the directory
specified by \fIfiledes\fR. If \fIfiledes\fR is set to \fBAT_FDCWD\fR, then
\fIpath\fR is resolved relative to the current working directory. The behavior
when encountering symbolic links may be controlled by the value of the
\fIflag\fR argument. If the value of flag is the constant
\fBAT_SYMLINK_NOFOLLOW\fR, then when a symbolic link is encountered while
resolving a path, it will not be followed. Otherwise, the value of \fIflag\fR
should be \fB0\fR.
```

```
.SH RETURN VALUES
.LP
Upon successful completion, \fB0\fR is returned.  Otherwise, \fB\(mi1\fR is
returned, \fBerrno\fR is set to indicate the error, and the file times will not
be affected.
.SH ERRORS
.LP
The \fButimes()\fR, \fBfutimesat()\fR, \fBfutimens()\fR, and \fButimensat()\fR
functions will fail if:
.sp
.ne 2
.na
\fB\fBEACCES\fR\fR
.ad
.RS 16n
Search permission is denied by a component of the path prefix; or the
\fItimes\fR argument is a null pointer and the effective user \fBID\fR of the
process does not match the owner of the file and write access is denied.
.RE

.sp
.ne 2
.na
\fB\fBEFAULT\fR\fR
.ad
.RS 16n
The \fIpath\fR or \fItimes\fR argument points to an illegal address. For
\fBfutimesat()\fR, \fIpath\fR might have the value \fINULL\fR if the
\fIfildes\fR argument refers to a valid open file descriptor.
.RE

.sp
.ne 2
.na
\fB\fBEINTR\fR\fR
.ad
.RS 16n
A signal was caught during the execution of the \fButimes()\fR,
\fBfutimesat()\fR, \fBfutimens()\fR, or \fButimensat()\fR functions.
.RE

.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 16n
The number of microseconds specified in one or both of the \fBtimeval\fR
structures pointed to by \fItimes\fR was greater than or equal to 1,000,000 or
less than 0. The number of nanoseconds specified in one or both of the
\fBtimespec_t\fR structures pointed to by \fInstimes\fR was greater than or
equal to 1,000,000,000 or less than 0.
.RE

.sp
.ne 2
.na
\fB\fBEIO\fR\fR
.ad
.RS 16n
An I/O error occurred while reading from or writing to the file system.
.RE

.sp
.ne 2
.na
```



```
\fB\fBELOOP\fR\fR
.ad
.RS 16n
Too many symbolic links were encountered in resolving \fIpath\fR.
.RE

.sp
.ne 2
.na
\fB\fBENAMETOOLONG\fR\fR
.ad
.RS 16n
The length of the \fIpath\fR argument exceeds {\fIPATH_MAX\fR} or a pathname
component is longer than {\fINAME_MAX\fR}.
.RE

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

.sp
.ne 2
.na
\fB\fBENOENT\fR\fR
.ad
.RS 16n
A component of \fIpath\fR does not name an existing file or \fIpath\fR is an
empty string.
.RE

.sp
.ne 2
.na
\fB\fBENOTDIR\fR\fR
.ad
.RS 16n
A component of the path prefix is not a directory or the \fIpath\fR argument is
relative and the \fIfildes\fR argument is not \fBAT_FDCWD\fR or does not refer
to a valid directory.
.RE

.sp
.ne 2
.na
\fB\fBEPERM\fR\fR
.ad
.RS 16n
The \fItimes\fR argument is not a null pointer and the calling process's
effective user \fBID\fR has write access to the file but does not match the
owner of the file and the calling process does not have the appropriate
privileges.
.RE

.sp
.ne 2
.na
\fB\fBEROFS\fR\fR
.ad
.RS 16n
The file system containing the file is read-only.
```

```
.RE

.sp
.LP
The \fButimes()\fR, \fBfutimesat()\fR, and \fButimensat()\fR functions may fail
if:
.sp
.ne 2
.na
\fB\fBENAMETOOLONG\fR\fR
.ad
.RS 16n
Path name resolution of a symbolic link produced an intermediate result whose
length exceeds {\fIPATH_MAX\fR}.
.RE

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

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

.sp
.LP
For \fButimes()\fR, \fButimensat()\fR and \fBfutimensat()\fR, see \fBstandards\f
.SH SEE ALSO
.LP
\fBfutimens\fR(2), \fBstat\fR(2), \fButime\fR(2), \fBattributes\fR(5),
\fBfsattr\fR(5), \fBstandards\fR(5)
```

```
**********************************************************
    6147 Fri Jan 11 21:14:02 2019
new/usr/src/man/man3c/pthread_getschedparam.3c
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
.\"
.\" 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 1991, 1992, 1994, The X/Open Company Ltd.
.\" Copyright (c) 2001, The IEEE and The Open Group.  All Rights Reserved.
.\" Copyright (c) 2008, Sun Microsystems, Inc.  All Rights Reserved.
.\"
.TH PTHREAD_GETSCHEDPARAM 3C "Apr 1, 2008"
.SH NAME
pthread_getschedparam, pthread_setschedparam \- access dynamic thread
scheduling parameters
.SH SYNOPSIS
.LP
.nf
cc -mt [ \fIflag\fR... ] \fIfile\fR... -lpthread [ \fIlibrary\fR... ]
#include <pthread.h>

\fBint\fR \fBpthread_getschedparam\fR(\fBpthread_t\fR \fIthread\fR, \fBint *rest
     \fBstruct sched_param *restrict\fR \fIparam\fR);
```



```
.fi

.LP
.nf
\fBint\fR \fBpthread_setschedparam\fR(\fBpthread_t\fR \fIthread\fR, \fBint\fR \f
     \fBconst struct sched_param *\fR\fIparam\fR);
.fi

.SH DESCRIPTION
.sp
.LP
The \fBpthread_getschedparam()\fR and \fBpthread_setschedparam()\fR functions
allow the scheduling policy and scheduling parameters of individual threads
within a multithreaded process to be retrieved and set. Supported policies are
:
.sp
.ne 2
.na
\fB\fBSCHED_OTHER\fR\fR
.ad
.RS 15n
traditional time-sharing scheduling class
.RE

.sp
.ne 2
.na
\fB\fBSCHED_FIFO\fR\fR
.ad
.RS 15n
real-time class: run to completion
.RE

.sp
.ne 2
.na
\fB\fBSCHED_RR\fR\fR
.ad
.RS 15n
real-time class: round-robin
.RE

.sp
.ne 2
.na
\fB\fBSCHED_IA\fR\fR
.ad
.RS 15n
interactive time-sharing class
.RE

.sp
.ne 2
.na
\fB\fBSCHED_FSS\fR\fR
.ad
.RS 15n
fair-share scheduling class
.RE

.sp
.ne 2
.na
\fB\fBSCHED_FX\fR\fR
.ad
.RS 15n
```

```
124 fixed priority scheduling class
125 .RE

127 .sp
128 .LP
129 See \fBpthreads\fR(5). The affected scheduling parameter is the
130 \fIsched_priority\fR member of the \fBsched_param\fR structure.
131 .sp
132 .LP
133 The \fBpthread_getschedparam()\fR function retrieves the scheduling policy and
134 scheduling parameters for the thread whose thread \fBID\fR is given by
135 \fIthread\fR and stores those values in \fIpolicy\fR and \fIparam\fR,
136 respectively. The priority value returned from \fBpthread_getschedparam()\fR is
137 the value specified by the most recent \fBpthread_setschedparam()\fR or
138 \fBpthread_create()\fR call affecting the target thread, and does not reflect
139 any temporary adjustments to its priority as a result of any priority
140 inheritance or ceiling functions. The \fBpthread_setschedparam()\fR function
141 sets the scheduling policy and associated scheduling parameters for the thread
142 whose thread  \fBID\fR is given by \fIthread\fR to the policy and associated
143 parameters provided in \fIpolicy\fR and \fIparam\fR, respectively.
144 .sp
145 .LP
146 If the \fBpthread_setschedparam()\fR function fails, no scheduling parameters
147 will be changed  for the target thread.
148 .SH RETURN VALUES
150 .sp
149 .LP
150 If successful, the \fBpthread_getschedparam()\fR and
151 \fBpthread_setschedparam()\fR functions return  \fB0\fR. Otherwise, an error
152 number is returned to indicate the error.
153 .SH ERRORS
156 .sp
154 .LP
155 The \fBpthread_getschedparam()\fR and \fBpthread_setschedparam()\fR functions
158 The \fBpthread_getschedparam()\fR and \fBpthread_gsetschedparam()\fR functions
156 will fail if:
157 .sp
158 .ne 2
159 .na
160 \fB\fBESRCH\fR\fR
161 .ad
162 .RS 9n
163 The value specified by \fIthread\fR does not refer to an existing thread.
164 .RE

166 .sp
167 .LP
168 The \fBpthread_setschedparam()\fR function will fail if:
169 .sp
170 .ne 2
171 .na
172 \fB\fBEINVAL\fR\fR
173 .ad
174 .RS 10n
175 The value specified by \fIpolicy\fR or one of the scheduling parameters
176 associated with the scheduling policy \fIpolicy\fR is invalid.
177 .RE

179 .sp
180 .ne 2
181 .na
182 \fB\fBEPERM\fR\fR
183 .ad
184 .RS 10n
185 The caller does not have the appropriate permission to set either the
186 scheduling parameters or the scheduling policy of the specified thread.
```



```
187 .RE

189 .SH ATTRIBUTES
193 .sp
190 .LP
191 See \fBattributes\fR(5) for descriptions of the following attributes:
192 .sp

194 .sp
195 .TS
196 box;
197 c | c
198 l | l .
199 ATTRIBUTE TYPE  ATTRIBUTE VALUE
200 _
201 Interface Stability     Committed
202 _
203 MT-Level        MT-Safe
204 _
205 Standard        See \fBstandards\fR(5).
206 .TE

208 .SH SEE ALSO
213 .sp
209 .LP
210 \fBpthread_attr_init\fR(3C), \fBsched_getparam\fR(3C),
211 \fBsched_get_priority_max\fR(3C), \fBsched_get_priority_max\fR(3C),
212 \fBsched_get_priority_min\fR(3C), \fBsched_setparam\fR(3C),
213 \fBsched_getscheduler\fR(3C), \fBsched_setscheduler\fR(3C),
214 \fBattributes\fR(5), \fBpthreads\fR(5), \fBstandards\fR(5)
```

```
**********************************************************
    6111 Fri Jan 11 21:14:02 2019
new/usr/src/man/man3c/pthread_rwlock_rdlock.3c
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
.\"
.\" 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 1991, 1992, 1994, The X/Open Company Ltd.
.\" Copyright (c) 2001, The IEEE and The Open Group.  All Rights Reserved.
.\" Copyright (c) 2008, Sun Microsystems, Inc.  All Rights Reserved.
.\"
.TH PTHREAD_RWLOCK_RDLOCK 3C "Mar 23, 2005"
.SH NAME
pthread_rwlock_rdlock, pthread_rwlock_tryrdlock \- lock or attempt to lock
read-write lock object for reading
.SH SYNOPSIS
.LP
.nf
cc -mt [ \fIflag\fR... ] \fIfile\fR... -lpthread [ \fIlibrary\fR... ]
#include <pthread.h>

\fBint\fR \fBpthread_rwlock_rdlock\fR(\fBpthread_rwlock_t\fR \fI*rwlock\fR);
.fi
```



```
.LP
.nf
\fBint\fR \fBpthread_rwlock_tryrdlock\fR(\fBpthread_rwlock_t\fR \fI*rwlock\fR);
.fi

.SH DESCRIPTION
.sp
.LP
The \fBpthread_rwlock_rdlock()\fR function applies a read lock to the
read-write lock referenced by \fIrwlock\fR.  The calling thread acquires the
read lock if a writer does not hold the lock and there are no writers blocked
on the lock.
.sp
.LP
The calling thread does not acquire the lock if a writer holds the lock or if
writers of higher or equal priority are blocked on the lock; otherwise, the
calling thread acquires the lock. If the read lock is not acquired, the calling
thread blocks until it can acquire the lock.
.sp
.LP
A thread can hold multiple concurrent read locks on \fIrwlock\fR (that is,
successfully call the \fBpthread_rwlock_rdlock()\fR function \fIn\fR times). If
so, the thread must perform matching unlocks (that is, it must call the
\fBpthread_rwlock_unlock()\fR function \fIn\fR times).
.sp
.LP
The maximum number of concurrent read locks that a thread can hold on one
read-write lock is currently set at 100,000, though this number could change in
a future release. There is no imposed limit on the number of different threads
that can apply a read lock to one read-write lock.
.sp
.LP
The \fBpthread_rwlock_tryrdlock()\fR function applies a read lock like the
\fBpthread_rwlock_rdlock()\fR function, with the exception that the function
fails if the equivalent \fBpthread_rwlock_rdlock()\fR call would have blocked
the calling thread.  In no case will the \fBpthread_rwlock_tryrdlock()\fR
function ever block. It always either acquires the lock or fails and returns
function ever bloc. It always either acquires the lock or fails and returns
immediately.
.sp
.LP
Results are undefined if any of these functions are called with an
uninitialized read-write lock.
.sp
.LP
If a signal is delivered to a thread waiting for a read-write lock for reading,
upon return from the signal handler the thread resumes waiting for the
read-write lock for reading as if it was not interrupted.
.SH RETURN VALUES
.sp
.LP
If successful, the \fBpthread_rwlock_rdlock()\fR function returns \fB0\fR.
Otherwise, an error number is returned to indicate the error.
.sp
.LP
The \fBpthread_rwlock_tryrdlock()\fR function returns \fB0\fR if the lock for
reading on the read-write lock object referenced by \fIrwlock\fR is acquired.
Otherwise an error number  is returned to indicate the error.
.SH ERRORS
.sp
.LP
The \fBpthread_rwlock_rdlock()\fR and \fBpthread_rwlock_tryrdlock()\fR
functions will fail if:
.sp
.ne 2
```

```
.na
\fB\fBEAGAIN\fR\fR
.ad
.RS 10n
The read lock could not be acquired because the maximum number of read locks by
the current thread for \fIrwlock\fR has been exceeded.
.RE

.sp
.LP
The \fBpthread_rwlock_rdlock()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBEDEADLK\fR\fR
.ad
.RS 11n
The current thread already owns the read-write lock for writing.
.RE

.sp
.LP
The \fBpthread_rwlock_tryrdlock()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBEBUSY\fR\fR
.ad
.RS 9n
The read-write lock could not be acquired for reading because a writer holds
the lock or a writer with the appropriate priority was blocked on it.
.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
_
MT-Level        MT-Safe
.TE

.SH SEE ALSO
.sp
.LP
\fBpthread_rwlock_init\fR(3C), \fBpthread_rwlock_wrlock\fR(3C),
\fBpthread_rwlockattr_init\fR(3C), \fBpthread_rwlock_unlock\fR(3C),
\fBattributes\fR(5), \fBstandards\fR(5)
```

```
**********************************************************
    6386 Fri Jan 11 21:14:02 2019
new/usr/src/man/man3c/pthread_rwlock_timedwrlock.3c
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
.\"
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for
.\" permission to reproduce portions of its copyrighted documentation.
.\" Original documentation from The Open Group can be obtained online at
.\" http://www.opengroup.org/bookstore/.
.\"
.\" The Institute of Electrical and Electronics Engineers and The Open
.\" Group, have given us permission to reprint portions of their
.\" documentation.
.\"
.\" In the following statement, the phrase ``this text'' refers to portions
.\" of the system documentation.
.\"
.\" Portions of this text are reprinted and reproduced in electronic form
.\" in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition,
.\" Standard for Information Technology -- Portable Operating System
.\" Interface (POSIX), The Open Group Base Specifications Issue 6,
.\" Copyright (C) 2001-2004 by the Institute of Electrical and Electronics
.\" Engineers, Inc and The Open Group.  In the event of any discrepancy
.\" between these versions and the original IEEE and The Open Group
.\" Standard, the original IEEE and The Open Group Standard is the referee
.\" document.  The original Standard can be obtained online at
.\" http://www.opengroup.org/unix/online.html.
.\"
.\" This notice shall appear on any product containing this material.
.\"
.\" The contents of this file are subject to the terms of the
.\" Common Development and Distribution License (the "License").
.\" You may not use this file except in compliance with the License.
.\"
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
.\" or http://www.opensolaris.org/os/licensing.
.\" See the License for the specific language governing permissions
.\" and limitations under the License.
.\"
.\" When distributing Covered Code, include this CDDL HEADER in each
.\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
.\" If applicable, add the following below this CDDL HEADER, with the
.\" fields enclosed by brackets "[]" replaced with your own identifying
.\" information: Portions Copyright [yyyy] [name of copyright owner]
.\"
.\"
.\" Copyright (c) 2001, The IEEE and The Open Group.  All Rights Reserved.
.\" Portions Copyright (c) 2004, Sun Microsystems, Inc. All Rights Reserved.
.\"
.TH PTHREAD_RWLOCK_TIMEDWRLOCK 3C "Jan 30, 2004"
.SH NAME
pthread_rwlock_timedwrlock, pthread_rwlock_reltimedwrlock_np \- lock a
read-write lock for writing
.SH SYNOPSIS
.LP
.nf
cc \fB-mt\fR [ \fIflag\fR... ] \fIfile\fR... [ \fIlibrary\fR... ]
#include <pthread.h>
#include <time.h>

\fBint\fR \fBpthread_rwlock_timedwrlock\fR(\fBpthread_rwlock_t *restrict\fR \fIr
     \fBconst struct timespec *restrict\fR \fIabs_timeout\fR);
```



```
.fi

.LP
.nf
\fBint\fR \fBpthread_rwlock_reltimedwrlock_np\fR(\fBpthread_rwlock_t *restrict\f
     \fBconst struct timespec *restrict\fR \fIrel_timeout\fR);
.fi

.SH DESCRIPTION
.sp
.LP
The \fBpthread_rwlock_timedwrlock()\fR function applies a write lock to the
read-write lock referenced by \fIrwlock\fR as in the
\fBpthread_rwlock_wrlock\fR(3C) function. If the lock cannot be acquired
without waiting for other threads to unlock the lock, this wait will be
terminated when the specified timeout expires. The timeout expires when the
absolute time specified by \fIabs_timeout\fR passes, as measured by the
\fBCLOCK_REALTIME\fR clock (that is, when the value of that clock equals or
exceeds \fIabs_timeout\fR), or if the absolute time specified by
\fIabs_timeout\fR has already been passed at the time of the call.
.sp
.LP
The \fBpthread_rwlock_reltimedwrlock_np()\fR function is identical to the
\fBpthread_rwlock_timedwrlock()\fR function, except that the timeout is
specified as a relative time interval. The timeout expires when the time
interval specified by \fIrel_timeout\fR passes, as measured by the
\fBCLOCK_REALTIME\fR clock, or if the time interval specified by
\fIrel_timeout\fR is negative at the time of the call.
.sp
.LP
The resolution of the timeout is the resolution of the \fBCLOCK_REALTIME\fR
clock. The \fBtimespec\fR data type is defined in the <\fBtime.h\fR> header.
Under no circumstances does either function fail with a timeout if the lock can
be acquired immediately. The validity of the \fIabs_timeout\fR parameter need
be acquired immediately. The validity of the a\fIbs_timeout\fR parameter need
not be checked if the lock can be immediately acquired.
.sp
.LP
If a signal that causes a signal handler to be executed is delivered to a
thread blocked on a read- write lock with a call to
\fBpthread_rwlock_timedwrlock()\fR or \fBpthread_rwlock_reltimedwrlock_np()\fR,
upon return from the signal handler the thread resumes waiting for the lock as
if it was not interrupted.
.sp
.LP
The calling thread can deadlock if at the time the call is made it holds the
read-write lock. The results are undefined if this function is called with an
uninitialized read-write lock.
.SH RETURN VALUES
.sp
.LP
The \fBpthread_rwlock_timedwrlock()\fR and
\fBpthread_rwlock_reltimedwrlock_np()\fR functions return 0 if the lock for
writing on the read-write lock object referenced by \fIrwlock\fR is acquired.
Otherwise, an error number is returned to indicate the error.
.SH ERRORS
.sp
.LP
The \fBpthread_rwlock_timedwrlock()\fR and
\fBpthread_rwlock_reltimedwrlock_np()\fR functions will fail if:
.sp
.ne 2
.na
\fB\fBETIMEDOUT\fR\fR
.ad
.RS 13n
```

```
121 The lock could not be acquired before the specified timeout expired.
122 .RE

124 .sp
125 .LP
126 The \fBpthread_rwlock_timedwrlock()\fR and
127 \fBpthread_rwlock_reltimedwrlock_np()\fR functions may fail if:
128 .sp
129 .ne 2
130 .na
131 \fB\fBEDEADLK\fR\fR
132 .ad
133 .RS 11n
134 The calling thread already holds the rwlock.
135 .RE

137 .sp
138 .ne 2
139 .na
140 \fB\fBEINVAL\fR\fR
141 .ad
142 .RS 11n
143 The value specified by \fIrwlock\fR does not refer to an initialized read-write
144 lock object, or the timeout nanosecond value is less than zero or greater than
145 or equal to 1,000 million.
146 .RE

148 .SH ATTRIBUTES
152 .sp
149 .LP
150 See \fBattributes\fR(5) for descriptions of the following attributes:
151 .sp

153 .sp
154 .TS
155 box;
156 c | c
157 l | l .
158 ATTRIBUTE TYPE  ATTRIBUTE VALUE
159 _
160 Interface Stability     See below.
161 _
162 MT-Level        MT-Safe
163 .TE

165 .sp
166 .LP
167 The \fBpthread_rwlock_timedwrlock()\fR function is Standard. The
168 \fBpthread_rwlock_reltimedwrlock_np()\fR function is Stable.
169 .SH SEE ALSO
174 .sp
170 .LP
171 \fBpthread_rwlock_destroy\fR(3C), \fBpthread_rwlock_rdlock\fR(3C),
172 \fBpthread_rwlock_timedrdlock\fR(3C), \fBpthread_rwlock_trywrlock\fR(3C),
173 \fBpthread_rwlock_unlock\fR(3C), \fBpthread_rwlock_wrlock\fR(3C),
174 \fBattributes\fR(5), \fBstandards\fR(5)
```

```
**********************************************************
   14417 Fri Jan 11 21:14:02 2019
new/usr/src/man/man3c/select.3c
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
```

```
.\"
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for
.\" permission to reproduce portions of its copyrighted documentation.
.\" Original documentation from The Open Group can be obtained online at
.\" http://www.opengroup.org/bookstore/.
.\"
.\" The Institute of Electrical and Electronics Engineers and The Open
.\" Group, have given us permission to reprint portions of their
.\" documentation.
.\"
.\" In the following statement, the phrase ‘‘this text’’ refers to portions
.\" of the system documentation.
.\"
.\" Portions of this text are reprinted and reproduced in electronic form
.\" in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition,
.\" Standard for Information Technology -- Portable Operating System
.\" Interface (POSIX), The Open Group Base Specifications Issue 6,
.\" Copyright (C) 2001-2004 by the Institute of Electrical and Electronics
.\" Engineers, Inc and The Open Group.  In the event of any discrepancy
.\" between these versions and the original IEEE and The Open Group
.\" Standard, the original IEEE and The Open Group Standard is the referee
.\" document.  The original Standard can be obtained online at
.\" http://www.opengroup.org/unix/online.html.
.\"
.\" This notice shall appear on any product containing this material.
.\"
.\" The contents of this file are subject to the terms of the
.\" Common Development and Distribution License (the "License").
.\" You may not use this file except in compliance with the License.
.\"
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
.\" or http://www.opensolaris.org/os/licensing.
.\" See the License for the specific language governing permissions
.\" and limitations under the License.
.\"
.\" When distributing Covered Code, include this CDDL HEADER in each
.\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
.\" If applicable, add the following below this CDDL HEADER, with the
.\" fields enclosed by brackets "[]" replaced with your own identifying
.\" information: Portions Copyright [yyyy] [name of copyright owner]
.\"
.\"
.\" Portions Copyright (c) 1992, X/Open Company Limited.  All Rights Reserved.
.\" Copyright (c) 2004, Sun Microsystems, Inc. All Rights Reserved.
.\"
.Dd "Dec 28, 2016"
.Dt SELECT 3C
.Os
.Sh NAME
.Nm select ,
.Nm pselect ,
.Nm FD_SET ,
.Nm FD_CLR ,
.Nm FD_ISSET ,
.Nm FD_ZERO
.Nd synchronous I/O multiplexing
.Sh SYNOPSIS
.In sys/time.h
```



```
.Ft int
.Fo select
.Fa "int nfds"
.Fa "fd_set *restrict readfds"
.Fa "fd_set *restrict writefds"
.Fa "fd_set *restrict errorfds"
.Fa "struct timeval *restrict timeout"
.Fc
.Ft int
.Fo pselect
.Fa "int nfds"
.Fa "fd_set *restrict readfds"
.Fa "fd_set *restrict writefds"
.Fa "fd_set *restrict errorfds"
.Fa "const struct timespec *restrict timeout"
.Fa "const sigset_t *restrict sigmask"
.Fc
.Ft void
.Fo FD_SET
.Fa "int fd"
.Fa "fd_set *fdset"
.Fc
.Ft void
.Fo FD_CLR
.Fa "int fd"
.Fa "fd_set *fdset"
.Fc
.Ft int
.Fo FD_ISSET
.Fa "int fd"
.Fa "fd_set *fd_set"
.Fc
.Ft void
.Fo FD_ZERO
.Fa "fd_set *fdset"
.Fc
.Sh DESCRIPTION
The
.Fn pselect
function examines the file descriptor sets whose addresses
are passed in the
.Fa readfds ,
.Fa writefds ,
and
.Fa errorfds
parameters to see if some of their descriptors are ready for reading,
are ready for writing, or have an exceptional condition pending,
respectively.
.Pp
The
.Fn select
function is equivalent to the
.Fn pselect
function, except as follows:
.Bl -bullet
.It
For the
.Fn select
function, the timeout period is given in seconds and
microseconds in an argument of type
.Vt struct timeval ,
whereas for the
.Fn pselect
function the timeout period is given in seconds and nanoseconds
in an argument of type
.Vt struct timespec
```

```
.It
The
.Fn select
function has no
.Fa sigmask
argument.
It behaves as
.Fn pselect
does when
.Fa sigmask
is a null pointer.
.It
Upon successful completion, the
.Fn select
function might modify the object pointed to by the
.Fa Itimeout
argument.
.El
.Pp
The
.Fn select
and
.Fn pselect
functions support regular files, terminal and pseudo-terminal devices,
STREAMS-based files, FIFOs, pipes, and sockets.
The behavior of
.Fn select
and
.Fn pselect
on file descriptors that refer to other types of file is unspecified.
.Pp
The
.Fa nfds
argument specifies the range of file descriptors to be tested.
The first
.Fa nfds
descriptors are checked in each set; that is, the
descriptors from zero through
.Dq Li nfds - 1
in the descriptor sets are examined.
.Pp
If the
.Fa readfds
.Fa readfs
argument is not a null pointer, it points to an object of
type
.Vt fd_set
that on input specifies the file descriptors to be checked
for being ready to read, and on output indicates which file descriptors are
ready to read.
.Pp
If the
.Fa writefds
.Fa writefs
argument is not a null pointer, it points to an object of
type
.Vt fd_set
that on input specifies the file descriptors to be checked
for being ready to write, and on output indicates which file descriptors are
ready to write.
.Pp
If the
.Fa errorfds
argument is not a null pointer, it points to an object of
type
.Vt fd_set
```



```
that on input specifies the file descriptors to be checked
for error conditions pending, and on output indicates which file descriptors
have error conditions pending.
.Pp
Upon successful completion, the objects pointed to by the
.Fa readfds ,
.Fa writefds ,
.Fa readfs ,
.Fa writefs ,
and
.Fa errorfds
arguments are modified to indicate which file descriptors are ready for reading,
ready for writing, or have an error condition pending, respectively, and return
the total number of ready descriptors in all the output sets.
For each file descriptor less than
.Fa nfds ,
the corresponding bit will be set on successful completion if it was set on
input and the associated condition is true for that file descriptor.
.Pp
If none of the selected descriptors are ready for the requested operation, the
.Fn select
or
.Fn pselect
function blocks until at least one of the
requested operations becomes ready, until the timeout occurs, or until
interrupted by a signal.
The
.Fa timeout
parameter controls how long the
.Fn select
or
.Fn pselect
function takes before timing out.
If the
.Fa timeout
parameter is not a null pointer, it specifies a maximum interval
to wait for the selection to complete.
If the specified time interval expires without any requested operation becoming
ready, the function returns.
If the
.Fa timeout
parameter is a null pointer, then the call to
.Fn select
or
.Fn pselect
blocks indefinitely until at least one descriptor meets the
specified criteria.
To effect a poll, the
.Fa timeout
parameter should not be a null pointer, and should point to a zero-valued
.Vt timespec
structure.
.Pp
The use of a
.Fa timeout
does not affect any pending timers set up by
.Xr alarm 2 ,
.Xr ualarm 3C ,
or
.Xr setitimer 2 .
.Pp
If
.Fa sigmask
is not a null pointer, then the
.Fn pselect
function replaces the signal mask of the process by the set of signals pointed
```

```
253 to by
254 .Fa sigmask
255 before examining the descriptors, and restores the signal mask of
256 the process before returning.
257 .Pp
258 A descriptor is considered ready for reading when a call to an input function
259 with
260 .Dv O_NONBLOCK
261 clear would not block, whether or not the function would
262 transfer data successfully.
263 (The function might return data, an end-of-file indication, or an error other
264 than one indicating that it is blocked, and in each of these cases the
265 descriptor will be considered ready for reading.)
266 .Pp
267 A descriptor is considered ready for writing when a call to an output function
268 with
269 .Dv O_NONBLOCK
270 clear would not block, whether or not the function would
271 transfer data successfully.
272 .Pp
273 If a socket has a pending error, it is considered to have an exceptional
274 condition pending.
275 Otherwise, what constitutes an exceptional condition is file type-specific.
276 For a file descriptor for use with a socket, it is protocol-specific except as
277 noted below.
278 For other file types, if the operation is meaningless for a particular file
279 type,
280 .Fn select
281 or
282 .Fn pselect
283 indicates that the descriptor is ready for read or write operations and
284 indicates that the descriptor has no exceptional condition pending.
285 .Pp
286 If a descriptor refers to a socket, the implied input function is the
287 .Xr recvmsg 3XNET
288 function with parameters requesting normal and ancillary
289 data, such that the presence of either type causes the socket to be marked as
290 readable.
291 The presence of out-of-band data is checked if the socket option
292 .Dv SO_OOBINLINE
293 has been enabled, as out-of-band data is enqueued with
294 normal data.
295 If the socket is currently listening, then it is marked as readable if an
296 incoming connection request has been received, and a call to the accept function
297 completes without blocking.
298 .Pp
299 If a descriptor refers to a socket, the implied output function is the
300 .Xr sendmsg 3XNET
301 function supplying an amount of normal data equal to the
302 current value of the
303 .Dv SO_SNDLOWAT
304 option for the socket.
305 If a non-blocking call to the connect function has been made for a socket, and
306 the connection attempt has either succeeded or failed leaving a pending error,
307 the socket is marked as writable.
308 .Pp
309 A socket is considered to have an exceptional condition pending if a receive
310 operation with
311 .Dv O_NONBLOCK
312 clear for the open file description and with the
313 .Dv MSG_OOB
314 flag set would return out-of-band data without blocking.
315 (It is protocol-specific whether the
316 .Dv MSG_OOB
317 flag would be used to read out-of-band data.)
318 A socket will also be considered to have an exceptional condition pending if an
```



```
319 out-of-band data mark is present in the receive queue.
320 .Pp
321 A file descriptor for a socket that is listening for connections will indicate
322 that it is ready for reading, when connections are available.
323 A file descriptor for a socket that is connecting asynchronously will indicate
324 that it is ready for writing, when a connection has been established.
325 .Pp
326 Selecting true for reading on a socket descriptor upon which a
327 .Xr listen 3XNET
328 call has been performed indicates that a subsequent
329 .Xr accept 3XNET
330 call on that descriptor will not block.
331 .Pp
332 If the
333 .Fa timeout
334 argument is not a null pointer, it points to an object of type
335 .Vt struct timeval
336 that specifies a maximum interval to wait for the
337 selection to complete.
338 If the
339 .Fa timeout
340 argument points to an object of type
341 .Vt struct timeval
342 whose members are 0,
343 .Fn select
344 does not block.
345 If the
346 .Fa timeout
347 argument is a null pointer,
348 .Fn select
349 blocks until an event causes one of the masks to be returned with a valid
350 (non-zero) value.
351 If the time limit expires before any event occurs that would cause one of the
352 masks to be set to a non-zero value,
353 .Fn select
354 completes successfully and returns 0.
355 .Pp
356 If the
357 .Fa readfds ,
357 .Fa readfs ,
358 .Fa writefds ,
359 and
360 .Fa errorfds
361 arguments are all null pointers and the
362 .Fa timeout
363 argument is not a null pointer,
364 .Fn select
365 or
366 .Fn pselect
367 blocks for the time specified, or until interrupted by a
368 signal.
369 If the
370 .Fa readfds ,
371 .Fa writefds ,
372 and
373 .Fa errorfds
374 arguments are all null pointers and the
375 .Fa timeout
376 argument is a null pointer,
377 .Fn select
378 blocks until interrupted by a signal.
379 .Pp
380 File descriptors associated with regular files always select true for ready to
381 read, ready to write, and error conditions.
382 .Pp
383 On failure, the objects pointed to by the
```

```
.Fa readfds ,
.Fa writefds ,
and
.Fa errorfds
arguments are not modified.
If the timeout interval expires without the specified condition being true for
any of the specified file descriptors, the objects pointed to by the
.Fa readfds ,
.Fa writefds ,
and
.Fa errorfds
arguments have all bits set to 0.
.Pp
File descriptor masks of type
.Vt fd_set
can be initialized and tested with the macros
.Fn FD_CLR ,
.Fn FD_ISSET ,
.Fn FD_SET ,
and
.Fn FD_ZERO .
.Bl -tag -width indent
.It Fn FD_CLR "fd" "&fdset"
Clears the bit for the file descriptor
.Fa fd
in the file descriptor set
.Fa fdset .
.It Fn FD_ISSET "fd" "&fdset"
Returns a non-zero value if the bit for the file descriptor
.Fa fd
is set in
the file descriptor set pointed to by
.Fa fdset ,
and 0 otherwise.
.It Fn FD_SET "fd" "&fdset"
Sets the bit for the file descriptor
.Fa fd
in the file descriptor set
.Fa fdset
.It Fn FD_ZERO "&fdset"
Initializes the file descriptor set \fIfdset\fR to have zero bits for all file
descriptors.
.El
.Pp
The behavior of these macros is undefined if the
.Fa fd
argument is less than 0 or greater than or equal to
.Dv FD_SETSIZE ,
or if
.Fa fd
is not a valid file descriptor, or if any of the arguments are expressions
with side effects.
.Sh RETURN VALUES
On successful completion,
.Fn select
and
.Fn pselect
return the total
number of bits set in the bit masks.
Otherwise,
.Sy 1
is returned and
.Dv errno
is set to indicate the error.
.Pp
The
```



```
.Fn FD_CLR ,
.Fn FD_SET ,
and
.Fn FD_ZERO ,
macros return no value.
The
.Fn FD_ISSET
macro returns a non-zero value if the bit for the file
descriptor
.Fa fd
is set in the file descriptor set pointed to by
.Fa fdset ,
and
.Sy 0
otherwise.
.Sh ERRORS
The
.Fn select
and
.Fn pselect
functions will fail if:
.Bl -tag -width indent
.It Er EBADF
One or more of the file descriptor sets specified a file descriptor that is not
a valid open file descriptor.
.It Er EINTR
The function was interrupted before any of the selected events occurred and
before the timeout interval expired.
.Pp
If
.Dv SA_RESTART
has been set for the interrupting signal, it is implementation-dependent whether
.Fn select
restarts or returns with
.Dv EINTR
.It Er EINVAL
An invalid timeout interval was specified.
.It Er EINVAL
The
.Fa nfds
argument is less than 0 or greater than
.Dv FD_SETSIZE .
.It Er EINVAL
One of the specified file descriptors refers to a STREAM or multiplexer that is
linked (directly or indirectly) downstream from a multiplexer.
.It Er EINVAL
A component of the pointed-to time limit is outside the acceptable range:
.Dv t_sec
must be between 0 and 10^8, inclusive.
.Dv t_usec
must be greater than or equal to 0, and less than 10^6.
.El
.Sh USAGE
The
.Xr poll 2
function is preferred over this function.
.Pp
The use of a timeout does not affect any pending timers set up by
.Xr alarm 2 ,
.Xr ualarm 3C ,
or
.Xr setitimer 2 .
.Pp
On successful completion, the object pointed to by the
.Fa timeout
argument may be modified.
```

```
 516 .Sh INTERFACE STABILITY
 517 .Sy Standard
 518 .Sh MT Level
 519 .Sy MT-Safe
 520 .Sh SEE ALSO
 521 .Xr alarm 2 ,
 522 .Xr fcntl 2 ,
 523 .Xr poll 2 ,
 524 .Xr read 2 ,
 525 .Xr setitimer 2 ,
 526 .Xr write 2 ,
 527 .Xr ualarm 3C ,
 528 .Xr accept 3SOCKET ,
 529 .Xr listen 3SOCKET ,
 530 .Xr attributes 5 ,
 531 .Xr standards 5
```

```
**********************************************************
    6290 Fri Jan 11 21:14:02 2019
new/usr/src/man/man3contract/ct_dev_tmpl_set_aset.3contract
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
'\" te
.\" Copyright (c) 2007, Sun Microsystems, Inc. All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH CT_DEV_TMPL_SET_ASET 3CONTRACT "Aug 9, 2007"
.SH NAME
ct_dev_tmpl_set_aset, ct_dev_tmpl_get_aset, ct_dev_tmpl_set_minor,
ct_dev_tmpl_get_minor, ct_dev_tmpl_set_noneg, ct_dev_tmpl_clear_noneg,
ct_dev_tmpl_get_noneg \- device contract template functions
.SH SYNOPSIS
.LP
.nf
cc [ \fIflag\fR\&.\|.\|. ] \fIfile\fR\&.\|.\|. \fB-D_LARGEFILE64_SOURCE\fR \fB -
#include <libcontract.h>
#include <sys/contract/device.h>

int ct_dev_tmpl_set_aset(int fd, uint_t aset);
.fi

.LP
.nf
\fBint\fR \fBct_dev_tmpl_get_aset\fR(\fBint\fR \fIfd\fR, \fBuint_t *\fR\fIasetp\
.fi

.LP
.nf
\fBint\fR \fBct_dev_tmpl_set_minor\fR(\fBint\fR \fIfd\fR, \fBchar *\fR\fIminor\f
.fi

.LP
.nf
\fBint\fR \fBct_dev_tmpl_get_minor\fR(\fBint\fR \fIfd\fR, \fBchar *\fR\fIbuf\fR,
.fi

.LP
.nf
\fBint\fR \fBct_dev_tmpl_set_noneg\fR(\fBint\fR \fIfd\fR);
.fi

.LP
.nf
\fBint\fR \fBct_dev_tmpl_clear_noneg\fR(\fBint\fR \fIfd\fR);
.fi

.LP
.nf
\fBint\fR \fBct_dev_tmpl_get_noneg\fR(\fBint\fR \fIfd\fR, \fBuint_t *\fR\fInoneg
.fi

.SH PARAMETERS
.sp
.ne 2
.na
\fB\fIaset\fR\fR
.ad
.RS 11n
a bitset of one or more of device states
```



```
.RE

.sp
.ne 2
.na
\fB\fIasetp\fR\fR
.ad
.RS 11n
a pointer to a variable into which the current A-set is to be returned
.RE

.sp
.ne 2
.na
\fB\fIbuf\fR\fR
.ad
.RS 11n
a buffer into which the minor path is to be returned
.RE

.sp
.ne 2
.na
\fB\fIbuflenp\fR\fR
.ad
.RS 11n
a pointer to variable of type \fBsize_t\fR in which the size of the buffer
\fIbuf\fR is passed in. If the buffer is too small the size of the buffer
needed for a successful call is passed back to the caller.
.RE

.sp
.ne 2
.na
\fB\fIfd\fR\fR
.ad
.RS 11n
a file descriptor from an open of the device contract template file in the
contract filesystem (ctfs)
.RE

.sp
.ne 2
.na
\fB\fIminor\fR\fR
.ad
.RS 11n
the \fBdevfs\fR path (the \fB/devices\fR path without the "\fB/devices\fR"
prefix) of a minor which is to be the subject of a contract
.RE

.sp
.ne 2
.na
\fB\fInonegp\fR\fR
.ad
.RS 11n
a pointer to a \fIuint_t\fR variable for receiving the current setting of the
"nonnegotiable" term in the template
.RE

.SH DESCRIPTION
.sp
.LP
These functions read and write device contract terms and operate on device
contract template file descriptors obtained from the \fBcontract\fR(4)
```

```
filesystem (ctfs).
.sp
.LP
The \fBct_dev_tmpl_set_aset()\fR and \fBct_dev_tmpl_get_aset()\fR functions
write and read the "acceptable states" set (or A-set for short).  This is the
set of device states guaranteed by the contract. Any departure from these
states will result in the breaking of the contract and a delivery of a critical
contract event to the contract holder. The A-set value is a bitset of one or
more of the following device states: \fBCT_DEV_EV_ONLINE\fR,
\fBCT_DEV_EV_DEGRADED\fR, and \fBCT_DEV_EV_OFFLINE\fR.
.sp
.LP
The \fBct_dev_tmpl_set_minor()\fR and \fBct_dev_tmpl_get_minor()\fR functions
write and read the minor term (the device resource that is to be the subject of
the contract.) The value is a \fBdevfs\fR path to a device minor node (minus
the "\fB/devices\fR" prefix). For the \fBct_dev_tmpl_get_minor()\fR function, a
buffer at least \fBPATH_MAX\fR in size must be passed in. If the buffer is
smaller than \fBPATH_MAX\fR, then the minimum size of the buffer required
(\fBPATH_MAX\fR) for this function is passed back to the caller via the
\fIbuflenp\fR argument.
.sp
.LP
The \fBct_dev_tmpl_set_noneg()\fR and \fBct_dev_tmpl_get_noneg()\fR functions
write and read the nonnegotiable term. If this term is set, synchronous
negotiation events are automatically NACKed on behalf of the contract holder.
For \fBct_dev_tmpl_get_noneg()\fR, the variable pointed to by \fInonegp\fR is
set to 1 if the "noneg" term is set or to 0 otherwise.  The
\fBct_dev_tmpl_clear_noneg()\fR term clears the nonnegotiable term from a
template.
.SH RETURN VALUES
.sp
.LP
Upon successful completion, these functions return 0. Otherwise, they return a
non-zero error value.
.SH ERRORS
.sp
.LP
The \fBct_dev_tmpl_set_aset()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 10n
A template file descriptor or A-set is invalid
.RE

.sp
.LP
The \fBct_dev_tmpl_set_minor()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 10n
One or more arguments is invalid.
.RE

.sp
.ne 2
.na
\fB\fBENXIO\fR\fR
.ad
.RS 10n
The minor named by minor path does not exist.
```



```
.RE

.sp
.LP
The \fBct_dev_tmpl_set_noneg()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBEPERM\fR\fR
.ad
.RS 9n
A process lacks sufficient privilege to NACK a device state change.
.RE

.sp
.LP
The \fBct_dev_tmpl_get_aset()\fR and \fBct_dev_tmpl_get_minor()\fR functions
will fail if:
.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 10n
One or more arguments is invalid.
.RE

.sp
.ne 2
.na
\fB\fBENOENT\fR\fR
.ad
.RS 10n
The requested term is not set.
.RE

.sp
.LP
The \fBct_dev_tmpl_get_noneg()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 10n
One or more arguments is invalid.
.RE

.sp
.LP
The \fBct_dev_tmpl_get_minor()\fR function will fail if:
.sp
.ne 2
.na
\fB\fBEOVERFLOW\fR\fR
\fB\fBEOVEFLOW\fR\fR
.ad
.RS 12n
The supplied buffer is too small.
.RE

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

```
252 .sp
253 .TS
254 box;
255 c | c
256 l | l .
257 ATTRIBUTE TYPE  ATTRIBUTE VALUE
258 _
259 Interface Stability     Committed
260 _
261 MT-Level        Safe
262 .TE

264 .SH SEE ALSO
270 .sp
265 .LP
266 \fBlibcontract\fR(3LIB), \fBcontract\fR(4), \fBdevices\fR(4),
267 \fBattributes\fR(5), \fBlfcompile\fR(5)
```

```
**********************************************************
    2121 Fri Jan 11 21:14:02 2019
new/usr/src/man/man3curses/form_cursor.3curses
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
'\" te
.\" Copyright 1989 AT&T
.\" Portions Copyright (c) 1996, 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 FORM_CURSOR 3CURSES "Dec 31, 1996"
.SH NAME
form_cursor, pos_form_cursor \- position forms window cursor
.SH SYNOPSIS
.LP
.nf
\fBcc\fR [ \fIflag\fR... ] \fIfile\fR... \fB-lform\fR \fB -lcurses \fR [ \fIlibr
#include <form.h>

\fBint\fR \fBpos_form_cursor\fR(\fBFORM *\fR\fIform\fR);
.fi

.SH DESCRIPTION
.sp
.LP
\fBpos_form_cursor()\fR moves the form window cursor to the location required
by the form driver to resume form processing. This may be needed after the
application calls a \fBcurses\fR library I/O routine.
.SH RETURN VALUES
.sp
.LP
\fBpos_form_cursor()\fR returns one of the following:
.sp
.ne 2
.na
\fBE_OK\fR
.ad
.RS 18n
The function returned successfully.
Thefunction returned successfully.
.RE

.sp
.ne 2
.na
\fBE_SYSTEM_ERROR\fR
.ad
.RS 18n
System error.
.RE

.sp
.ne 2
.na
\fBE_BAD_ARGUMENT\fR
.ad
.RS 18n
An argument is incorrect.
.RE

.sp
.ne 2
```



```
.na
\fBE_NOT_POSTED\fR
.ad
.RS 18n
The form is not posted.
.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
_
MT-Level        Unsafe
.TE

.SH SEE ALSO
.sp
.LP
\fBcurses\fR(3CURSES), \fBforms\fR(3CURSES), \fBattributes\fR(5)
.SH NOTES
.sp
.LP
The header \fB<form.h>\fR automatically includes the headers  \fB<eti.h>\fR and
\fB<curses.h>\fR\&.
```

```
**********************************************************
    3164 Fri Jan 11 21:14:03 2019
new/usr/src/man/man3curses/form_field_buffer.3curses
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
'\" te
.\" Copyright 1989 AT&T
.\" Portions Copyright (c) 1996, 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 FORM_FIELD_BUFFER 3CURSES "Dec 31, 1996"
.SH NAME
form_field_buffer, set_field_buffer, field_buffer, set_field_status,
field_status, set_max_field \- set and get forms field attributes
.SH SYNOPSIS
.LP
.nf
\fBcc\fR [ \fIflag\fR... ] \fIfile\fR... \fB-lform\fR \fB -lcurses \fR [ \fIlibr
#include <form.h>



.LP
.nf
\fBint\fR \fBset_field_buffer\fR(\fBFIELD *\fR\fIfield\fR, \fBint\fR \fIbuf\fR,
.fi

.LP
.nf
\fBchar *\fR\fBfield_buffer\fR(\fBFIELD *\fR\fIfield\fR, \fBint\fR \fIbuf\fR);
.fi

.LP
.nf
\fBint\fR \fBset_field_status\fR(\fBFIELD *\fR\fIfield\fR, \fBint\fR \fIstatus\f
.fi

.LP
.nf
\fBint\fR \fBfield_status\fR(\fBFIELD *\fR\fIfield\fR);
.fi

.LP
.nf
\fBint\fR \fBset_max_field\fR(\fBFIELD *\fR\fIfield\fR, \fBint\fR \fImax\fR);
.fi

.SH DESCRIPTION
.sp
.LP
\fBset_field_buffer()\fR sets buffer \fIbuf\fR of \fIfield\fR to \fIvalue\fR.
Buffer 0 stores the displayed contents of the field. Buffers other than 0 are
application specific and not used by the \fBforms\fR library routines.
\fBfield_buffer()\fR returns the value of \fIfield\fR buffer \fIbuf\fR.
.sp
.LP
Every field has an associated status flag that is set whenever the contents of
field buffer 0 changes. \fBset_field_status()\fR sets the status flag of
\fIfield\fR to \fIstatus\fR. \fBfield_status()\fR returns the status of
\fIfield\fR.
.sp
.LP
\fBset_max_field()\fR sets a maximum growth on a dynamic field, or if
\fImax=\fR\fB0\fR turns off any maximum growth.
```



```
.SH RETURN VALUES
.sp
.LP
\fBfield_buffer()\fR returns \fINULL\fR on error.
.sp
.LP
\fBfield_status()\fR returns \fBTRUE\fR or \fBFALSE\fR.
.sp
.LP
\fBset_field_buffer()\fR, \fBset_field_status()\fR, and \fBset_max_field()\fR
return one of the following:
.sp
.ne 2
.na
\fBE_OK\fR
.ad
.RS 18n
The function returned successfully.
Thefunction returned successfully.
.RE

.sp
.ne 2
.na
\fBE_SYSTEM_ERROR\fR
.ad
.RS 18n
System error
.RE

.sp
.ne 2
.na
\fBE_BAD_ARGUMENT\fR
.ad
.RS 18n
An argument is incorrect.
.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
_
MT-Level        Unsafe
.TE

.SH SEE ALSO
.sp
.LP
\fBcurses\fR(3CURSES), \fBforms\fR(3CURSES), \fBattributes\fR(5)
.SH NOTES
.sp
.LP
The header \fB<form.h>\fR automatically includes the headers  \fB<eti.h>\fR and
\fB<curses.h>\fR\&.
```

```
**********************************************************
    3229 Fri Jan 11 21:14:03 2019
new/usr/src/man/man3curses/form_field_new.3curses
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
```

```
   1 '\" te
   2 .\" Copyright 1989 AT&T
   3 .\" Portions Copyright (c) 1996, Sun Microsystems, Inc. All Rights Reserved.
   4 .\" The contents of this file are subject to the terms of the Common Development
   5 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
   6 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
   7 .TH FORM_FIELD_NEW 3CURSES "Dec 31, 1996"
   8 .SH NAME
   9 form_field_new, new_field, dup_field, link_field, free_field \- create and
  10 destroy forms fields
  11 .SH SYNOPSIS
  12 .LP
  13 .nf
  14 \fBcc\fR [ \fIflag\fR... ] \fIfile\fR... \fB-lform\fR \fB -lcurses \fR [ \fIlibr
  15 #include <form.h>

  17 \fBFIELD *\fR\fBnew_field\fR(\fBint\fR \fIr\fR, \fBint\fR \fIc\fR, \fBint\fR \fI
  18 .fi

  20 .LP
  21 .nf
  22 \fBFIELD *\fR\fBdup_field\fR(\fBFIELD *\fR\fIfield\fR, \fBint\fR \fIfrow\fR, \fB
  23 .fi

  25 .LP
  26 .nf
  27 \fBFIELD *\fR\fBlink_field\fR(\fBFIELD *\fR\fIfield\fR, \fBint\fR \fIfrow\fR, \f
  28 .fi

  30 .LP
  31 .nf
  32 \fBint\fR \fBfree_field\fR(\fBFIELD *\fR\fIfield\fR);
  33 .fi

  35 .SH DESCRIPTION
  36 .sp
  36 .LP
  37 \fBnew_field()\fR creates a new field with \fIr\fR rows and \fIc\fR columns,
  38 starting at \fIfrow\fR, \fIfcol\fR, in the subwindow of a form. \fInrow\fR is
  39 the number of off-screen rows and \fInbuf\fR is the number of additional
  40 working buffers. This routine returns a pointer to the new field.
  41 .sp
  42 .LP
  43 \fBdup_field()\fR duplicates \fIfield\fR at the specified location. All field
  44 attributes are duplicated, including the current contents of the field buffers.
  45 .sp
  46 .LP
  47 \fBlink_field()\fR also duplicates \fIfield\fR at the specified location.
  48 However, unlike \fBdup_field()\fR, the new field shares the field buffers with
  49 the original field. After creation, the attributes of the new field can be
  50 changed without affecting the original field.
  51 .sp
  52 .LP
  53 \fBfree_field()\fR frees the storage allocated for \fIfield\fR.
  54 .SH RETURN VALUES
  56 .sp
  55 .LP
  56 Routines that return pointers return \fINULL\fR on error. \fBfree_field()\fR
```



```
  57 returns one of the following:
  58 .sp
  59 .ne 2
  60 .na
  61 \fBE_OK\fR
  62 .ad
  63 .RS 18n
  64 The function returned successfully.
  66 Thefunction returned successfully.
  65 .RE

  67 .sp
  68 .ne 2
  69 .na
  70 \fBE_CONNECTED\fR
  71 .ad
  72 .RS 18n
  73 The field is already connected to a form.
  74 .RE

  76 .sp
  77 .ne 2
  78 .na
  79 \fBE_SYSTEM_ERROR\fR
  80 .ad
  81 .RS 18n
  82 System error.
  83 .RE

  85 .sp
  86 .ne 2
  87 .na
  88 \fBE_BAD_ARGUMENT\fR
  89 .ad
  90 .RS 18n
  91 An argument is incorrect.
  92 .RE

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

  99 .sp
 100 .TS
 101 box;
 102 c | c
 103 l | l .
 104 ATTRIBUTE TYPE  ATTRIBUTE VALUE
 105 _
 106 MT-Level        Unsafe
 107 .TE

 109 .SH SEE ALSO
 113 .sp
 110 .LP
 111 \fBcurses\fR(3CURSES), \fBforms\fR(3CURSES), \fBattributes\fR(5)
 112 .SH NOTES
 117 .sp
 113 .LP
 114 The header \fB<form.h>\fR automatically includes the headers  \fB<eti.h>\fR and
 115 \fB<curses.h>\fR\&.
```

```
**********************************************************
    1998 Fri Jan 11 21:14:03 2019
new/usr/src/man/man3elf/elf32_checksum.3elf
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
'\" te
.\"  Copyright 1989 AT&T  Copyright (c) 2001, 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 ELF32_CHECKSUM 3ELF "Jul 11, 2001"
.SH NAME
elf32_checksum, elf64_checksum \- return checksum of elf image
.SH SYNOPSIS
.LP
.nf
cc [ \fIflag\fR ... ] \fIfile\fR ... \fB-lelf\fR [ \fIlibrary\fR ... ]
#include <libelf.h>

\fBlong\fR \fBelf32_checksum\fR(\fBElf *\fR\fIelf\fR);
.fi

.LP
.nf
\fBlong\fR \fBelf64_checksum\fR(\fBElf *\fR\fIelf\fR);
.fi

.SH DESCRIPTION
.sp
.LP
The \fBelf32_checksum()\fR function returns a simple checksum of selected
sections of the image identified by \fIelf\fR. The value is typically used as
the \fB\&.dynamic\fR tag \fBDT_CHECKSUM\fR, recorded in dynamic executables and
shared objects.
.sp
.LP
Selected sections of the image are used to calculate the checksum in order that
Selected sections of the image are used to calcluate the checksum in order that
its value is not affected by utilities such as \fBstrip\fR(1).
.sp
.LP
For the 64\(mibit class, replace 32 with 64 as appropriate.
.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     Stable
_
MT-Level        MT-Safe
.TE

.SH SEE ALSO
.sp
.LP
```



```
\fBelf\fR(3ELF), \fBelf_version\fR(3ELF), \fBgelf\fR(3ELF), \fBlibelf\fR(3LIB),
\fBattributes\fR(5)
```

```
**********************************************************
    8172 Fri Jan 11 21:14:03 2019
new/usr/src/man/man3nsl/getrpcbyname.3nsl
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
'\" te
.\"  Copyright (c) 1998 Sun Microsystems, Inc.  All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH GETRPCBYNAME 3NSL "Feb 20, 1998"
.SH NAME
getrpcbyname, getrpcbyname_r, getrpcbynumber, getrpcbynumber_r, getrpcent,
getrpcent_r, setrpcent, endrpcent \- get RPC entry
.SH SYNOPSIS
.LP
.nf
\fBcc\fR [ \fIflag\fR ... ] \fIfile\fR ... \fB-lnsl\fR [ \fIlibrary\fR ... ]
#include <rpc/rpcent.h>



\fBstruct rpcent *\fR\fBgetrpcbyname\fR(\fBconst char *\fR\fIname\fR);
.fi

.LP
.nf
\fBstruct rpcent *\fR\fBgetrpcbyname_r\fR(\fBconst char *\fR\fIname\fR, \fBstruc
     \fBchar *\fR\fIbuffer\fR, \fBint\fR \fIbuflen\fR);
.fi

.LP
.nf
\fBstruct rpcent *\fR\fBgetrpcbynumber\fR(\fBconst int\fR \fInumber\fR);
.fi

.LP
.nf
\fBstruct rpcent *\fR\fBgetrpcbynumber_r\fR(\fBconst int\fR \fInumber\fR, \fBstr
     \fBchar *\fR\fIbuffer\fR, \fBint\fR \fIbuflen\fR);
.fi

.LP
.nf
\fBstruct rpcent *\fR\fBgetrpcent\fR(\fBvoid\fR);
.fi

.LP
.nf
\fBstruct rpcent *\fR\fBgetrpcent_r\fR(\fBstruct rpcent *\fR\fIresult\fR, \fBcha
     \fBint\fR \fIbuflen\fR);
.fi

.LP
.nf
\fBvoid\fR \fBsetrpcent\fR(\fBconst int\fR \fIstayopen\fR);
.fi

.LP
.nf
\fBvoid\fR \fBendrpcent\fR(\fBvoid\fR);
.fi
```



```
.SH DESCRIPTION
.sp
.LP
These functions are used to obtain entries for RPC (Remote Procedure Call)
services.  An entry may come from any of the sources for \fBrpc\fR specified in
the \fB/etc/nsswitch.conf\fR file (see \fBnsswitch.conf\fR(4)).
.sp
.LP
\fBgetrpcbyname()\fR searches for an entry with the RPC service name specified
by the parameter \fIname\fR.
.sp
.LP
\fBgetrpcbynumber()\fR searches for an entry with the RPC program number
\fInumber\fR.
.sp
.LP
The functions \fBsetrpcent()\fR, \fBgetrpcent()\fR, and \fBendrpcent()\fR are
used to enumerate RPC entries from the database.
.sp
.LP
\fBsetrpcent()\fR sets (or resets) the enumeration to the beginning of the set
of RPC entries.  This function should be called before the first call to
\fBgetrpcent()\fR. Calls to \fBgetrpcbyname()\fR and \fBgetrpcbynumber()\fR
leave the enumeration position in an indeterminate state.   If the
\fIstayopen\fR flag is non-zero, the system may keep allocated resources such
as open file descriptors until a subsequent call to \fBendrpcent()\fR.
.sp
.LP
Successive calls to \fBgetrpcent()\fR return either successive entries or
\fBNULL,\fR indicating the end of the enumeration.
.sp
.LP
\fBendrpcent()\fR may be called to indicate that the caller expects to do no
further RPC entry retrieval operations; the system may then  deallocate
resources it was using.  It is still allowed, but possibly less efficient, for
the process to call more RPC entry retrieval functions after calling
\fBendrpcent()\fR.
.SS "Reentrant Interfaces"
.sp
.LP
The functions \fBgetrpcbyname()\fR, \fBgetrpcbynumber()\fR, and
\fBgetrpcent()\fR use static storage that is re-used in each call, making these
routines unsafe for use in multithreaded applications.
.sp
.LP
The functions \fBgetrpcbyname_r()\fR, \fBgetrpcbynumber_r()\fR, and
\fBgetrpcent_r()\fR provide reentrant interfaces for these operations.
.sp
.LP
Each reentrant interface performs the same operation as its non-reentrant
counterpart, named by removing the  ''\fB_r\fR'' suffix.  The reentrant
interfaces, however, use buffers supplied by the caller to store returned
results, and  are safe for use in both single-threaded and multithreaded
applications.
.sp
.LP
Each reentrant interface takes the same parameters as its non-reentrant
counterpart, as well as the following additional parameters.  The parameter
\fIresult\fR must be a pointer to a \fBstruct rpcent\fR structure allocated by
the caller.  On successful completion, the function returns the RPC entry in
this structure. The parameter \fIbuffer\fR must be a pointer to a buffer
supplied by the caller.  This buffer is used as storage space for the RPC entry
data.  All of the pointers within the returned \fBstruct rpcent\fR \fIresult\fR
point to data stored within this buffer (see \fBRETURN VALUES\fR). The buffer
must be large enough to hold all of the data associated with the RPC entry. The
parameter \fIbuflen\fR should give the size in bytes of the buffer indicated by
```

```
\fIbuffer\fR.
.sp
.LP
For enumeration in multithreaded applications, the position within the
enumeration is a process-wide property shared by all threads. \fBsetrpcent()\fR
may be used in a multithreaded application but resets the enumeration position
for all threads.  If multiple threads interleave calls to \fBgetrpcent_r()\fR,
the threads will enumerate disjoint subsets of the RPC entry database.
.sp
.LP
Like their non-reentrant counterparts, \fBgetrpcbyname_r()\fR and
\fBgetrpcbynumber_r()\fR leave the enumeration position in an indeterminate
state.
.SH RETURN VALUES
.sp
.LP
RPC entries are represented by the \fBstruct rpcent\fR structure defined in
\fB<rpc/rpcent.h>\fR:
.sp
.in +2
.nf
struct rpcent {
   char *r_name;       /* name of this rpc service
   char **r_aliases;   /* zero-terminated list of alternate names */
   int r_number;       /* rpc program number */
};
.fi
.in -2

.sp
.LP
The functions \fBgetrpcbyname()\fR, \fBgetrpcbyname_r(\|)\fR,
\fBgetrpcbynumber(\|)\fR, and \fBgetrpcbynumber_r()\fR each return a pointer to
a \fBstruct rpcent\fR if they successfully locate the requested entry;
otherwise they return \fBNULL.\fR
.sp
.LP
The functions \fBgetrpcent()\fR and \fBgetrpcent_r()\fR each return a pointer
to a \fBstruct rpcent\fR if they successfully enumerate an entry; otherwise
they return \fBNULL,\fR indicating the end of the enumeration.
.sp
.LP
The functions \fBgetrpcbyname()\fR, \fBgetrpcbynumber()\fR, and
\fBgetrpcent()\fR use static storage, so returned data must be copied before a
subsequent call to any of these functions if the data is to be saved.
.sp
.LP
When the pointer returned by the reentrant functions \fBgetrpcbyname_r()\fR,
\fBgetrpcbynumber_r()\fR, and \fBgetrpcent_r()\fR is non-NULL, it is always
equal to the \fIresult\fR pointer that was supplied by the caller.
.SH ERRORS
.sp
.LP
\fBThe reentrant functions  \fBgetrpcbyname_r()\fR, \fBgetrpcbynumber_r(\|)\fR and\fR
\fIThe reentrant functions  \fBgetrpcyname_r()\fR, \fBgetrpcbynumber_r(\|)\fR and\fR
\fBgetrpcent_r()\fR will return \fBNULL\fR and set \fBerrno\fR to \fBERANGE\fR
if the length of the buffer supplied by caller is not large enough to store the
result. See \fBIntro\fR(2) for the proper usage and interpretation of
\fBerrno\fR in multithreaded applications.
.SH FILES
.sp
.LP
\fB/etc/rpc\fR
.sp
.LP
\fB/etc/nsswitch.conf\fR
```



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

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
MT-Level        T{
See "Reentrant Interfaces" in \fBDESCRIPTION\fR.
T}
.TE

.SH SEE ALSO
.sp
.LP
\fBrpcinfo\fR(1M), \fBrpc\fR(3NSL), \fBnsswitch.conf\fR(4), \fBrpc\fR(4),
\fBattributes\fR(5)
.SH WARNINGS
.sp
.LP
The reentrant interfaces \fBgetrpcbyname_r()\fR, \fBgetrpcbynumber_r()\fR, and
\fBgetrpcent_r()\fR are included in this release on an uncommitted basis only,
and are subject to change or removal in future minor releases.
.SH NOTES
.sp
.LP
When compiling multithreaded applications, see  \fBIntro\fR(3), \fINotes On
Multithreaded Applications\fR, for information about the use of the
\fB_REENTRANT\fR flag.
.sp
.LP
Use of the enumeration interfaces \fBgetrpcent()\fR and \fBgetrpcent_r()\fR is
discouraged; enumeration may not be supported for all database sources.  The
semantics of enumeration are discussed further in \fBnsswitch.conf\fR(4).
```

```
**********************************************************
    8694 Fri Jan 11 21:14:03 2019
new/usr/src/man/man3nsl/t_sndudata.3nsl
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
```

```
.\"
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for
.\" permission to reproduce portions of its copyrighted documentation.
.\" Original documentation from The Open Group can be obtained online at
.\" http://www.opengroup.org/bookstore/.
.\"
.\" The Institute of Electrical and Electronics Engineers and The Open
.\" Group, have given us permission to reprint portions of their
.\" documentation.
.\"
.\" In the following statement, the phrase ``this text'' refers to portions
.\" of the system documentation.
.\"
.\" Portions of this text are reprinted and reproduced in electronic form
.\" in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition,
.\" Standard for Information Technology -- Portable Operating System
.\" Interface (POSIX), The Open Group Base Specifications Issue 6,
.\" Copyright (C) 2001-2004 by the Institute of Electrical and Electronics
.\" Engineers, Inc and The Open Group.  In the event of any discrepancy
.\" between these versions and the original IEEE and The Open Group
.\" Standard, the original IEEE and The Open Group Standard is the referee
.\" document.  The original Standard can be obtained online at
.\" http://www.opengroup.org/unix/online.html.
.\"
.\" This notice shall appear on any product containing this material.
.\"
.\" The contents of this file are subject to the terms of the
.\" Common Development and Distribution License (the "License").
.\" You may not use this file except in compliance with the License.
.\"
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
.\" or http://www.opensolaris.org/os/licensing.
.\" See the License for the specific language governing permissions
.\" and limitations under the License.
.\"
.\" When distributing Covered Code, include this CDDL HEADER in each
.\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
.\" If applicable, add the following below this CDDL HEADER, with the
.\" fields enclosed by brackets "[]" replaced with your own identifying
.\" information: Portions Copyright [yyyy] [name of copyright owner]
.\"
.\"
.\" Portions Copyright 1989 AT&T
.\" Copyright 1994, The X/Open Company Ltd.  All Rights Reserved.
.\" Portions Copyright (c) 1998, Sun Microsystems, Inc.  All Rights Reserved.
.\"
.TH T_SNDUDATA 3NSL "May 7, 1998"
.SH NAME
t_sndudata \- send a data unit
.SH SYNOPSIS
.LP
.nf
#include <xti.h>



\fBint\fR \fBt_sndudata\fR(\fBint\fR \fIfd\fR, \fBconst struct t_unitdata *\fR\f
```



```
.fi

.SH DESCRIPTION
.sp
.LP
This routine is part of the \fBXTI\fR interfaces which evolved from the
\fBTLI\fR interfaces. \fBXTI\fR represents the future evolution of these
interfaces. However, \fBTLI\fR interfaces are supported for compatibility. When
using a \fBTLI\fR routine that has the same name as an \fBXTI\fR routine, the
\fBtiuser.h\fR header file must be used.  Refer to the  \fBTLI\fR
\fBCOMPATIBILITY\fR section for a description of differences between the two
interfaces.
.sp
.LP
This function is used in connectionless-mode to send a data unit to another
transport user. The argument \fIfd\fR identifies the local transport endpoint
through which data will be sent, and \fIunitdata\fR points to a
\fBt_unitdata\fR structure containing the following members:
.sp
.in +2
.nf
struct netbuf addr;
struct netbuf opt;
struct netbuf udata;
.fi
.in -2

.sp
.LP
In \fIunitdata\fR, \fIaddr\fR specifies the protocol address of the destination
user, \fIopt\fR identifies options that the user wants associated with this
request, and \fIudata\fR specifies the user data to be sent. The user may
choose not to specify what protocol options are associated with the transfer by
setting the \fIlen\fR field of \fIopt\fR to zero. In this case, the provider
uses the option values currently set for the communications endpoint.
.sp
.LP
If the \fIlen\fR field of \fIudata\fR is zero, and sending of zero octets is
not supported by the underlying transport service, the \fBt_sndudata()\fR will
return  -1 with \fBt_errno\fR set to \fBTBADDATA\fR.
.sp
.LP
By default, \fBt_sndudata()\fR operates in synchronous mode and may wait if
flow control restrictions prevent the data from being accepted by the local
transport provider at the time the call is made.  However, if  \fBO_NONBLOCK\fR
\fBis\fR \fBset\fR \fBby\fR \fBmeans\fR \fBof\fR \fBt_open\fR(3NSL) or
\fBfcntl\fR(2), \fBt_sndudata()\fR will execute in asynchronous mode and will
fail under such conditions. The process can arrange to be notified of the
clearance of a flow control restriction by means of either \fBt_look\fR(3NSL)
or the EM interface.
.sp
.LP
If the amount of data specified in \fIudata\fR exceeds the TSDU size as
returned in the \fItsdu\fR field of the \fIinfo\fR argument of
\fBt_open\fR(3NSL) or \fBt_getinfo\fR(3NSL), a \fBTBADDATA\fR error will be
generated. If \fBt_sndudata()\fR is called before the destination user has
activated its transport endpoint (see \fBt_bind\fR(3NSL)), the data unit may be
discarded.
.sp
.LP
If it is not possible for the transport provider to immediately detect the
conditions that cause the errors \fBTBADADDR\fR and \fBTBADOPT\fR, these
conditions that cause the errors \fBTBADADDR\fR and \fBTBADOPT\fR, these
errors will alternatively be returned by \fIt_rcvuderr.\fR Therefore, an
application must be prepared to receive these errors in both of these ways.
.sp
```

```
123 .LP
124 If the call is interrupted, \fBt_sndudata()\fR will return \fBEINTR\fR and the
125 datagram will not be sent.
126 .SH RETURN VALUES
128 .sp
127 .LP
128 Upon successful completion, a value of  \fB0\fR is returned.  Otherwise, a
129 value of  -1 is returned and \fBt_errno\fR is set to indicate an error.
130 .SH VALID STATES
133 .sp
131 .LP
132 \fBT_IDLE\fR.
133 .SH ERRORS
137 .sp
134 .LP
135 On failure, \fBt_errno\fR is set to one of the following:
136 .sp
137 .ne 2
138 .na
139 \fB\fBTBADADDR\fR\fR
140 .ad
141 .RS 15n
142 The specified protocol address was in an incorrect format or contained illegal
143 information.
144 .RE

146 .sp
147 .ne 2
148 .na
149 \fB\fBTBADDATA\fR\fR
150 .ad
151 .RS 15n
152 Illegal amount of data. A single send was attempted specifying a TSDU greater
153 than that specified in the \fIinfo\fR argument, or a send of a zero byte TSDU
154 is not supported by the provider.
155 .RE

157 .sp
158 .ne 2
159 .na
160 \fB\fBTBADF\fR\fR
161 .ad
162 .RS 15n
163 The specified file descriptor does not refer to a transport endpoint.
164 .RE

166 .sp
167 .ne 2
168 .na
169 \fB\fBTBADOPT\fR\fR
170 .ad
171 .RS 15n
172 The specified options were in an incorrect format or contained illegal
173 information.
174 .RE

176 .sp
177 .ne 2
178 .na
179 \fB\fBTFLOW\fR\fR
180 .ad
181 .RS 15n
182 \fBO_NONBLOCK\fR was set, but the flow control mechanism prevented the
183 transport provider from accepting any data at this time.
184 .RE
```



```
186 .sp
187 .ne 2
188 .na
189 \fB\fBTLOOK\fR\fR
190 .ad
191 .RS 15n
192 An asynchronous event has occurred on this transport endpoint.
193 .RE

195 .sp
196 .ne 2
197 .na
198 \fB\fBTNOTSUPPORT\fR\fR
199 .ad
200 .RS 15n
201 This function is not supported by the underlying transport provider.
202 .RE

204 .sp
205 .ne 2
206 .na
207 \fB\fBTOUTSTATE\fR\fR
208 .ad
209 .RS 15n
210 The communications endpoint referenced by  \fIfd\fR is not in one of the states
211 in which a call to this function is valid.
212 .RE

214 .sp
215 .ne 2
216 .na
217 \fB\fBTPROTO\fR\fR
218 .ad
219 .RS 15n
220 This error indicates that a communication problem has been detected between XTI
221 and the transport provider for which there is no other suitable XTI error
222 \fB(t_errno)\fR.
223 .RE

225 .sp
226 .ne 2
227 .na
228 \fB\fBTSYSERR\fR\fR
229 .ad
230 .RS 15n
231 A system error has occurred during execution of this function.
232 .RE

234 .SH TLI COMPATIBILITY
239 .sp
235 .LP
236 The \fBXTI\fR and \fBTLI\fR interface definitions have common names but use
237 different header files. This, and other semantic differences between the two
238 interfaces are described in the subsections below.
239 .SS "Interface Header"
245 .sp
240 .LP
241 The \fBXTI\fR interfaces use the header file, \fBxti.h.\fR \fBTLI\fR interfaces
242 should \fInot\fR use this header.  They should use the header:
243 .br
244 .in +2
245 #include <tiuser.h>
246 .in -2
247 .SS "Error Description Values"
254 .sp
248 .LP
```

```
249 The \fBt_errno\fR values that can be set by the \fBXTI\fR interface and cannot
250 be set by the \fBTLI\fR interface are:
251 .br
252 .in +2
253 \fBTPROTO\fR
254 .in -2
255 .br
256 .in +2
257 \fBTBADADDR\fR
258 .in -2
259 .br
260 .in +2
261 \fBTBADOPT\fR
262 .in -2
263 .br
264 .in +2
265 \fBTLOOK\fR
266 .in -2
267 .br
268 .in +2
269 \fBTOUTSTATE\fR
270 .in -2
271 .SS "Notes"
279 .sp
272 .LP
273 Whenever this function fails with \fBt_error\fR set to \fBTFLOW,\fR
274 \fBO_NONBLOCK\fR must have been set.
275 .SS "Option Buffers"
284 .sp
276 .LP
277 The format of the options in an \fBopt\fR buffer is dictated by the transport
278 provider. Unlike the \fBXTI\fR interface, the \fBTLI\fR interface does not fix
279 the buffer format.
280 .SH ATTRIBUTES
290 .sp
281 .LP
282 See \fBattributes\fR(5)  for descriptions of the following attributes:
283 .sp

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

295 .SH SEE ALSO
306 .sp
296 .LP
297 \fBfcntl\fR(2), \fBt_alloc\fR(3NSL), \fBt_bind\fR(3NSL), \fBt_error\fR(3NSL),
298 \fBt_getinfo\fR(3NSL), \fBt_look\fR(3NSL), \fBt_open\fR(3NSL),
299 \fBt_rcvudata\fR(3NSL), \fBt_rcvuderr\fR(3NSL), \fBattributes\fR(5)
```

```
**********************************************************
    8045 Fri Jan 11 21:14:03 2019
new/usr/src/man/man3nsl/t_sndvudata.3nsl
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
```

```
.\"
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for
.\" permission to reproduce portions of its copyrighted documentation.
.\" Original documentation from The Open Group can be obtained online at
.\" http://www.opengroup.org/bookstore/.
.\"
.\" The Institute of Electrical and Electronics Engineers and The Open
.\" Group, have given us permission to reprint portions of their
.\" documentation.
.\"
.\" In the following statement, the phrase ‘‘this text’’ refers to portions
.\" of the system documentation.
.\"
.\" Portions of this text are reprinted and reproduced in electronic form
.\" in the SunOS Reference Manual, from IEEE Std 1003.1, 2004 Edition,
.\" Standard for Information Technology -- Portable Operating System
.\" Interface (POSIX), The Open Group Base Specifications Issue 6,
.\" Copyright (C) 2001-2004 by the Institute of Electrical and Electronics
.\" Engineers, Inc and The Open Group.  In the event of any discrepancy
.\" between these versions and the original IEEE and The Open Group
.\" Standard, the original IEEE and The Open Group Standard is the referee
.\" document.  The original Standard can be obtained online at
.\" http://www.opengroup.org/unix/online.html.
.\"
.\" This notice shall appear on any product containing this material.
.\"
.\" The contents of this file are subject to the terms of the
.\" Common Development and Distribution License (the "License").
.\" You may not use this file except in compliance with the License.
.\"
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
.\" or http://www.opensolaris.org/os/licensing.
.\" See the License for the specific language governing permissions
.\" and limitations under the License.
.\"
.\" When distributing Covered Code, include this CDDL HEADER in each
.\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
.\" If applicable, add the following below this CDDL HEADER, with the
.\" fields enclosed by brackets "[]" replaced with your own identifying
.\" information: Portions Copyright [yyyy] [name of copyright owner]
.\"
.\"
.\" Portions Copyright 1989 AT&T
.\" Copyright 1994, The X/Open Company Ltd.  All Rights Reserved.
.\" Portions Copyright (c) 1998, Sun Microsystems, Inc.  All Rights Reserved.
.\"
.TH T_SNDVUDATA 3NSL "Aug 23, 2001"
.SH NAME
t_sndvudata \- send a data unit from one or more noncontiguous buffers
.SH SYNOPSIS
.LP
.nf
#include <xti.h>



\fBint\fR \fBt_sndvudata\fR(\fBint\fR \fIfd\fR, \fBstruct t_unitdata *\fR\fIunit
     \fBunsigned int\fR \fIiovcount\fR);
.fi

.SH DESCRIPTION
.sp
.LP
This function is used in connectionless mode to send a data unit to another
transport user.  The argument  \fIfd\fR identifies the local transport endpoint
through which data will be sent,  \fIiovcount\fR contains the number of
non-contiguous  \fIudata\fR buffers and is limited to an implementation-defined
value given by  \fBT_IOV_MAX\fR which is at least 16, and  \fIunitdata\fR
points to a  \fBt_unitdata\fR structure containing the following members:
.sp
.in +2
.nf
struct netbuf addr;
struct netbuf opt;
struct netbuf udata;
.fi
.in -2

.sp
.LP
If the limit on  \fIiovcount\fR is exceeded, the function fails with
\fBTBADDATA\fR.
.sp
.LP
In  \fBunitdata\fR, \fIaddr\fR specifies the protocol address of the
destination user, and  \fIopt\fR identifies options that the user wants
associated with this request. The  \fIudata\fR field is not used.  The user may
choose not to specify what protocol options are associated with the transfer by
setting the  \fIlen\fR field of  \fIopt\fR to zero.  In this case, the provider
may use default options.
.sp
.LP
The data to be sent is identified by  \fIiov\fR\fB[0]\fR\fI\fR through \fIiov
[iovcount-1]\fR\fB\&.\fR
.sp
.LP
Note that the limit on the total number of bytes available in all buffers
passed:
.sp
.in +2
.nf
\fIiov(0).iov_len + . . + iov(iovcount-1).iov_len \fR
.fi
.in -2

.sp
.LP
may be constrained by implementation limits. If no other constraint applies, it
will be limited by \fBINT_MAX\fR. In practice, the availability of memory to an
application is likely to impose a lower limit on the amount of data that can be
sent or received using scatter/gather functions.
.sp
.LP
By default,  \fBt_sndvudata()\fR operates in synchronous mode and may wait if
flow control restrictions prevent the data from being accepted by the local
transport provider at the time the call is made.  However, if  \fBO_NONBLOCK\fR
is set by means of  \fBt_open\fR(3NSL) or  \fBfcntl\fR(2), \fBt_sndvudata()\fR
executes in asynchronous mode and will fail under such conditions.  The process
can arrange to be notified of the clearance of a flow control restriction by
means of either  \fBt_look\fR(3NSL) or the EM interface.
.sp
.LP
If the amount of data specified in  \fIiov\fR\fB0\fR\fI\fR through \fIiov
```

[iovcount-1]\fR exceeds the TSDU size as returned in the \fItsdu\fR field of the \fIinfo\fR argument of \fBt_open\fR(3NSL) or \fBt_getinfo\fR(3NSL), or is zero and sending of zero octets is not supported by the underlying transport service, a \fBTBADDATA\fR error is generated. If \fBt_sndvudata()\fR is called before the destination user has activated its transport endpoint (see \fBt_bind\fR(3NSL)\|), the data unit may be discarded.

.sp
.LP
If it is not possible for the transport provider to immediately detect the **conditions that cause the errors \fBTBADADDR\fR and \fBTBADOPT\fR, these**
conditions that cause the errors \fBTBADADDR\fR and \fBTBADOPT\fR, these
errors will alternatively be returned by \fBt_rcvuderr\fR(3NSL). An application must therefore be prepared to receive these errors in both of these ways.

.SH RETURN VALUES
.sp
.LP
Upon successful completion, a value of \fB0\fR is returned. Otherwise, a value of -1 is returned and \fBt_errno\fR is set to indicate an error.

.SH VALID STATES
.sp
.LP
\fBT_IDLE\fR.

.SH ERRORS
.sp
.LP
On failure, \fBt_errno\fR is set to one of the following:

.sp
.ne 2
.na
\fB\fBTBADADDR\fR\fR
.ad
.RS 15n
The specified protocol address was in an incorrect format or contained illegal information.
.RE

.sp
.ne 2
.na
\fB\fBTBADDATA\fR\fR
.ad
.RS 15n
Illegal amount of data.
.RS +4
.TP
.ie t \(bu
.el o
A single send was attempted specifying a TSDU greater than that specified in the \fIinfo\fR argument, or a send of a zero byte TSDU is not supported by the provider.
.RE
.RS +4
.TP
.ie t \(bu
.el o
\fIiovcount\fR is greater than \fBT_IOV_MAX.\fR
.RE
.RE

.sp
.ne 2
.na
\fB\fBTBADF\fR\fR
.ad
.RS 15n

The specified file descriptor does not refer to a transport endpoint.
.RE

.sp
.ne 2
.na
\fB\fBTBADOPT\fR\fR
.ad
.RS 15n
The specified options were in an incorrect format or contained illegal information.
.RE

.sp
.ne 2
.na
\fB\fBTFLOW\fR\fR
.ad
.RS 15n
\fBO_NONBLOCK\fR \fBi\fR was set, but the flow control mechanism prevented the transport provider from accepting any data at this time.
.RE

.sp
.ne 2
.na
\fB\fBTLOOK\fR\fR
.ad
.RS 15n
An asynchronous event has occurred on this transport endpoint.
.RE

.sp
.ne 2
.na
\fB\fBTNOTSUPPORT\fR\fR
.ad
.RS 15n
This function is not supported by the underlying transport provider.
.RE

.sp
.ne 2
.na
\fB\fBTOUTSTATE\fR\fR
.ad
.RS 15n
The communications endpoint referenced by \fIfd\fR is not in one of the states in which a call to this function is valid.
.RE

.sp
.ne 2
.na
\fB\fBTPROTO\fR\fR
.ad
.RS 15n
This error indicates that a communication problem has been detected between XTI and the transport provider for which there is no other suitable XTI error \fB(t_errno)\fR.
.RE

.sp
.ne 2
.na
\fB\fBTSYSERR\fR\fR

```
252 .ad
253 .RS 15n
254 A system error has occurred during execution of this function.
255 .RE

257 .SH TLI COMPATIBILITY
262 .sp
258 .LP
259 In the \fBTLI\fR interface definition, no counterpart of this routine was
260 defined.
261 .SH ATTRIBUTES
267 .sp
262 .LP
263 See \fBattributes\fR(5)  for descriptions of the following attributes:
264 .sp

266 .sp
267 .TS
268 box;
269 c | c
270 l | l .
271 ATTRIBUTE TYPE  ATTRIBUTE VALUE
272 _
273 MT Level        Safe
274 .TE

276 .SH SEE ALSO
283 .sp
277 .LP
278 \fBfcntl\fR(2), \fBt_alloc\fR(3NSL), \fBt_open\fR(3NSL),
279 \fBt_rcvudata\fR(3NSL), \fBt_rcvvudata\fR(3NSL) \fBt_rcvuderr\fR(3NSL),
280 \fBt_sndudata\fR(3NSL), \fBattributes\fR(5)
```

```
**********************************************************
    7271 Fri Jan 11 21:14:03 2019
new/usr/src/man/man3secdb/getauthattr.3secdb
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
'\" te
.\" Copyright (c) 2009, Sun Microsystems, Inc.  All Rights Reserved.
.\" The contents of this file are subject to the terms of the Common Development
.\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
.\" When distributing Covered Code, include this CDDL HEADER in each file and in
.TH GETAUTHATTR 3SECDB "Aug 13, 2018"
.SH NAME
getauthattr, getauthnam, free_authattr, setauthattr, endauthattr, chkauthattr
\- get authorization entry
.SH SYNOPSIS
.LP
.nf
cc [ \fIflag\fR... ] \fIfile\fR... -lsecdb  -lsocket  -lnsl  [ \fIlibrary\fR...
#include <auth_attr.h>
#include <secdb.h>

\fBauthattr_t *\fR\fBgetauthattr\fR(\fBvoid\fR);
.fi

.LP
.nf
\fBauthattr_t *\fR\fBgetauthnam\fR(\fBconst char *\fR\fIname\fR);
.fi

.LP
.nf
\fBvoid\fR \fBfree_authattr\fR(\fBauthattr_t *\fR\fIauth\fR);
.fi

.LP
.nf
\fBvoid\fR \fBsetauthattr\fR(\fBvoid\fR);
.fi

.LP
.nf
\fBvoid\fR \fBendauthattr\fR(\fBvoid\fR);
.fi

.LP
.nf
\fBint\fR \fBchkauthattr\fR(\fBconst char *\fR\fIauthname\fR, \fBconst char *\fR
.fi

.SH DESCRIPTION
.LP
The \fBgetauthattr()\fR and \fBgetauthnam()\fR functions each return an
\fBauth_attr\fR(4) entry. Entries can come from any of the sources specified in
the \fBnsswitch.conf\fR(4) file.
.sp
.LP
The \fBgetauthattr()\fR function enumerates \fBauth_attr\fR entries. The
\fBgetauthnam()\fR function searches for an \fBauth_attr\fR entry with a given
authorization name \fIname\fR. Successive calls to these functions return
either successive \fBauth_attr\fR entries or \fINULL\fR.
.sp
.LP
The internal representation of an \fBauth_attr\fR entry is an \fBauthattr_t\fR
```



```
Th internal representation of an \fBauth_attr\fR entry is an \fBauthattr_t\fR
structure defined in  <\fBauth_attr.h\fR> with the following members:
.sp
.in +2
.nf
char   *name;        /* name of the authorization */
char   *res1;        /* reserved for future use */
char   *res2;        /* reserved for future use */
char   *short_desc;  /* short description */
char   *long_desc;   /* long description */
kva_t  *attr;        /* array of key-value pair attributes */
.fi
.in -2

.sp
.LP
The \fBsetauthattr()\fR function "rewinds" to the beginning of the enumeration
of \fBauth_attr\fR entries.  Calls to \fBgetauthnam()\fR can leave the
enumeration in an indeterminate state. Therefore, \fBsetauthattr()\fR should be
called before the first call to \fBgetauthattr()\fR.
.sp
.LP
The \fBendauthattr()\fR function may be called to indicate that \fBauth_attr\fR
processing is complete; the system may then close any open \fBauth_attr\fR
file, deallocate storage, and so forth.
.sp
.LP
The \fBchkauthattr()\fR function verifies whether or not a user has a given
authorization. It first reads the \fBAUTHS_GRANTED\fR key in the
\fB/etc/security/policy.conf\fR file and returns 1 if it finds a match for the
given authorization. If \fBchkauthattr()\fR does not find a match and the
\fIusername\fR is the name of the "console user", defined as the owner of
\fB/dev/console\fR, it first reads the \fBCONSOLE_USER\fR key in
\fB/etc/security/policy.conf\fR and returns 1 if the given authorization is in
any of the profiles specified in the \fBCONSOLE_USER\fR keyword, then reads the
\fBPROFS_GRANTED\fR key in \fB/etc/security/policy.conf\fR and returns 1 if the
given authorization is in any profiles specified with the \fBPROFS_GRANTED\fR
keyword. If a match is not found from the default authorizations and default
profiles, \fBchkauthattr()\fR reads the \fBuser_attr\fR(4) database. If it does
not find a match in \fBuser_attr\fR, it reads the \fBprof_attr\fR(4) database,
using the list of profiles assigned to the user, and checks if any of the
profiles assigned to the user has the given authorization.  The
\fBchkauthattr()\fR function returns 0 if it does not find a match in any of
the three sources or if the user does not exist.
.sp
.LP
A user is considered to have been assigned an authorization if either of the
following are true:
.RS +4
.TP
.ie t \(bu
.el o
The authorization name matches exactly any authorization assigned in the
\fBuser_attr\fR or \fBprof_attr\fR databases (authorization names are
case-sensitive).
.RE
.RS +4
.TP
.ie t \(bu
.el o
The authorization name suffix is not the key word \fBgrant\fR and the
authorization name matches any authorization up to the asterisk (*) character
assigned in the \fBuser_attr\fR or \fBprof_attr\fR databases.
.RE
.sp
.LP
```

```
124 The examples in the following table illustrate the conditions under which a
125 user is assigned an authorization.
126 .sp

128 .sp
129 .TS
130 box;
131 c | c | c
132 c | c | c .
133         \fB/etc/security/policy.conf\fR or      Is user
134 \fBAuthorization name\fR        \fBuser_attr\fR or \fBprof_attr\fR entry
135 _
136 solaris.printer.postscript      solaris.printer.postscript      Yes
137 solaris.printer.postscript      solaris.printer.*       Yes
138 solaris.printer.grant   solaris.printer.*       No
139 .TE

141 .sp
142 .LP
143 The \fBfree_authattr()\fR function releases memory allocated by the
144 \fBgetauthnam()\fR and \fBgetauthattr()\fR functions.
145 .SH RETURN VALUES
146 .LP
147 The \fBgetauthattr()\fR function returns a pointer to an \fBauthattr_t\fR if
148 it successfully enumerates an entry; otherwise it returns \fINULL\fR,
149 indicating the end of the enumeration.
150 .sp
151 .LP
152 The \fBgetauthnam()\fR function returns a pointer to an \fBauthattr_t\fR if it
153 successfully locates the requested entry; otherwise it returns \fINULL\fR.
154 .sp
155 .LP
156 The \fBchkauthattr()\fR function returns 1 if the user is authorized and 0 if
157 the user does not exist or is not authorized.
158 .SH USAGE
159 .LP
160 The \fBgetauthattr()\fR and \fBgetauthnam()\fR functions both allocate memory
161 for the pointers they return. This memory should be deallocated with the
162 \fBfree_authattr()\fR call.
163 .sp
164 .LP
165 Individual attributes in the \fBattr\fR structure can be referred to by calling
166 the \fBkva_match\fR(3SECDB) function.
167 .SH WARNINGS
168 .LP
169 Because the list of legal keys is likely to expand, code must be written to
170 ignore unknown key-value pairs without error.
171 .SH FILES
172 .ne 2
173 .na
174 \fB\fB/etc/nsswitch.conf\fR\fR
175 .ad
176 .RS 29n
177 configuration file lookup information for the name service switch
178 .RE

180 .sp
181 .ne 2
182 .na
183 \fB\fB/etc/user_attr\fR\fR
184 .ad
185 .RS 29n
186 extended user attributes
187 .RE

189 .sp
```



```
190 .ne 2
191 .na
192 \fB\fB/etc/security/auth_attr\fR\fR
193 .ad
194 .RS 29n
195 authorization attributes
196 .RE

198 .sp
199 .ne 2
200 .na
201 \fB\fB/etc/security/policy.conf\fR\fR
202 .ad
203 .RS 29n
204 policy definitions
205 .RE

207 .sp
208 .ne 2
209 .na
210 \fB\fB/etc/security/prof_attr\fR\fR
211 .ad
212 .RS 29n
213 profile information
214 .RE

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

221 .sp
222 .TS
223 box;
224 c | c
225 l | l .
226 ATTRIBUTE TYPE  ATTRIBUTE VALUE
227 _
228 MT-Level        MT-Safe
229 .TE

231 .SH SEE ALSO
232 .LP
233 \fBgetexecattr\fR(3SECDB), \fBgetprofattr\fR(3SECDB), \fBgetuserattr\fR(3SECDB),
234 \fBkva_match\fR(3SECDB), \fBauth_attr\fR(4), \fBnsswitch.conf\fR(4),
235 \fBpolicy.conf\fR(4), \fBprof_attr\fR(4), \fBuser_attr\fR(4),
236 \fBattributes\fR(5), \fBrbac\fR(5)
```

```
**********************************************************
   11261 Fri Jan 11 21:14:04 2019
new/usr/src/man/man3socket/connect.3socket
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
'\" te
.\" Copyright (C) 2005, Sun Microsystems, Inc.
.\" All Rights Reserved
.\" Copyright (c) 2014, Joyent, Inc.
.\" Copyright 1989 AT&T  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 CONNECT 3SOCKET "Nov 25, 2014"
.SH NAME
connect \- initiate a connection on a socket
.SH SYNOPSIS
.LP
.nf
\fBcc\fR [ \fIflag\fR ... ] \fIfile\fR ... \fB-lsocket\fR \fB -lnsl \fR [ \fIlib
#include <sys/types.h>
#include <sys/socket.h>



\fBint\fR \fBconnect\fR(\fBint\fR \fIs\fR, \fBconst struct sockaddr *\fR\fIname\
.fi

.SH DESCRIPTION
.LP
The parameter \fIs\fR is a socket. If it is of type \fBSOCK_DGRAM\fR,
\fBconnect()\fR specifies the peer with which the socket is to be associated.
This address is the address to which datagrams are to be sent if a receiver is
not explicitly designated. This address is the only address from which
datagrams are to be received. If the socket \fIs\fR is of type
\fBSOCK_STREAM\fR, \fBconnect()\fR attempts to make a connection to another
socket. The other socket is specified by \fIname\fR. \fIname\fR is an address
in the communication space of the socket. Each communication space interprets
the \fIname\fR parameter in its own way. If \fIs\fR is not bound, then \fIs\fR
will be bound to an address selected by the underlying transport provider.
Generally, stream sockets can successfully \fBconnect()\fR only once. Datagram
sockets can use \fBconnect()\fR multiple times to change their association.
Datagram sockets can dissolve the association by connecting to a null address.
.SS Non-blocking Sockets
When a socket is created, it is by default a \fBblocking\fR socket. A socket may
be configured to be \fBnon-blocking\fR either at socket creation time or through
the use of \fBfcntl\fR(2). When a socket is set to be \fBnon-blocking\fR, a call
to connect initiates an asynchronous connection. If the connection cannot be
completed without blocking, such as when making a TCP connection to a remote
server, then  the connection attempt is made in the background and \fBconnect\fR
returns -1 and errno is set to \fBEINPROGRESS\fR.
.LP
Applications can obtain the state of this connection attempt by polling the
socket's file descriptor for \fBPOLLOUT\fR. The event ports facility is the
preferred means of polling on the file descriptor, see \fBport_create\fR(3C) and
\fBport_get\fR(3C) for more information on event ports; however, applications
may also use traditional portable routines like \fBpoll\fR(2) and
\fBselect\fR(3C).
.LP
When an asynchronous connection has completed, the application must call
\fBgetsockopt\fR(3SOCKET) using the macro \fBSOL_SOCKET\fR as the \fIlevel\fR
argument and the macro \fBSO_ERROR\fR as the value of the \fIoption\fR argument.
If the value of the \fBSO_ERROR\fR socket option is zero, then the
```



```
connect was successfully established. Otherwise, the connection could not be
established and the value is the corresponding error code that would be commonly
found in \fBerrno\fR.
.LP
Even when a socket is in \fBnon-blocking\fR mode, a call to \fBconnect\fR may
fail synchronously. If any error other \fBEINPROGRESS\fR or \fBEINTR\fR occurs,
then there is no need for the application to poll for asynchronous completion.
Similarly, if a call to \fBconnect\fR returns successfully, then the socket
connection will be established and there is no need to poll for completion.
.SH EXAMPLES
.LP
\fBExample 1\fR  Performing an asynchronous connection
.sp
.LP
The following sample C program shows how to create and connect to a remote host
using TCP. The program should be compiled and linked against libnsl and
libsocket. For example, if the contents of this example where in a file called
example.c, one would run cc example.c -lnsl -lsocket.
.sp
.in +2
.nf
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <inttypes.h>
#include <stdio.h>
#include <strings.h>
#include <stdlib.h>
#include <errno.h>
#include <port.h>
#include <unistd.h>
#include <assert.h>

int
main(int argc, char *argv[])
{
        char *eptr;
        long port;
        int sock, ret, eport;
        struct sockaddr_in6 sin6;

        if (argc != 3) {
                fprintf(stderr, "connect: <IP> <port>\\n");
                return (1);
        }

        bzero(&sin6, sizeof (struct sockaddr_in6));
        sin6.sin6_family = AF_INET6;

        /*
         * Try to parse as an IPv6 address and then try v4.
         */
        ret = inet_pton(AF_INET6, argv[1], &sin6.sin6_addr);
        if (ret == -1) {
                perror("inet_pton");
                return (1);
        } else if (ret == 0) {
                struct in_addr v4;
                ret = inet_pton(AF_INET, argv[1], &v4);
                if (ret == -1) {
                        perror("inet_pton");
                        return (1);
                } else if (ret == 0) {
                        fprintf(stderr, "connect: %s is not a valid "
                            "IPv4 or IPv6 address\\n", argv[1]);
```

```
125                         return (1);
126                 }
127                 /* N.B. Not a portable macro */
128                 IN6_INADDR_TO_V4MAPPED(&v4, &sin6.sin6_addr);
129         }

131         errno = 0;
132         port = strtol(argv[2], &eptr, 10);
133         if (errno != 0 || *eptr != '\e0') {
134                 fprintf(stderr, "failed to parse port %s\\n", argv[2]);
135                 return (1);
136         }
137         if (port <= 0 || port > UINT16_MAX) {
138                 fprintf(stderr, "invalid port: %ld\\n", port);
139                 return (1);
140         }
141         sin6.sin6_port = htons(port);

143         sock = socket(AF_INET6, SOCK_STREAM | SOCK_NONBLOCK, 0);
144         if (sock < 0) {
145                 perror("socket");
146                 return (1);
147         }

149         eport = port_create();
150         if (eport < 0) {
151                 perror("port_create");
152                 (void) close(sock);
153                 return (1);
154         }

156         ret = connect(sock, (struct sockaddr *)&sin6,
157             sizeof (struct sockaddr_in6));
158         if (ret != 0 && errno != EINPROGRESS && errno != EINTR) {
159                 perror("connect");
160                 (void) close(sock);
161                 (void) close(eport);
162                 return (1);
163         }

165         if (ret != 0) {
166                 port_event_t pe;
167                 int err;
168                 socklen_t sz = sizeof (err);
169                 if (port_associate(eport, PORT_SOURCE_FD, sock, POLLOUT,
170                     NULL) != 0) {
171                         perror("port_associate");
172                         (void) close(sock);
173                         (void) close(eport);
174                         return (1);
175                 }
176                 if (port_get(eport, &pe, NULL) != 0) {
177                         perror("port_get");
178                         (void) close(sock);
179                         (void) close(eport);
180                         return (1);
181                 }
182                 assert(pe.portev_source == PORT_SOURCE_FD);
183                 assert(pe.portev_object == (uintptr_t)sock);
184                 if (getsockopt(sock, SOL_SOCKET, SO_ERROR, &err, &sz) != 0) {
185                         perror("getsockopt");
186                         (void) close(sock);
187                         (void) close(eport);
188                         return (1);
189                 }
190                 if (err != 0) {
```



```
191                         /* Asynch connect failed */
192                         fprintf(stderr, "asynchronous connect: %s\\n",
192                         fprintf(stderr, "asnchronous connect: %s\\n",
193                             strerror(err));
194                         (void) close(sock);
195                         (void) close(eport);
196                         return (1);
197                 }
198         }

200         /* Read and write to the socket and then clean up */

202         return (0);
203 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
   26072 Fri Jan 11 21:14:04 2019
new/usr/src/man/man4/ike.config.4
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
```

```
   1 '\" te
   2 .\" Copyright (c) 2009, Sun Microsystems, Inc. All Rights Reserved.
   3 .\" Copyright (c) 2015, Circonus, Inc. All Rights Reserved.
   4 .\" The contents of this file are subject to the terms of the Common Development
   5 .\"  See the License for the specific language governing permissions and limitat
   6 .\" fields enclosed by brackets "[]" replaced with your own identifying informat
   7 .TH IKE.CONFIG 4 "Apr 27, 2009"
   8 .SH NAME
   9 ike.config \- configuration file for IKE policy
  10 .SH SYNOPSIS
  11 .LP
  12 .nf
  13 \fB/etc/inet/ike/config\fR
  14 .fi

  16 .SH DESCRIPTION
  17 .LP
  18 The \fB/etc/inet/ike/config\fR file contains rules for matching inbound IKE
  19 requests. It also contains rules for preparing outbound \fBIKE\fR requests.
  20 .sp
  21 .LP
  22 You can test the syntactic correctness of an \fB/etc/inet/ike/config\fR file by
  23 using the \fB-c\fR or \fB-f\fR options of \fBin.iked\fR(1M). You must use the
  24 \fB-c\fR option to test a \fBconfig\fR file. You might need to use the \fB-f\fR
  25 option if it is not in \fB/etc/inet/ike/config\fR.
  26 .SS "Lexical Components"
  27 .LP
  28 On any line, an unquoted \fB#\fR character introduces a comment. The remainder
  29 of that line is ignored. Additionally, on any line, an unquoted \fB//\fR
  30 sequence introduces a comment. The remainder of that line is ignored.
  31 .sp
  32 .LP
  33 There are several types of lexical tokens in the \fBike.config\fR file:
  34 .sp
  35 .ne 2
  36 .na
  37 \fB\fInum\fR\fR
  38 .ad
  39 .sp .6
  40 .RS 4n
  41 A decimal, hex, or octal number representation is as in 'C'.
  42 .RE

  44 .sp
  45 .ne 2
  46 .na
  47 \fB\fIIPaddr\fR/\fIprefix\fR/\fIrange\fR\fR
  48 .ad
  49 .sp .6
  50 .RS 4n
  51 An IPv4 or IPv6 address with an optional /\fINNN\fR suffix, (where \fINNN\fR is
  52 a \fInum\fR) that indicates an address (\fBCIDR\fR) prefix (for example,
  53 \fB10.1.2.0/24\fR). An optional /\fIADDR\fR suffix (where \fIADDR\fR is a
  54 second IP address) indicates an address/mask pair (for example,
  55 \fB10.1.2.0/255.255.255.0\fR). An optional -\fIADDR\fR suffix (where \fIADDR\fR
  56 is a second IPv4 address) indicates an inclusive range of addresses (for
  57 example, \fB10.1.2.0-10.1.2.255\fR). The \fB/\fR or \fB-\fR can be surrounded
  58 by an arbitrary amount of white space.
```



```
  59 .RE

  61 .sp
  62 .ne 2
  63 .na
  64 \fB\fBXXX\fR | \fBYYY\fR | \fBZZZ\fR\fR
  65 .ad
  66 .sp .6
  67 .RS 4n
  68 Either the words \fBXX\fRX, \fBYYY\fR, or \fBZZZ\fR, for example, {yes,no}.
  69 .RE

  71 .sp
  72 .ne 2
  73 .na
  74 \fBp1-id-type\fR
  75 .ad
  76 .sp .6
  77 .RS 4n
  78 An IKE phase 1 identity type. IKE phase 1 identity types include:
  79 .br
  80 .in +2
  81 \fBdn, DN\fR
  82 .in -2
  83 .br
  84 .in +2
  85 \fBdns, DNS\fR
  86 .in -2
  87 .br
  88 .in +2
  89 \fBfqdn, FQDN\fR
  90 .in -2
  91 .br
  92 .in +2
  93 \fBgn, GN\fR
  94 .in -2
  95 .br
  96 .in +2
  97 \fBip, IP\fR
  98 .in -2
  99 .br
 100 .in +2
 101 \fBipv4\fR
 102 .in -2
 103 .br
 104 .in +2
 105 \fBipv4_prefix\fR
 106 .in -2
 107 .br
 108 .in +2
 109 \fBipv4_range\fR
 110 .in -2
 111 .br
 112 .in +2
 113 \fBipv6\fR
 114 .in -2
 115 .br
 116 .in +2
 117 \fBipv6_prefix\fR
 118 .in -2
 119 .br
 120 .in +2
 121 \fBipv6_range\fR
 122 .in -2
 123 .br
 124 .in +2
```

```
125 \fBmbox, MBOX\fR
126 .in -2
127 .br
128 .in +2
129 \fBuser_fqdn\fR
130 .in -2
131 .RE

133 .sp
134 .ne 2
135 .na
136 \fB\fB"\fR\fIstring\fR\fB"\fR\fR
137 .ad
138 .sp .6
139 .RS 4n
140 A quoted string.
141 .sp
142 Examples include:\fB"Label foo"\fR, or \fB"C=US, OU=Sun Microsystems\e, Inc.,
143 N=olemcd@eng.example.com"\fR
144 .sp
145 A backslash (\fB\e\fR) is an escape character. If the string needs an actual
146 backslash, two must be specified.
147 .RE

149 .sp
150 .ne 2
151 .na
152 \fB\fIcert-sel\fR\fR
153 .ad
154 .sp .6
155 .RS 4n
156 A certificate selector, a \fIstring\fR which specifies the identities of zero
157 or more certificates. The specifiers can conform to \fBX.509\fR naming
158 conventions.
159 .sp
160 A \fIcert-sel\fR can also use various shortcuts to match either subject
161 alternative names, the filename or \fBslot\fR of a certificate in
162 \fB/etc/inet/ike/publickeys\fR, or even the \fBISSUER\fR. For example:
163 .sp
164 .in +2
165 .nf
166 "SLOT=0"
167 "EMAIL=postmaster@domain.org"
168 "webmaster@domain.org" # Some just work w/o TYPE=
169 "IP=10.0.0.1"
170 "10.21.11.11"          # Some just work w/o TYPE=
171 "DNS=www.domain.org"
172 "mailhost.domain.org"  # Some just work w/o TYPE=
173 "ISSUER=C=US, O=Sun Microsystems\\, Inc., CN=Sun CA"
174 .fi
175 .in -2
176 .sp

178 Any \fIcert-sel\fR preceded by the character \fB!\fR indicates a negative
179 match, that is, not matching this specifier. These are the same kind of strings
180 used in \fBikecert\fR(1M).
181 .RE

183 .sp
184 .ne 2
185 .na
186 \fB\fIldap-list\fR\fR
187 .ad
188 .sp .6
189 .RS 4n
190 A quoted, comma-separated list of LDAP servers and ports.
```



```
191 .sp
192 For example, \fB"ldap1.example.com"\fR, \fB"ldap1.example.com:389"\fR,
193 \fB"ldap1.example.com:389,ldap2.example.com"\fR.
194 .sp
195 The default port for LDAP is \fB389\fR.
196 .RE

198 .sp
199 .ne 2
200 .na
201 \fB\fIparameter-list\fR\fR
202 .ad
203 .sp .6
204 .RS 4n
205 A list of parameters.
206 .RE

208 .SS "File Body Entries"
209 .LP
210 There are four main types of entries:
211 .RS +4
212 .TP
213 .ie t \(bu
214 .el o
215 global parameters
216 .RE
217 .RS +4
218 .TP
219 .ie t \(bu
220 .el o
221 IKE phase 1 transform defaults
222 .RE
223 .RS +4
224 .TP
225 .ie t \(bu
226 .el o
227 IKE rule defaults
228 .RE
229 .RS +4
230 .TP
231 .ie t \(bu
232 .el o
233 IKE rules
234 .RE
235 .sp
236 .LP
237 The global parameter entries are as follows:
238 .sp
239 .ne 2
240 .na
241 \fBcert_root \fIcert-sel\fR\fR
242 .ad
243 .sp .6
244 .RS 4n
245 The X.509 distinguished name of a certificate that is a trusted root CA
246 certificate. It must be encoded in a file in the \fB/etc/inet/ike/publickeys\fR
246 certificate.It must be encoded in a file in the \fB/etc/inet/ike/publickeys\fR
247 directory. It must have a CRL in \fB/etc/inet/ike/crl\fRs. Multiple
248 \fBcert_root\fR parameters aggregate.
249 .RE

251 .sp
252 .ne 2
253 .na
254 \fBcert_trust \fIcert-sel\fR\fR
255 .ad
```

```
256 .sp .6
257 .RS 4n
258 Specifies an X.509 distinguished name of a certificate that is self-signed, or
259 has otherwise been verified as trustworthy for signing IKE exchanges. It must
260 be encoded in a file in \fB/etc/inet/ike/publickeys\fR. Multiple
261 \fBcert_trust\fR parameters aggregate.
262 .RE

264 .sp
265 .ne 2
266 .na
267 \fBexpire_timer \fIinteger\fR\fR
268 .ad
269 .sp .6
270 .RS 4n
271 The number of seconds to let a not-yet-complete IKE Phase I (Main Mode)
272 negotiation linger before deleting it. Default value: 300 seconds.
273 .RE

275 .sp
276 .ne 2
277 .na
278 \fBignore_crls\fR
279 .ad
280 .sp .6
281 .RS 4n
282 If this keyword is present in the file, \fBin.iked\fR(1M) ignores Certificate
283 Revocation Lists (\fBCRL\fRs) for root \fBCA\fRs (as given in \fBcert_root\fR)
284 .RE

286 .sp
287 .ne 2
288 .na
289 \fBldap_server \fIldap-list\fR\fR
290 .ad
291 .sp .6
292 .RS 4n
293 A list of LDAP servers to query for certificates. The list can be additive.
294 .RE

296 .sp
297 .ne 2
298 .na
299 \fBpkcs11_path \fIstring\fR\fR
300 .ad
301 .sp .6
302 .RS 4n
303 The string that follows is a name of a shared object (\fB\&.so\fR) that
304 implements the PKCS#11 standard. The name is passed directly into
305 \fBdlopen\fR(3C) for linking, with all of the semantics of that library call.
306 By default, \fBin.iked\fR(1M) runs the same ISA as the running kernel, so a
307 library specified using \fBpkcs11_path\fR and an absolute pathname \fBmust\fR
308 match the same ISA as the kernel. One can use the start/exec SMF property (see
309 \fBsvccfg\fR(1M)) to change \fBin.iked\fR's ISA, but it is not recommended.
310 .sp
311 If this setting is not present, the default value is set to \fBlibpkcs11.so\fR.
312 Most cryptographic providers go through the default library, and this parameter
313 should only be used if a specialized provider of IKE-useful cryptographic
314 services cannot interface with the Solaris Cryptographic Framework. See
315 \fBcryptoadm\fR(1M).
316 .sp
317 This option is now deprecated, and may be removed in a future release.
318 .RE

320 .sp
321 .ne 2
```



```
322 .na
323 \fBretry_limit \fIinteger\fR\fR
324 .ad
325 .sp .6
326 .RS 4n
327 The number of retransmits before any IKE negotiation is aborted. Default value:
328 5 times.
329 .RE

331 .sp
332 .ne 2
333 .na
334 \fBretry_timer_init \fIinteger\fR or \fIfloat\fR\fR
335 .ad
336 .sp .6
337 .RS 4n
338 The initial interval (in seconds) between retransmits. This interval is doubled
339 until the \fBretry_timer_max\fR value (see below) is reached. Default value:
340 0.5 seconds.
341 .RE

343 .sp
344 .ne 2
345 .na
346 \fBretry_timer_max \fIinteger\fR or \fIfloat\fR\fR
347 .ad
348 .sp .6
349 .RS 4n
350 The maximum interval (in seconds) between retransmits. The doubling retransmit
351 interval stops growing at this limit. Default value: 30 seconds.
352 .LP
353 Note -
354 .sp
355 .RS 2
356 This value is never reached with the default configuration. The longest
357 interval is 8 (0.5 * 2 ^ (5 - 1)) seconds.
358 .RE
359 .RE

361 .sp
362 .ne 2
363 .na
364 \fBproxy \fIstring\fR\fR
365 .ad
366 .sp .6
367 .RS 4n
368 The string following this keyword must be a URL for an HTTP proxy, for example,
369 \fBhttp://proxy:8080\fR.
370 .RE

372 .sp
373 .ne 2
374 .na
375 \fBsocks \fIstring\fR\fR
376 .ad
377 .sp .6
378 .RS 4n
379 The string following this keyword must be a URL for a SOCKS proxy, for example,
380 \fBsocks://socks-proxy\fR.
381 .RE

383 .sp
384 .ne 2
385 .na
386 \fBuse_http\fR
387 .ad
```

```
.sp .6
.RS 4n
If this keyword is present in the file, \fBin.iked\fR(1M) uses HTTP to retrieve
Certificate Revocation Lists (\fBCRL\fRs).
.RE

.sp
.LP
The following IKE phase 1 transform parameters can be prefigured using
file-level defaults. Values specified within any given transform override these
defaults.
.sp
.LP
The IKE phase 1 transform defaults are as follows:
.sp
.ne 2
.na
\fBp1_lifetime_secs \fInum\fR\fR
.ad
.sp .6
.RS 4n
The proposed default lifetime, in seconds, of an IKE phase 1 security
association (\fBSA\fR).
.RE

.sp
.ne 2
.na
\fBp1_nonce_len \fInum\fR\fR
.ad
.sp .6
.RS 4n
The length in bytes of the phase 1 (quick mode) nonce data. This cannot be
specified on a per-rule basis.
.RE

.sp
.LP
The following IKE rule parameters can be prefigured using file-level defaults.
Values specified within any given rule override these defaults, unless a rule
cannot.
.sp
.ne 2
.na
\fBp2_lifetime_secs \fInum\fR\fR
.ad
.sp .6
.RS 4n
The proposed default lifetime, in seconds, of an IKE phase 2 security
association (SA). This value is optional. If omitted, a default value is used.
.RE

.sp
.ne 2
.na
\fBp2_softlife_secs \fInum\fR\fR
.ad
.sp .6
.RS 4n
The soft lifetime of a phase 2 SA, in seconds. If this value is specified, the
SA soft expires after the number of seconds specified by
\fBp2_softlife_secs\fR. This causes \fBin.iked\fR to renegotiate a new phase 2
SA before the original SA expires.
.sp
This value is optional, if omitted soft expiry occurs after 90% of the lifetime
specified by \fBp2_lifetime_secs\fR. The value specified by
```



```
\fBp2_softlife_secs\fR is ignored if \fBp2_lifetime_secs\fR is not specified.
.sp
Setting \fBp2_softlife_secs\fR to the same value as \fBp2_lifetime_secs\fR
disables soft expires.
.RE

.sp
.ne 2
.na
\fBp2_idletime_secs \fInum\fR\fR
.ad
.sp .6
.RS 4n
The idle lifetime of a phase 2 SA, in seconds. If the value is specified, the
value specifies the lifetime of the SA, if the security association is not used
before the SA is revalidated.
.RE

.sp
.ne 2
.na
\fBp2_lifetime_kb \fInum\fR\fR
.ad
.sp .6
.RS 4n
The lifetime of an SA can optionally be specified in kilobytes. This parameter
specifies the default value. If lifetimes are specified in both seconds and
kilobytes, the SA expires when either the seconds or kilobyte thresholds are
passed.
.RE

.sp
.ne 2
.na
\fBp2_softlife_kb \fInum\fR\fR
.ad
.sp .6
.RS 4n
This value is the number of kilobytes that can be protected by an SA before a
soft expire occurs (see \fBp2_softlife_secs\fR, above).
.sp
This value is optional. If omitted, soft expiry occurs after 90% of the
lifetime specified by \fBp2_lifetime_kb\fR. The value specified by
\fBp2_softlife_kb\fR is ignored if \fBp2_lifetime_kb\fR is not specified.
.RE

.sp
.ne 2
.na
\fBp2_nonce_len \fInum\fR\fR
.ad
.sp .6
.RS 4n
The length in bytes of the phase 2 (quick mode) nonce data. This cannot be
specified on a per-rule basis.
.RE

.sp
.ne 2
.na
\fBlocal_id_type \fIp1-id-type\fR\fR
.ad
.sp .6
.RS 4n
The local identity for IKE requires a type. This identity type is reflected in
the IKE exchange. The type can be one of the following:
```

```
.RS +4
.TP
.ie t \(bu
.el o
an IP address (for example, \fB10.1.1.2\fR)
.RE
.RS +4
.TP
.ie t \(bu
.el o
DNS name (for example, \fBtest.domain.com\fR)
.RE
.RS +4
.TP
.ie t \(bu
.el o
MBOX RFC 822 name (for example, \fBroot@domain.com\fR)
.RE
.RS +4
.TP
.ie t \(bu
.el o
DNX.509 distinguished name (for example, \fBC=US, O=Sun Microsystems\, Inc.,
CN=Sun Test cert\fR)
.RE
.RE

.sp
.ne 2
.na
\fBp1_xform ’{’ parameter-list ’}\fR
.ad
.sp .6
.RS 4n
A phase 1 transform specifies a method for protecting an IKE phase 1 exchange.
An initiator offers up lists of phase 1 transforms, and a receiver is expected
to only accept such an entry if it matches one in a phase 1 rule. There can be
several of these, and they are additive. There must be either at least one
phase 1 transform in a rule or a global default phase 1 transform list. In a
configuration file without a global default phase 1 transform list \fBand\fR a
rule without a phase, transform list is an invalid file. Unless specified as
optional, elements in the parameter-list must occur exactly once within a given
transform’s parameter-list:
.sp
.ne 2
.na
\fBoakley_group \fInumber\fR\fR
.ad
.sp .6
.RS 4n
The Oakley Diffie-Hellman group used for IKE SA key derivation. The group
numbers are defined in RFC 2409, Appendix A, RFC 3526, and RFC 5114, section
3.2. Acceptable values are currently:
.br
.in +2
1 (MODP 768-bit)
.in -2
.br
.in +2
2 (MODP 1024-bit)
.in -2
.br
.in +2
3 (EC2N 155-bit)
.in -2
.br
```



```
.in +2
4 (EC2N 185-bit)
.in -2
.br
.in +2
5 (MODP 1536-bit)
.in -2
.br
.in +2
14 (MODP 2048-bit)
.in -2
.br
.in +2
15 (MODP 3072-bit)
.in -2
.br
.in +2
16 (MODP 4096-bit)
.in -2
.br
.in +2
17 (MODP 6144-bit)
.in -2
.br
.in +2
18 (MODP 8192-bit)
.in -2
.br
.in +2
19 (ECP 256-bit)
.in -2
.br
.in +2
20 (ECP 384-bit)
.in -2
.br
.in +2
21 (ECP 521-bit)
.in -2
.br
.in +2
22 (MODP 1024-bit, with 160-bit Prime Order Subgroup)
.in -2
.br
.in +2
23 (MODP 2048-bit, with 224-bit Prime Order Subgroup)
.in -2
.br
.in +2
24 (MODP 2048-bit, with 256-bit Prime Order Subgroup)
.in -2
.br
.in +2
25 (ECP 192-bit)
.in -2
.br
.in +2
26 (ECP 224-bit)
.in -2
.RE

.sp
.ne 2
.na
\fBencr_alg {3des, 3des-cbc, blowfish, blowfish-cdc, des, des-cbc, aes,
aes-cbc}\fR
```

```
652 .ad
653 .sp .6
654 .RS 4n
655 An encryption algorithm, as in \fBipsecconf\fR(1M). However, of the ciphers
656 listed above, only \fBaes\fR and \fBaes-cbc\fR allow optional key-size setting,
657 using the "low value-to-high value" syntax. To specify a single AES key size,
658 the low value must equal the high value. If no range is specified, all three
659 AES key sizes are allowed.
660 .RE

662 .sp
663 .ne 2
664 .na
665 \fBauth_alg {md5, sha, sha1, sha256, sha384, sha512}\fR
666 .ad
667 .sp .6
668 .RS 4n
669 An authentication algorithm.
670 .sp
671 Use \fBipsecalgs\fR(1M) with the \fB-l\fR option to list the IPsec protocols
672 and algorithms currently defined on a system. The \fBcryptoadm list\fR command
673 displays a list of installed providers and their mechanisms. See
674 \fBcryptoadm\fR(1M).
675 .RE

677 .sp
678 .ne 2
679 .na
680 \fBauth_method {preshared, rsa_sig, rsa_encrypt, dss_sig}\fR
681 .ad
682 .sp .6
683 .RS 4n
684 The authentication method used for IKE phase 1.
685 .RE

687 .sp
688 .ne 2
689 .na
690 \fBp1_lifetime_secs \fInum\fR\fR
691 .ad
692 .sp .6
693 .RS 4n
694 Optional. The lifetime for a phase 1 SA.
695 .RE

697 .RE

699 .sp
700 .ne 2
701 .na
702 \fBp2_lifetime_secs \fInum\fR\fR
703 .ad
704 .sp .6
705 .RS 4n
706 If configuring the kernel defaults is not sufficient for different tasks, this
707 parameter can be used on a per-rule basis to set the IPsec \fBSA\fR lifetimes
708 in seconds.
709 .RE

711 .sp
712 .ne 2
713 .na
714 \fBp2_pfs \fInum\fR\fR
715 .ad
716 .sp .6
717 .RS 4n
```



```
718 Use perfect forward secrecy for phase 2 (quick mode). If selected, the oakley
719 group specified is used for phase 2 PFS. Acceptable values are:
720 .br
721 .in +2
722 0 (do not use Perfect Forward Secrecy for IPsec SAs)
723 .in -2
724 .br
725 .in +2
726 1 (768-bit)
727 .in -2
728 .br
729 .in +2
730 2 (1024-bit)
731 .in -2
732 .br
733 .in +2
734 5 (1536-bit)
735 .in -2
736 .br
737 .in +2
738 14 (2048-bit)
739 .in -2
740 .br
741 .in +2
742 15 (3072-bit)
743 .in -2
744 .br
745 .in +2
746 16 (4096-bit)
747 .in -2
748 .RE

750 .sp
751 .LP
752 An IKE rule starts with a right-curly-brace (\fB{\fR), ends with a
753 left-curly-brace (\fB}\fR), and has the following parameters in between:
754 .sp
755 .ne 2
756 .na
757 \fBlabel \fIstring\fR\fR
758 .ad
759 .sp .6
760 .RS 4n
761 Required parameter. The administrative interface to \fBin.iked\fR looks up
762 phase 1 policy rules with the label as the search string. The administrative
763 interface also converts the label into an index, suitable for an extended
764 ACQUIRE message from PF_KEY - effectively tying IPsec policy to IKE policy in
765 the case of a node initiating traffic. Only one \fBlabel\fR parameter is
766 allowed per rule.
767 .RE

769 .sp
770 .ne 2
771 .na
772 \fBlocal_addr <\fIIPaddr\fR/\fIprefix\fR/\fIrange\fR>\fR
773 .ad
774 .sp .6
775 .RS 4n
776 Required parameter. The local address, address prefix, or address range for
777 this phase 1 rule. Multiple \fBlocal_addr\fR parameters accumulate within a
778 given rule.
779 .RE

781 .sp
782 .ne 2
783 .na
```

```
\fBremote_addr <\fIIPaddr\fR/\fIprefix\fR/\fIrang\fRe>\fR
.ad
.sp .6
.RS 4n
Required parameter. The remote address, address prefix, or address range for
this phase 1 rule. Multiple \fBremote_addr\fR parameters accumulate within a
given rule.
.RE

.sp
.ne 2
.na
\fBlocal_id_type \fIp1-id-type\fR\fR
.ad
.sp .6
.RS 4n
Which phase 1 identity type I uses. This is needed because a single certificate
can contain multiple values for use in IKE phase 1. Within a given rule, all
phase 1 transforms must either use preshared or non-preshared authentication
(they cannot be mixed). For rules with preshared authentication, the
\fBlocal_id_type\fR parameter is optional, and defaults to \fBIP\fR. For rules
which use non-preshared authentication, the 'local_id_type' parameter is
required. Multiple 'local_id_type' parameters within a rule are not allowed.
.RE

.sp
.ne 2
.na
\fBlocal_id \fIcert-sel\fR\fR
.ad
.sp .6
.RS 4n
Disallowed for preshared authentication method; required parameter for
non-preshared authentication method. The local identity string or certificate
selector. Only one local identity per rule is used, the first one stated.
.RE

.sp
.ne 2
.na
\fBremote_id \fIcert-sel\fR\fR
.ad
.sp .6
.RS 4n
Disallowed for preshared authentication method; required parameter for
non-preshared authentication method. Selector for which remote phase 1
identities are allowed by this rule. Multiple \fBremote_id\fR parameters
accumulate within a given rule. If a single empty string (\fB""\fR) is given,
then this accepts any remote \fBID\fR for phase 1. It is recommended that
certificate trust chains or address enforcement be configured strictly to
prevent a breakdown in security if this value for \fBremote_id\fR is used.
.RE

.sp
.ne 2
.na
\fBp2_lifetime_secs \fInum\fR\fR
.ad
.sp .6
.RS 4n
If configuring the kernel defaults is not sufficient for different tasks, this
parameter can be used on a per-rule basis to set the IPsec \fBSA\fR lifetimes
in seconds.
.RE

.sp
```



```
.ne 2
.na
\fBp2_pfs \fInum\fR\fR
.ad
.sp .6
.RS 4n
Use perfect forward secrecy for phase 2 (quick mode). If selected, the oakley
group specified is used for phase 2 PFS. Acceptable values are:
.br
.in +2
0 (do not use Perfect Forward Secrecy for IPsec SAs)
.in -2
.br
.in +2
1 (768-bit)
.in -2
.br
.in +2
2 (1024-bit)
.in -2
.br
.in +2
5 (1536-bit)
.in -2
.br
.in +2
14 (2048-bit)
.in -2
.br
.in +2
15 (3072-bit)
.in -2
.br
.in +2
16 (4096-bit)
.in -2
.RE

.sp
.ne 2
.na
\fBp1_xform \fB{\fR \fIparameter-list\fR \fB}\fR\fR
.ad
.sp .6
.RS 4n
A phase 1 transform specifies a method for protecting an IKE phase 1 exchange.
An initiator offers up lists of phase 1 transforms, and a receiver is expected
to only accept such an entry if it matches one in a phase 1 rule. There can be
several of these, and they are additive. There must be either at least one
phase 1 transform in a rule or a global default phase 1 transform list. A
\fBike.config\fR file without a global default phase 1transform list \fBand\fR
a rule without a phase 1 transform list is an invalid file. Elements within the
parameter-list; unless specified as optional, must occur exactly once within a
given transform's parameter-list:
.sp
.ne 2
.na
\fBoakley_group \fInumber\fR\fR
.ad
.sp .6
.RS 4n
The Oakley Diffie-Hellman group used for \fBIKE SA\fR key derivation.
Acceptable values are currently:
.br
.in +2
1 (768-bit)
```

```
 916 .in -2
 917 .br
 918 .in +2
 919 2 (1024-bit)
 920 .in -2
 921 .br
 922 .in +2
 923 5 (1536-bit)
 924 .in -2
 925 .br
 926 .in +2
 927 14 (2048-bit)
 928 .in -2
 929 .br
 930 .in +2
 931 15 (3072-bit)
 932 .in -2
 933 .br
 934 .in +2
 935 16 (4096-bit)
 936 .in -2
 937 .RE

 939 .sp
 940 .ne 2
 941 .na
 942 \fBencr_alg {3des, 3des-cbc, blowfish, blowfish-cdc, des, des-cbc, aes,
 943 aes-cbc}\fR
 944 .ad
 945 .sp .6
 946 .RS 4n
 947 An encryption algorithm, as in \fBipsecconf\fR(1M). However, of the ciphers
 948 listed above, only \fBaes\fR and \fBaes-cbc\fR allow optional key-size setting,
 949 using the "low value-to-high value" syntax. To specify a single AES key size,
 950 the low value must equal the high value. If no range is specified, all three
 951 AES key sizes are allowed.
 952 .RE

 954 .sp
 955 .ne 2
 956 .na
 957 \fBauth_alg {md5, sha, sha1}\fR
 958 .ad
 959 .sp .6
 960 .RS 4n
 961 An authentication algorithm, as specified in \fBipseckey\fR(1M).
 962 .RE

 964 .sp
 965 .ne 2
 966 .na
 967 \fBauth_method {preshared, rsa_sig, rsa_encrypt, dss_sig}\fR
 968 .ad
 969 .sp .6
 970 .RS 4n
 971 The authentication method used for IKE phase 1.
 972 .RE

 974 .sp
 975 .ne 2
 976 .na
 977 \fBp1_lifetime_secs \fInum\fR\fR
 978 .ad
 979 .sp .6
 980 .RS 4n
 981 Optional. The lifetime for a phase 1 SA.
```



```
 982 .RE

 984 .RE

 986 .SH EXAMPLES
 987 .LP
 988 \fBExample 1 \fRA Sample \fBike.config\fR File
 989 .sp
 990 .LP
 991 The following is an example of an \fBike.config\fR file:

 993 .sp
 994 .in +2
 995 .nf

 997 ### BEGINNING OF FILE

 999 ### First some global parameters...

1001 ### certificate parameters...

1003 # Root certificates. I SHOULD use a full Distinguished Name.
1004 # I must have this certificate in my local filesystem, see ikecert(1m).
1005 cert_root    "C=US, O=Sun Microsystems\\, Inc., CN=Sun CA"

1007 # Explicitly trusted certs that need no signatures, or perhaps
1008 # self-signed ones. Like root certificates, use full DNs for them
1009 # for now.
1010 cert_trust    "EMAIL=root@domain.org"

1012 # Where do I send LDAP requests?
1013 ldap_server        "ldap1.domain.org,ldap2.domain.org:389"

1015 ## phase 1 transform defaults...

1017 p1_lifetime_secs 14400
1018 p1_nonce_len 20

1020 ## Parameters that might also show up in rules.

1022 p1_xform { auth_method preshared oakley_group 5 auth_alg sha
1023           encr_alg 3des }
1024 p2_pfs 2



1028 ### Now some rules...

1030 {
1031    label "simple inheritor"
1032    local_id_type ip
1033    local_addr 10.1.1.1
1034    remote_addr 10.1.1.2
1035 }
```

_____***unchanged_portion_omitted_***

```
**********************************************************
    4773 Fri Jan 11 21:14:04 2019
new/usr/src/man/man4/service_provider.conf.4
10067 Miscellaneous man page typos
Reviewed by: Robert Mustacchi <rm@joyent.com>
Reviewed by: Andy Fiddaman <andy@omniosce.org>
Reviewed by: Volker A. Brandt <vab@bb-c.de>
**********************************************************
```

```
   1 '\" te
   2 .\" Copyright (c) 2003, 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 SERVICE_PROVIDER.CONF 4 "Jun 18, 2004"
   7 .SH NAME
   8 service_provider.conf \- service provider configuration file
   9 .SH SYNOPSIS
  10 .LP
  11 .nf
  12 \fBservice_provider.conf\fR
  13 .fi

  15 .SH DESCRIPTION
  16 .sp
  16 .LP
  17 \fBservice_provider.conf\fR contains information about the device type that the
  18 service provider supports. This information includes the pathname of the
  19 service provider library, the library version and other library characteristics
  20 that are required by the system administrative command, \fBdatadm\fR(1M).
  21 \fBdatadm\fR(1M) puts this information in the DAT static register file,
  22 \fBdat.conf\fR(4).
  23 .sp
  24 .LP
  25 The \fBdatadm\fR program enumerates each device entry into a list of interface
  26 adapters, that is, interfaces to external network that are available to uDAPL
  27 consumers. This new list of interface adapters is appended to other service
  28 providers' information in the DAT static registry, \fBdat.conf\fR. You can do
  29 this if you invoke the \fBdatadm\fR program with the \fB-a\fR option and the
  30 this is you invoke the \fBdatadm\fR program with the \fB-a\fR option and the
  30 pathname of the \fBservice_provider.conf\fR file.
  31 .sp
  32 .LP
  33 Each entry in the service_provider.conf is a single line of 7 fields.
  34 .sp
  35 .LP
  36 The following shows the order of the fields in a \fBservice_provider.conf\fR
  37 entry:
  38 .sp
  39 .in +2
  40 .nf
  41 "\fIdriver_name\fR" "\fIAPI_version\fR" "\fIthreadsafe_library\fR | \e
  42       \fInonthreadsafe_library\fR"\e
  43       \fInonthreadsafe_librar\fRy"\e
  43 "\fIdefault_version\fR | \fInondefault_version\fR" \e
  44     "\fIservice_provider_library_pathname\fR"\e
  45 "\fIservice_provider_version\fR" "\fIservice_provider_instance_data\fR"\e
  46 .fi
  47 .in -2

  49 .sp
  50 .LP
  51 The fields are defined as follows:
  52 .sp
  53 .ne 2
  54 .na
  55 \fB\fIdriver_name\fR\fR
```



```
  56 .ad
  57 .sp .6
  58 .RS 4n
  59 Specifies a driver name in the format of \fBdriver_name\fR=\fIvalue pair\fR,
  60 for example, \fBdriver_name=tavor\fR.
  61 .RE

  63 .sp
  64 .ne 2
  65 .na
  66 \fB\fIAPI_version\fR\fR
  67 .ad
  68 .sp .6
  69 .RS 4n
  70 Specifies the API version of the service provider library: For example,
  71 Specifies the API version of the service provide library: For example,
  71 \fB"u"major.minor\fR is \fBu1.2\fR.
  72 .RE

  74 .sp
  75 .ne 2
  76 .na
  77 \fB\fIthreadsafe_library\fR | \fInonthreadsafe_library\fR\fR
  78 \fB\fIthreadsafe_library\fR | \fInonthreadsafe_librar\fR\fR
  78 .ad
  79 .sp .6
  80 .RS 4n
  81 Specifies a threadsafe or non-threadsafe library.
  82 .RE

  84 .sp
  85 .ne 2
  86 .na
  87 \fB\fIdefault_version\fR | \fInondefault_version\fR\fR
  88 .ad
  89 .sp .6
  90 .RS 4n
  91 Specifies a default or non-default version of a library. A service provider can
  92 Specifies a default or non-default version of library. A service provider can
  92 offer several versions of the library. If so, one version is designated as
  93 \fBdefault\fR with the rest as \fBnondefault\fR.
  94 .RE

  96 .sp
  97 .ne 2
  98 .na
  99 \fB\fIservice_provider_library_pathname\fR\fR
 100 .ad
 101 .sp .6
 102 .RS 4n
 103 Specifies the pathname of the library image.
 104 .RE

 106 .sp
 107 .ne 2
 108 .na
 109 \fB\fIservice_provider_version\fR\fR
 110 .ad
 111 .sp .6
 112 .RS 4n
 113 Specifies the version of the service provider. By convention, specify the
 114 company stock symbol as the service provider, followed by major and minor
 115 version numbers, for example, \fBSUNW1.0\fR.
 116 .RE

 118 .sp
```

```
.ne 2
.na
\fB\fIservice_provider_instance_data\fR\fR
.ad
.sp .6
.RS 4n
Specifies the service provider instance data.
.RE

.SH EXAMPLES
.LP
\fBExample 1 \fRUsing a Logical Device Name
.sp
.LP
The following example \fBservice_provider.conf\fR entry uses a logical device
name:

.sp
.in +2
.nf
#
# Sample service_provider.conf entry showing an uDAPL 1.2 service
# provider, udapl_tavor.so.1 supporting a device with a driver named
# tavor
driver_name=tavor u1.2 nonthreadsafe default udapl_tavor.so.1 \e
    SUNW.1.0 ""
.fi
.in -2

.LP
\fBExample 2 \fRUsing a Physical Device Name
.sp
.LP
The following example \fBservice_provider.conf\fR uses a physical device name:

.sp
.in +2
.nf
#
# Sample service_provider.conf entry showing an uDAPL 1.2
# service provider, udapl_tavor.so.1 supporting a device named
# pci15b3,5a44 that can be located under /devices
#
pci15b3,5a44 u1.2 nonthreadsafe default \e
    /usr/lib/tavor/udapl_tavor.so.1 SUNWudaplt1.0 ""
.fi
.in -2

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

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
Stability       Evolving
.TE

.SH SEE ALSO
.sp
.LP
```



```
\fBdatadm\fR(1M), \fBdat.conf\fR(4), \fBattributes\fR(5)
```