```
**********************************************************
    1815 Thu Jul 17 00:50:38 2014
new/usr/src/man/Makefile
manpage lint.
**********************************************************
#
# This file and its contents are supplied under the terms of the
# Common Development and Distribution License ("CDDL"), version 1.0.
# You may only use this file in accordance with the terms of version
# 1.0 of the CDDL.
#
# A full copy of the text of the CDDL should have accompanied this
# source.  A copy of the CDDL is also available via the Internet
# at http://www.illumos.org/license/CDDL.
#

#
# Copyright 2011, Richard Lowe
# Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
#

SUBDIRS=        man1            \
                man1b           \
                man1c           \
                man1has         \
                man1m           \
                man2            \
                man3            \
                man3bsm         \
                man3c           \
                man3c_db        \
                man3cfgadm      \
                man3commputil   \
                man3contract    \
                man3cpc         \
                man3curses      \
                man3dat         \
                man3devid       \
                man3devinfo     \
                man3dlpi        \
                man3dns_sd      \
                man3elf         \
                man3exacct      \
                man3ext         \
                man3fcoe        \
                man3fstyp       \
                man3gen         \
                man3gss         \
                man3head        \
                man3iscsit      \
                man3kstat       \
                man3kvm         \
                man3ldap        \
                man3lgrp        \
                man3lib         \
                man3mail        \
                man3malloc      \
                man3mp          \
                man3mpapi       \
                man3nsl         \
                man3nvpair      \
                man3pam         \
                man3papi        \
                man3perl        \
                man3picl        \
                man3picltree    \
```



```
                man3pool        \
                man3proc        \
                man3project     \
                man3resolv      \
                man3rpc         \
                man3rsm         \
                man3sasl        \
                man3scf         \
                man3sec         \
                man3secdb       \
                man3sip         \
                man3slp         \
                man3socket      \
                man3stmf        \
                man3sysevent    \
                man3tecla       \
                man3tnf         \
                man3tsol        \
                man3uuid        \
                man3volmgt      \
                man3xcurses     \
                man3xnet        \
                man4            \
                man5            \
                man7            \
                man7d           \
                man7fs          \
                man7i           \
                man7ipp         \
                man7m           \
                man7p           \
                man9            \
                man9e           \
                man9f           \
                man9p           \
                man9s

.PARALLEL: $(SUBDIRS)

all             := TARGET = all
clean           := TARGET = clean
clobber         := TARGET = clobber
install         := TARGET = install
check           := TARGET = check

all check clean clobber install: $(SUBDIRS)
all clean clobber install: $(SUBDIRS)

$(SUBDIRS):     FRC
        @cd $@; pwd; $(MAKE) $(TARGET)

FRC:
```

```
**********************************************************
    1239 Thu Jul 17 00:50:38 2014
new/usr/src/man/Makefile.man
manpage lint.
**********************************************************
```

```
   1 #
   2 # This file and its contents are supplied under the terms of the
   3 # Common Development and Distribution License ("CDDL"), version 1.0.
   4 # You may only use this file in accordance with the terms of version
   5 # 1.0 of the CDDL.
   6 #
   7 # A full copy of the text of the CDDL should have accompanied this
   8 # source.  A copy of the CDDL is also available via the Internet
   9 # at http://www.illumos.org/license/CDDL.
  10 #

  12 #
  13 # Copyright 2011, Richard Lowe
  14 # Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
  15 #

  17 MANDOC=          $(ROOT)/usr/bin/mandoc
  18 ROOTMAN=         $(ROOT)/usr/share/man
  19 ROOTHASMAN=      $(ROOT)/usr/has/man

  20 FILEMODE=        0444

  22 # The manual section being built, client Makefiles must set this to, for e.g.
  23 # "3perl", with case matching that of the section name as installed.
  24 #
  25 # MANSECT=

  27 MANCHECKS=       $(MANFILES:%=%.check)
  28 ROOTMANFILES=    $(MANFILES:%=$(ROOTMAN)/man$(MANSECT)/%)
  29 ROOTMANLINKS=    $(MANLINKS:%=$(ROOTMAN)/man$(MANSECT)/%)

  31 $(ROOTMAN)/man$(MANSECT)/% $(ROOTHASMAN)/man$(MANSECT)/%: %
  32                 $(INS.file)

  34 $(MANCHECKS):
  35                 $(MANDOC) -Tlint $(@:%.check=%)

  37 $(MANLINKS):
  38                 $(RM) $@; $(SYMLINK) $(LINKSRC) $@

  40 $(ROOTMANLINKS): $(MANLINKS)
  41                 $(RM) $@; $(CP) -RP $(@F) $(@D)

  43 all:

  45 check:          $(MANCHECKS)

  47 clean:

  49 clobber:
  50                 $(RM) $(MANLINKS)

  52 .PARALLEL:

  54 FRC:
```

```
**********************************************************
    8110 Thu Jul 17 00:50:38 2014
new/usr/src/man/man1/head.1
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
   59044 Thu Jul 17 00:50:38 2014
new/usr/src/man/man1/ld.1
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
    3098 Thu Jul 17 00:50:39 2014
new/usr/src/man/man1/man.1
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
   14742 Thu Jul 17 00:50:39 2014
new/usr/src/man/man1/mandoc.1
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
   22738 Thu Jul 17 00:50:39 2014
new/usr/src/man/man1/printf.1
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
   34827 Thu Jul 17 00:50:39 2014
new/usr/src/man/man1/tar.1
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
    4623 Thu Jul 17 00:50:39 2014
new/usr/src/man/man1m/arcstat.1m
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
   17974 Thu Jul 17 00:50:39 2014
new/usr/src/man/man1m/dd.1m
manpage lint.
**********************************************************
   1 '\" te
   2 .\" Copyright (c) 2014, Joyent, Inc.  All rights Reserved.
   3 .\" Copyright (c) 1992, X/Open Company Limited  All Rights Reserved
   4 .\" Copyright 1989 AT&T
   5 .\" Portions Copyright (c) 1995, Sun Microsystems, Inc.  All Rights Reserved
   6 .\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for permission
   7 .\" http://www.opengroup.org/bookstore/.
   8 .\" The Institute of Electrical and Electronics Engineers and The Open Group, ha
   9 .\"  This notice shall appear on any product containing this material.
  10 .\" The contents of this file are subject to the terms of the Common Development
  11 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
  12 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
  13 .TH DD 1M "Jan 04, 2014"
  14 .SH NAME
  15 dd \- convert and copy a file
  16 .SH SYNOPSIS
  17 .LP
  18 .nf
  19 \fB/usr/bin/dd\fR [\fIoperand=value\fR]...
  20 .fi

  22 .SH DESCRIPTION
  23 .sp
  24 .LP
  25 The \fBdd\fR utility copies the specified input file to the specified output
  26 with possible conversions. The standard input and output are used by default.
  27 The input and output block sizes may be specified to take advantage of raw
  28 physical I/O. Sizes are specified in bytes; a number may end with \fBk\fR,
  29 \fBb\fR, or \fBw\fR to specify multiplication by 1024, 512, or 2, respectively.
  30 Numbers may also be separated by \fBx\fR to indicate multiplication.
  31 .sp
  32 .LP
  33 The \fBdd\fR utility reads the input one block at a time, using the specified
  34 input block size. \fBdd\fR then processes the block of data actually returned,
  35 which could be smaller than the requested block size. \fBdd\fR applies any
  36 conversions that have been specified and writes the resulting data to the
  37 output in blocks of the specified output block size.
  38 .sp
  39 .LP
  40 \fBcbs\fR is used only if \fBascii\fR, \fBasciib\fR, \fBunblock\fR,
  41 \fBebcdic\fR, \fBebcdicb\fR, \fBibm\fR, \fBibmb\fR, or \fBblock\fR conversion
  42 is specified. In the first two cases, \fBcbs\fR characters are copied into the
  43 conversion buffer, any specified character mapping is done, trailing blanks are
  44 trimmed, and a \fBNEWLINE\fR is added before sending the line to output. In the
  45 last three cases, characters up to \fBNEWLINE\fR are read into the conversion
  46 buffer and blanks are added to make up an output record of size \fBcbs\fR.
  47 \fBASCII\fR files are presumed to contain \fBNEWLINE\fR characters. If
  48 \fBcbs\fR is unspecified or \fB0\fR, the \fBascii\fR, \fBasciib\fR,
  49 \fBebcdic\fR, \fBebcdicb\fR, \fBibm\fR, and \fBibmb\fR options convert the
  50 character set without changing the input file's block structure. The
  51 \fBunblock\fR and \fBblock\fR options become a simple file copy.
  52 .sp
  53 .LP
  54 After completion, \fBdd\fR reports the number of whole and partial input and
  55 output blocks.
  56 .SH OPERANDS
  57 .sp
  58 .LP
  59 The following operands are supported:
  60 .sp
  61 .ne 2
```



```
  62 .na
  63 \fB\fBif=\fR\fIfile\fR\fR
  64 .ad
  65 .sp .6
  66 .RS 4n
  67 Specifies the input path. Standard input is the default.
  68 .RE

  70 .sp
  71 .ne 2
  72 .na
  73 \fB\fBof=\fR\fIfile\fR\fR
  74 .ad
  75 .sp .6
  76 .RS 4n
  77 Specifies the output path. Standard output is the default. If the
  78 \fBseek=\fR\fBexpr\fR conversion is not also specified, the output file will be
  79 truncated before the copy begins, unless \fBconv=notrunc\fR is specified. If
  80 \fBseek=\fR\fBexpr\fR is specified, but \fBconv=notrunc\fR is not, the effect
  81 of the copy will be to preserve the blocks in the output file over which
  82 \fBdd\fR seeks, but no other portion of the output file will be preserved. (If
  83 the size of the seek plus the size of the input file is less than the previous
  84 size of the output file, the output file is shortened by the copy.)
  85 .RE

  87 .sp
  88 .ne 2
  89 .na
  90 \fB\fBibs=\fR\fIn\fR\fR
  91 .ad
  92 .sp .6
  93 .RS 4n
  94 Specifies the input block size in \fIn\fR bytes (default is \fB512\fR).
  95 .RE

  97 .sp
  98 .ne 2
  99 .na
 100 \fB\fBobs=\fR\fIn\fR\fR
 101 .ad
 102 .sp .6
 103 .RS 4n
 104 Specifies the output block size in \fIn\fR bytes (default is \fB512\fR).
 105 .RE

 107 .sp
 108 .ne 2
 109 .na
 110 \fB\fBbs=\fR\fIn\fR\fR
 111 .ad
 112 .sp .6
 113 .RS 4n
 114 Sets both input and output block sizes to \fIn\fR bytes, superseding \fBibs=\fR
 115 and \fBobs=\fR. If no conversion other than \fBsync\fR,\fB noerror\fR, and
 116 \fBnotrunc\fR is specified, each input block is copied to the output as a
 117 single block without aggregating short blocks.
 118 .RE

 120 .sp
 121 .ne 2
 122 .na
 123 \fB\fBcbs=\fR\fIn\fR\fR
 124 .ad
 125 .sp .6
 126 .RS 4n
 127 Specifies the conversion block size for \fBblock\fR and \fBunblock\fR in bytes
```

```
by \fIn\fR (default is \fB0\fR). If \fBcbs=\fR is omitted or given a value of
\fB0\fR, using \fBblock\fR or \fBunblock\fR produces unspecified results.
.sp
This option is used only if \fBASCII\fR or \fBEBCDIC\fR conversion is
specified. For the \fBascii\fR and \fBasciib\fR operands, the input is handled
as described for the \fBunblock\fR operand except that characters are converted
to \fBASCII\fR before the trailing \fBSPACE\fR characters are deleted. For the
\fBebcdic\fR, \fBebcdicb\fR, \fBibm\fR, and \fBibmb\fR operands, the input is
handled as described for the \fBblock\fR operand except that the characters are
converted to \fBEBCDIC\fR or IBM \fBEBCDIC\fR after the trailing \fBSPACE\fR
characters are added.
.RE

.sp
.ne 2
.na
\fB\fBfiles=\fR\fIn\fR\fR
.ad
.sp .6
.RS 4n
Copies and concatenates \fIn\fR input files before terminating (makes sense
only where input is a magnetic tape or similar device).
.RE

.sp
.ne 2
.na
\fB\fBskip=\fR\fIn\fR\fR
.ad
.sp .6
.RS 4n
Skips \fIn\fR input blocks (using the specified input block size) before
starting to copy. On seekable files, the implementation reads the blocks or
seeks past them. On non-seekable files, the blocks are read and the data is
discarded.
.RE

.sp
.ne 2
.na
\fB\fBiseek=\fR\fIn\fR\fR
.ad
.sp .6
.RS 4n
Seeks \fIn\fR blocks from beginning of input file before copying (appropriate
for disk files, where \fBskip\fR can be incredibly slow).
.RE

.sp
.ne 2
.na
\fB\fBoseek=\fR\fIn\fR\fR
.ad
.sp .6
.RS 4n
Seeks \fIn\fR blocks from beginning of output file before copying.
.RE

.sp
.ne 2
.na
\fB\fBseek=\fR\fIn\fR\fR
.ad
.sp .6
.RS 4n
Skips \fIn\fR blocks (using the specified output block size) from beginning of
```



```
output file before copying. On non-seekable files, existing blocks are read and
space from the current end-of-file to the specified offset, if any, is filled
with null bytes. On seekable files, the implementation seeks to the specified
offset or reads the blocks as described for non-seekable files.
.RE

.sp
.ne 2
.na
\fB\fBcount=\fR\fIn\fR\fR
.ad
.sp .6
.RS 4n
Copies only \fIn\fR input blocks.
.RE

.sp
.ne 2
.na
\fB\fBconv=\fR\fIvalue\fR[\fB,\fR\fIvalue\fR.\|.\|.\|]\fR
.ad
.sp .6
.RS 4n
Where \fIvalue\fRs are comma-separated symbols from the following list:
.sp
.ne 2
.na
\fB\fBascii\fR\fR
.ad
.RS 11n
Converts \fBEBCDIC\fR to \fBASCII\fR.
.RE

.sp
.ne 2
.na
\fB\fBasciib\fR\fR
.ad
.RS 11n
Converts \fBEBCDIC\fR to \fBASCII\fR using \fBBSD\fR-compatible character
translations.
.RE

.sp
.ne 2
.na
\fB\fBebcdic\fR\fR
.ad
.RS 11n
Converts \fBASCII\fR to \fBEBCDIC\fR. If converting fixed-length \fBASCII\fR
records without NEWLINEs, sets up a pipeline with \fBdd conv=unblock\fR
beforehand.
.RE

.sp
.ne 2
.na
\fB\fBebcdicb\fR\fR
.ad
.RS 11n
Converts \fBASCII\fR to \fBEBCDIC\fR using \fBBSD\fR-compatible character
translations. If converting fixed-length \fBASCII\fR records without
\fBNEWLINE\fRs, sets up a pipeline with \fBdd conv=unblock\fR beforehand.
.RE

.sp
```

```
260 .ne 2
261 .na
262 \fB\fBibm\fR\fR
263 .ad
264 .RS 11n
265 Slightly different map of \fBASCII\fR to \fBEBCDIC\fR. If converting
266 fixed-length \fBASCII\fR records without \fBNEWLINE\fRs, sets up a pipeline
267 with \fBdd conv=unblock\fR beforehand.
268 .RE

270 .sp
271 .ne 2
272 .na
273 \fB\fBibmb\fR\fR
274 .ad
275 .RS 11n
276 Slightly different map of \fBASCII\fR to \fBEBCDIC\fR using
277 \fBBSD\fR-compatible character translations. If converting fixed-length
278 \fBASCII\fR records without \fBNEWLINE\fRs, sets up a pipeline with \fBdd
279 conv=unblock\fR beforehand.
280 .RE

282 The \fBascii\fR (or \fBasciib\fR), \fBebcdic\fR (or \fBebcdicb\fR), and
283 \fBibm\fR (or \fBibmb\fR) values are mutually exclusive.
284 .sp
285 .ne 2
286 .na
287 \fB\fBblock\fR\fR
288 .ad
289 .RS 11n
290 Treats the input as a sequence of \fBNEWLINE\fR-terminated or
291 \fBEOF\fR-terminated variable-length records independent of the input block
292 boundaries. Each record is converted to a record with a fixed length specified
293 by the conversion block size. Any \fBNEWLINE\fR character is removed from the
294 input line. \fBSPACE\fR characters are appended to lines that are shorter than
295 their conversion block size to fill the block. Lines that are longer than the
296 conversion block size are truncated to the largest number of characters that
297 will fit into that size. The number of truncated lines is reported.
298 .RE

300 .sp
301 .ne 2
302 .na
303 \fB\fBunblock\fR\fR
304 .ad
305 .RS 11n
306 Converts fixed-length records to variable length. Reads a number of bytes equal
307 to the conversion block size (or the number of bytes remaining in the input, if
308 less than the conversion block size), delete all trailing \fBSPACE\fR
309 characters, and append a \fBNEWLINE\fR character.
310 .RE

312 The  \fBblock\fR and \fBunblock\fR values are mutually exclusive.
313 .sp
314 .ne 2
315 .na
316 \fB\fBlcase\fR\fR
317 .ad
318 .RS 9n
319 Maps upper-case characters specified by the \fBLC_CTYPE\fR keyword
320 \fBtolower\fR to the corresponding lower-case character. Characters for which
321 no mapping is specified are not modified by this conversion.
322 .RE

324 .sp
325 .ne 2
```



```
326 .na
327 \fB\fBucase\fR\fR
328 .ad
329 .RS 9n
330 Maps lower-case characters specified by the \fBLC_CTYPE\fR keyword
331 \fBtoupper\fR to the corresponding upper-case character. Characters for which
332 no mapping is specified are not modified by this conversion.
333 .RE

335 The \fBlcase\fR and \fBucase\fR symbols are mutually exclusive.
336 .sp
337 .ne 2
338 .na
339 \fB\fBswab\fR\fR
340 .ad
341 .RS 11n
342 Swaps every pair of input bytes. If the current input record is an odd number
343 of bytes, the last byte in the input record is ignored.
344 .RE

346 .sp
347 .ne 2
348 .na
349 \fB\fBnoerror\fR\fR
350 .ad
351 .RS 11n
352 Does not stop processing on an input error. When an input error occurs, a
353 diagnostic message is written on standard error, followed by the current input
354 and output block counts in the same format as used at completion. If the
355 \fBsync\fR conversion is specified, the missing input is replaced with null
356 bytes and processed normally. Otherwise, the input block will be omitted from
357 the output.
358 .RE

360 .sp
361 .ne 2
362 .na
363 \fB\fBnotrunc\fR\fR
364 .ad
365 .RS 11n
366 Does not truncate the output file. Preserves blocks in the output file not
367 explicitly written by this invocation of \fBdd\fR. (See also the preceding
368 \fBof=\fR\fIfile\fR operand.)
369 .RE

371 .sp
372 .ne 2
373 .na
374 \fB\fBsync\fR\fR
375 .ad
376 .RS 11n
377 Pads every input block to the size of the \fBibs=\fR buffer, appending null
378 bytes. (If either \fBblock\fR or \fBunblock\fR is also specified, appends
379 \fBSPACE\fR characters, rather than null bytes.)
380 .RE

382 .RE

384 .sp
385 .ne 2
386 .na
387 \fB\fBoflag=\fR\fIvalue\fR[\fB,\fR\fIvalue\fR.\|.\|.\|]\fR
388 .ad
389 .sp .6
390 Where \fIvalue\fRs are comma-separated symbols from the following list which
391 affect the behavior of writing the output file:
```

```
392 .sp
393 .ne 2
394 .na
395 \fB\fBdsync\fR\fR
396 .ad
397 .RS 11n
398 The output file is opened with the \fBO_DSYNC\fR flag set. All data writes will
399 be synchronous. For more information on \fBO_DSYNC\fR see \fBfcntl.h\fR(3HEAD).
400 .RE

402 .sp
403 .ne 2
404 .na
405 \fB\fBsync\fR\fR
406 .ad
407 .RS 11n
408 The output file is opened with the \fBO_SYNC\fR flag set. All data and metadata
409 writes will be synchronous. For more information on \fBO_SYNC\fR see
410 \fBfcntl.h\fR(3HEAD).
411 .RE

413 .RE

413 .sp
414 .LP
415 If operands other than \fBconv=\fR and \fBoflag=\fR are specified more than once
416 the last specified \fBoperand=\fR\fIvalue\fR is used.
417 .sp
418 .LP
419 For the \fBbs=\fR, \fBcbs=\fR, \fBibs=\fR, and \fBobs=\fR operands, the
420 application must supply an expression specifying a size in bytes. The
421 expression, \fBexpr\fR, can be:
422 .RS +4
423 .TP
424 1.
425 a positive decimal number
426 .RE
427 .RS +4
428 .TP
429 2.
430 a positive decimal number followed by \fBk\fR, specifying multiplication by
431 1024
432 .RE
433 .RS +4
434 .TP
435 3.
436 a positive decimal number followed by \fBM\fR, specifying multiplication by
437 1024*1024
438 .RE
439 .RS +4
440 .TP
441 4.
442 a positive decimal number followed by \fBG\fR, specifying multiplication by
443 1024*1024*1024
444 .RE
445 .RS +4
446 .TP
447 5.
448 a positive decimal number followed by \fBT\fR, specifying multiplication by
449 1024*1024*1024*1024
450 .RE
451 .RS +4
452 .TP
453 6.
454 a positive decimal number followed by \fBP\fR, specifying multiplication by
455 1024*1024*1024*1024*1024
```



```
456 .RE
457 .RS +4
458 .TP
459 7.
460 a positive decimal number followed by \fBE\fR, specifying multiplication by
461 1024*1024*1024*1024*1024*1024
462 .RE
463 .RS +4
464 .TP
465 8.
466 a positive decimal number followed by \fBZ\fR, specifying multiplication by
467 1024*1024*1024*1024*1024*1024*1024
468 .RE
469 .RS +4
470 .TP
471 9.
472 a positive decimal number followed by \fBb\fR, specifying multiplication by
473 512
474 .RE
475 .RS +4
476 .TP
477 10.
478 two or more positive decimal numbers (with or without \fBk\fR or \fBb\fR)
479 separated by \fBx\fR, specifying the product of the indicated values.
480 .RE
481 .sp
482 .LP
483 All of the operands will be processed before any input is read.
484 .SH SIGNALS
485 .sp
486 .LP
487 When \fBdd\fR receives either SIGINFO or SIGUSR1, \fBdd\fR will emit the current
488 input and output block counts, total bytes written, total time elapsed, and the
489 number of bytes per second to standard error. This is the same information
490 format that \fBdd\fR emits when it successfully completes. Users may send
491 SIGINFO via their terminal. The default character is ^T, see \fBstty\fR(1) for
492 more information.
493 .sp
494 .LP
495 For \fBSIGINT\fR, \fBdd\fR writes status information to standard error before
496 exiting. \fBdd\fR takes the standard action for all other signals.

498 .SH USAGE
499 .sp
500 .LP
501 See \fBlargefile\fR(5) for the description of the behavior of \fBdd\fR when
502 encountering files greater than or equal to 2 Gbyte ( 2^31 bytes).
503 .SH EXAMPLES
504 .LP
505 \fBExample 1 \fRCopying from one tape drive to another
506 .sp
507 .LP
508 The following example copies from tape drive \fB0\fR to tape drive \fB1\fR,
509 using a common historical device naming convention.

511 .sp
512 .in +2
513 .nf
514 example% \fBdd if=/dev/rmt/0h  of=/dev/rmt/1h\fR
515 .fi
516 .in -2
517 .sp

519 .LP
520 \fBExample 2 \fRStripping the first 10 bytes from standard input
521 .sp
```

```
522 .LP
523 The following example strips the first 10 bytes from standard input:

525 .sp
526 .in +2
527 .nf
528 example% \fBdd ibs=10  skip=1\fR
529 .fi
530 .in -2
531 .sp

533 .LP
534 \fBExample 3 \fRReading a tape into an ASCII file
535 .sp
536 .LP
537 This example reads an \fBEBCDIC\fR tape blocked ten 80-byte \fBEBCDIC\fR card
538 images per block into the \fBASCII\fR file \fBx\fR:

540 .sp
541 .in +2
542 .nf
543 example% \fBdd if=/dev/tape of=x ibs=800 cbs=80 conv=ascii,lcase\fR
544 .fi
545 .in -2
546 .sp

548 .LP
549 \fBExample 4 \fRUsing conv=sync to write to tape
550 .sp
551 .LP
552 The following example uses \fBconv=sync\fR when writing to a tape:

554 .sp
555 .in +2
556 .nf
557 example% \fBtar cvf - . | compress | dd obs=1024k of=/dev/rmt/0 conv=sync\fR
558 .fi
559 .in -2
560 .sp

562 .SH ENVIRONMENT VARIABLES
563 .sp
564 .LP
565 See \fBenviron\fR(5) for descriptions of the following environment variables
566 that affect the execution of \fBdd\fR: \fBLANG\fR, \fBLC_ALL\fR,
567 \fBLC_CTYPE\fR, \fBLC_MESSAGES\fR, and \fBNLSPATH\fR.
568 .SH EXIT STATUS
569 .sp
570 .LP
571 The following exit values are returned:
572 .sp
573 .ne 2
574 .na
575 \fB\fB0\fR\fR
576 .ad
577 .RS 6n
578 The input file was copied successfully.
579 .RE

581 .sp
582 .ne 2
583 .na
584 \fB\fB>0\fR\fR
585 .ad
586 .RS 6n
587 An error occurred.
```



```
588 .RE

590 .sp
591 .LP
592 If an input error is detected and the \fBnoerror\fR conversion has not been
593 specified, any partial output block will be written to the output file, a
594 diagnostic message will be written, and the copy operation will be
595 discontinued. If some other error is detected, a diagnostic message will be
596 written and the copy operation will be discontinued.
597 .SH ATTRIBUTES
598 .sp
599 .LP
600 See \fBattributes\fR(5) for descriptions of the following attributes:
601 .sp

603 .sp
604 .TS
605 box;
606 c | c
607 l | l .
608 ATTRIBUTE TYPE  ATTRIBUTE VALUE
609 _
610 Interface Stability     Standard
611 .TE

613 .SH SEE ALSO
614 .sp
615 .LP
616 \fBcp\fR(1), \fBsed\fR(1), \fBtr\fR(1), \fBfcntl.h\fR(3HEAD),
617 \fBattributes\fR(5), \fBenviron\fR(5), \fBlargefile\fR(5), \fBstandards\fR(5)
618 .SH DIAGNOSTICS
619 .sp
620 .ne 2
621 .na
622 \fB\fBf+p records in(out)\fR\fR
623 .ad
624 .RS 23n
625 numbers of full and partial blocks read(written)
626 .RE

628 .SH NOTES
629 .sp
630 .LP
631 Do not use \fBdd\fR to copy files between file systems having different block
632 sizes.
633 .sp
634 .LP
635 Using a  blocked device to copy a file will result in extra nulls being added
636 to the file to pad the final block to the block boundary.
637 .sp
638 .LP
639 When  \fBdd\fR reads from a pipe, using the  \fBibs=X\fR and  \fBobs=Y\fR
640 operands, the output will always be blocked in chunks of size Y. When
641 \fBbs=Z\fR is used, the output blocks will be whatever was available to be read
642 from the pipe at the time.
643 .sp
644 .LP
645 When using \fBdd\fR to copy files to a tape device, the file size must be a
646 multiple of the device sector size (for example, 512 Kbyte).  To  copy files of
647 arbitrary size to a tape device, use  \fBtar\fR(1) or  \fBcpio\fR(1).
```

```
**********************************************************
   22252 Thu Jul 17 00:50:39 2014
new/usr/src/man/man1m/ipadm.1m
manpage lint.
**********************************************************
'\" te
.\" Copyright (c) 2012, Joyent, Inc. All Rights Reserved
.\" Copyright (c) 2013 by Delphix. 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 IPADM 1M "May 14, 2012"
.SH NAME
ipadm \- configure IP network interfaces and protocol properties.
.SH SYNOPSIS
.LP
.nf
\fBipadm\fR create-if [\fB-t\fR] \fIinterface\fR
.fi

.LP
.nf
\fBipadm\fR disable-if [\fB-t\fR] \fIinterface\fR
.fi

.LP
.nf
\fBipadm\fR enable-if [\fB-t\fR] \fIinterface\fR
.fi

.LP
.nf
\fBipadm\fR delete-if \fIinterface\fR
.fi

.LP
.nf
\fBipadm\fR show-if [[\fB-p\fR] \fB-o\fR \fIfield\fR[,...]] [\fIinterface\fR]
.fi

.LP
.nf
\fBipadm\fR set-ifprop [\fB-t\fR] \fB-p\fR \fIprop\fR=<\fIvalue\fR[,...]> \fB-m\
.fi

.LP
.nf
\fBipadm\fR reset-ifprop [\fB-t\fR] \fB-p\fR \fIprop\fR \fB-m\fR \fIprotocol\fR
.fi

.LP
.nf
\fBipadm\fR show-ifprop [[\fB-c\fR]\fB-o\fR \fIfield\fR[,...]] [\fB-p\fR \fIprop
  [\fIinterface\fR]
.fi

.LP
.nf
\fBipadm\fR create-addr [\fB-t\fR] \fB-T\fR static [\fB-d\fR]
   \fB-a\fR {local|remote}=\fIaddr\fR[/\fIprefixlen\fR],... \fIaddrobj\fR
.fi

.LP
.nf
\fBipadm\fR create-addr [\fB-t\fR] \fB-T\fR dhcp [\fB-w\fR \fIseconds\fR | forev
.fi
```



```
.LP
.nf
\fBipadm\fR create-addr [\fB-t\fR] \fB-T\fR addrconf [\fB-i\fR \fIinterface-id\f
   [\fB-p\fR {stateful|stateless}={yes|no},..] \fIaddrobj\fR
.fi

.LP
.nf
\fBipadm\fR down-addr [\fB-t\fR] \fIaddrobj\fR
.fi

.LP
.nf
\fBipadm\fR up-addr [\fB-t\fR] \fIaddrobj\fR
.fi

.LP
.nf
\fBipadm\fR disable-addr [\fB-t\fR] \fIaddrobj\fR
.fi

.LP
.nf
\fBipadm\fR enable-addr [\fB-t\fR] \fIaddrobj\fR
.fi

.LP
.nf
\fBipadm\fR refresh-addr [\fB-i\fR] \fIaddrobj\fR
.fi

.LP
.nf
\fBipadm\fR delete-addr [\fB-r\fR] \fIaddrobj\fR
.fi

.LP
.nf
\fBipadm\fR show-addr [[\fB-p\fR] \fB-o\fR \fIfield\fR[,...]] [\fIaddrobj\fR]
.fi

.LP
.nf
\fBipadm\fR set-addrprop [\fB-t\fR] \fB-p\fR \fIprop\fR=<\fIvalue\fR[,...]> \fIa
.fi

.LP
.nf
\fBipadm\fR reset-addrprop [\fB-t\fR] \fB-p\fR \fIprop\fR=<\fIvalue\fR[,...]> \f
.fi

.LP
.nf
\fBipadm\fR show-addrprop [[\fB-c\fR] \fB-o\fR \fIfield\fR[,...]] [\fB-p\fR \fIp
.fi

.LP
.nf
\fBipadm\fR set-prop [\fB-t\fR] \fB-p\fR \fIprop\fR[+|-]=<\fIvalue\fR[,...]> \fI
.fi

.LP
.nf
\fBipadm\fR reset-prop [\fB-t\fR] \fB-p\fR \fIprop\fR \fIprotocol\fR
.fi
```

```
.LP
.nf
\fBipadm\fR show-prop [[\fB-c\fR] \fB-o\fR \fIfield\fR[,...]] [\fB-p\fR \fIprop\
.fi

.SH DESCRIPTION
.sp
.LP

The \fBipadm\fR command is a stable replacement for the \fBifconfig\fR(1M) and
\fBndd\fR(1M) commands. It is used to create IP interfaces and to confgure IP
addresses on those interfaces. It is also used to get, set or reset properties
on interfaces, addresses and protocols.
.LP
For subcommands that take an \fIaddrobj\fR, the \fIaddrobj\fR specifies a
unique address on the system. It is made up of two parts, delimited by a '/'.
The first part is the name of the interface and the second part is a string up
to 32 characters long.  For example, "lo0/v4" is a loopback interface
addrobj name.
.LP
For subcommands that take a \fIprotocol\fR, this can be one of
the following values: ip, ipv4, ipv6, icmp, tcp, sctp or udp.

.SH SUBCOMMANDS
.sp
.LP
The following subcommands are supported:
.sp
.ne 2
.na
\fB\fBcreate-if\fR [\fB-t\fR] \fIinterface\fR\fR
.ad
.sp .6
.RS 4n
The \fBcreate-if\fR subcommand is used to create an IP interface that will
handle both IPv4 and IPv6 packets. The interface will be enabled as part of
the creation process.  The IPv4 interface will have the address 0.0.0.0.
The IPv6 interface will have the adress ::.
.sp
The \fB-t\fR option (also \fB--temporary\fR) means
that the creation is temporary and will not be persistent across reboots.
.sp

.RE

.sp
.ne 2
.na
\fB\fBdisable-if\fR [\fB-t\fR] \fIinterface\fR\fR
.ad
.sp .6
.RS 4n
The \fBdisable-if\fR subcommand is used to disable an IP interface.
.sp
The \fB-t\fR option (also \fB--temporary\fR) means
that the disable is temporary and will not be persistent across reboots.
.sp

.RE

.sp
.ne 2
.na
\fB\fBenable-if\fR [\fB-t\fR] \fIinterface\fR\fR
.ad
```



```
.sp .6
.RS 4n
The \fBenable-if\fR subcommand is used to enable an IP interface.
.sp
The \fB-t\fR option (also \fB--temporary\fR) means
that the enable is temporary and will not be persistent across reboots.
.sp

.RE

.sp
.ne 2
.na
\fB\fbdelete-if\fR \fIinterface\fR\fR
.ad
.sp .6
.RS 4n
The \fBdelete-if\fR subcommand is used to permanently delete an IP interface.
.sp

.RE

.sp
.ne 2
.na
\fB\fBshow-if\fR [[\fB-p\fR] \fB-o\fR \fIfield\fR[,...]] [\fIinterface\fR]\fR
.ad
.sp .6
.RS 4n
The \fBshow-if\fR subcommand is used to show the current IP interface
configuration.
.sp
The \fB-p\fR option (also \fB--parsable\fR) prints
the output in a parsable format.
.sp
The \fB-o\fR option (also \fB--output\fR) is used
to select which fields will be shown.  The field value can be one of the
following names:
.sp
.ne 2
.na
.RS 4n
\fBALL\fR
.ad
.RS 4n
Display all fields
.RE

.sp
.ne 2
.na
\fBIFNAME\fR
.ad
.RS 4n
The name of the interface
.RE

.sp
.ne 2
.na
\fBSTATE\fR
.ad
.RS 4n
The state can be one of the following values:
.sp
.ne 2
```

```
.na
.RS 4n
ok - resources for the interface have been allocated
.sp
offline - the interface is offline
.sp
failed - the interface’s datalink is down
.sp
down - the interface is down
.sp
disabled - the interface is disabled
.RE
.RE

.sp
.ne 2
.na
\fBCURRENT\fR
.ad
.RS 4n
A set of single character flags indicating the following:
.sp
.ne 2
.na
.RS 4n
b - broadcast (mutually exclusive with ’p’)
.br
m - multicast
.br
p - point-to-point (mutually exclusive with ’b’)
.br
v - virtual interface
.br
I - IPMP
.br
s - IPMP standby
.br
i - IPMP inactive
.br
V - VRRP
.br
a - VRRP accept mode
.br
4 - IPv4
.br
6 - IPv6
.RE
.RE

.sp
.ne 2
.na
\fBPERSISTENT\fR
.ad
.RS 4n
A set of single character flags showing what configuration will be used the
next time the interface is enabled:
.sp
.ne 2
.na
.RS 4n
s - IPMP standby
.br
4 - IPv4
.br
6 - IPv6
```



```
.RE
.RE
.RE

.RE

.sp
.ne 2
.na
\fB\fBset-ifprop\fR [\fB-t\fR] \fB-p\fR \fIprop\fR=<\fIvalue\fR[,...]> \fB-m\fR
.ad
.sp .6
.RS 4n
The \fBset-ifprop\fR subcommand is used to set a property’s value(s) on the IP
interface.
.sp
The \fB-t\fR option (also \fB--temporary\fR) means
that the setting is temporary and will not be persistent across reboots.
.sp
The \fB-p\fR option (also \fB--prop\fR) specifies the property name and
value(s).  The property name can be one of the following:
.sp
.ne 2
.na

.RS 4n

\fBarp\fR
.ad
.RS 4n
Enables ("on") or disables ("off") ARP.
.RE

.sp
.ne 2
.na
\fBexchange_routes\fR
.ad
.RS 4n
Enables ("on") or disables ("off") the exchange of routing data.
.RE

.sp
.ne 2
.na
\fBforwarding\fR
.ad
.RS 4n
Enables ("on") or disables ("off") IP forwarding.
.RE

.sp
.ne 2
.na
\fBmetric\fR
.ad
.RS 4n
Set the routing metric to the numeric value. The value is treated as extra
hops to the destination.
.RE

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

```
.RS 4n
Set the maximum transmission unit to the numeric value.
.RE

.sp
.ne 2
.na
\fBnud\fR
.ad
.RS 4n
Enables ("on") or disables ("off") neighbor unreachability detection.
.RE

.sp
.ne 2
.na
\fBusesrc\fR
.ad
.RS 4n
Indicates which interface to use for source address selection. A value
"none" may also be used.
.RE
.RE

.sp
The \fB-m\fR option (also \fB--module\fR) specifies which protocol
the setting applies to.
.sp

.RE
.RE

.sp
.ne 2
.na
\fB\fBreset-ifprop\fR [\fB-t\fR] \fB-p\fR \fIprop\fR \fB-m\fR \fIprotocol\fR \fI
.ad
.sp .6
.RS 4n
The \fBreset-ifprop\fR subcommand is used to reset an IP interface's property
value to the default.
.sp
The \fB-t\fR option (also \fB--temporary\fR) means
that the disable is temporary and will not be persistent across reboots.
.sp
The \fB-p\fR option (also \fB--prop\fR) specifies the property name.
See the \fBset-ifprop\fR subcommand for the list of property names.
.sp
The \fB-m\fR option (also \fB--module\fR) specifies which protocol
the setting applies to.
.sp

.RE

.sp
.ne 2
.na
\fB\fBshow-ifprop\fR [[\fB-c\fR]\fB-o\fR \fIfield\fR[,...]] [\fB-p\fR \fIprop\fR
  [\fIinterface\fR]\fR
.ad
.sp .6
.RS 4n
The \fBshow-ifprop\fR subcommand is used to display the property values
for one or all of the IP interfaces.
.sp
The \fB-c\fR option (also \fB--parsable\fR) prints
```



```
the output in a parsable format.
.sp
The \fB-o\fR option (also \fB--output\fR) is used
to select which fields will be shown.  The field value can be one of the
following names:
.sp
.ne 2
.na
.RS 4n
\fBALL\fR
.ad
.RS 4n
Display all fields
.RE

.sp
.ne 2
.na
\fBIFNAME\fR
.ad
.RS 4n
The name of the interface
.RE

.sp
.ne 2
.na
\fBPROPERTY\fR
.ad
.RS 4n
The name of the property
.RE

.sp
.ne 2
.na
\fBPROTO\fR
.ad
.RS 4n
The name of the protocol
.RE

.sp
.ne 2
.na
\fBPERM\fR
.ad
.RS 4n
If the property is readable ("r") and/or writable ("w").
.RE

.sp
.ne 2
.na
\fBCURRENT\fR
.ad
.RS 4n
The value of the property
.RE

.sp
.ne 2
.na
\fBPERSISTENT\fR
.ad
.RS 4n
```

```
523 The persistent value of the property
524 .RE

526 .sp
527 .ne 2
528 .na
529 \fBDEFAULT\fR
530 .ad
531 .RS 4n
532 The default value of the property
533 .RE

535 .sp
536 .ne 2
537 .na
538 \fBPOSSIBLE\fR
539 .ad
540 .RS 4n
541 The possible values for the property
542 .RE
543 .RE

545 .sp
546 The \fB-p\fR option (also \fB--prop\fR) is used
547 to specify which properties to display.  See the \fBset-ifprop\fR
548 subcommand for the list of property names.
549 .sp
550 The \fB-m\fR option (also \fB--module\fR) specifies which protocol
551 to display.
552 .sp

554 .RE

556 .sp
557 .ne 2
558 .na
559 \fB\fBcreate-addr\fR [\fB-t\fR] \fB-T\fR static [\fB-d\fR] \\
560    \fB-a\fR {local|remote}=\fIaddr\fR[/\fIprefixlen\fR],... \fIaddrobj\fR\fR
561 .br
562 \fB\fBcreate-addr\fR [\fB-t\fR] \fB-T\fR dhcp [\fB-w\fR \fIseconds\fR | forever
563 .br
564 \fB\fBcreate-addr\fR [\fB-t\fR] \fB-T\fR addrconf [\fB-i\fR \fIinterface-id\fR]
565    [\fB-p\fR {stateful|stateless}={yes|no},..] \fIaddrobj\fR\fR
566 .ad
567 .sp .6
568 .RS 4n
569 The \fBcreate-addr\fR subcommand is used to set an address on an IP interface.
570 The address will be enabled but can disabled using the \fBdisable-addr\fR
571 subcommand. This subcommand has three different forms, depending on the
572 value of the \fB-T\fR option.
573 .sp
574 The \fB-t\fR option (also \fB--temporary\fR) means
575 that the address is temporary and will not be persistent across reboots.
576 .sp
577 The \fB-T\fR static option creates a static addrobj.  This takes the following
578 options:
579 .RS 4n

581 The \fB-d\fR option (also \fB--down\fR) means the address is down.
582 .sp
583 The \fB-a\fR option (also \fB--address\fR) specifies the address.
584 The "local" or "remote" prefix can be used for a point-to-point interface.
585 In this case, both addresses must be given.
586 Otherwise, the equal sign ("=") should be omitted and the address should be
587 provided by itself and with no second address.
588 .sp
```



```
590 .RE

592 The \fB-T\fR dhcp option causes the address to be obtained via DHCP.
593 This takes the following options:
594 .RS 4n

596 The \fB-w\fR option (also \fB--wait\fR) gives the time, in seconds,
597 that the command should wait to obtain an address.
598 .sp

600 .RE

602 The \fB-T\fR addrconf option creates an auto-configured address.
603 This takes the following options:
604 .RS 4n

606 The \fB-i\fR option (also \fB--interface-id\fR) gives the interface ID to
607 be used.
608 .sp
609 The \fB-p\fR option (also \fB--prop\fR) indicates which method of
610 auto-configuration should be used.
611 .sp

613 .RE
614 .RE

616 .sp
617 .ne 2
618 .na
619 \fB\fBdown-addr\fR [\fB-t\fR] \fIaddrobj\fR\fR
620 .ad
621 .sp .6
622 .RS 4n
623 The \fBdown-addr\fR subcommand is used to down the address. This will
624 stop packets from being sent or received.
625 .sp
626 The \fB-t\fR option (also \fB--temporary\fR) means
627 that the down is temporary and will not be persistent across reboots.
628 .sp

630 .RE

632 .sp
633 .ne 2
634 .na
635 \fB\fBup-addr\fR [\fB-t\fR] \fIaddrobj\fR\fR
636 .ad
637 .sp .6
638 .RS 4n
639 The \fBup-addr\fR subcommand is used to up the address. This will
640 enable packets to be sent and received.
641 .sp
642 The \fB-t\fR option (also \fB--temporary\fR) means
643 that the up is temporary and will not be persistent across reboots.
644 .sp

646 .RE

648 .sp
649 .ne 2
650 .na
651 \fB\fBdisable-addr\fR [\fB-t\fR] \fIaddrobj\fR\fR
652 .ad
653 .sp .6
654 .RS 4n
```

```
655 The \fBdisable-addr\fR subcommand is used to disable the address.
656 .sp
657 The \fB-t\fR option (also \fB--temporary\fR) means
658 that the disable is temporary and will not be persistent across reboots.
659 .sp

661 .RE

663 .sp
664 .ne 2
665 .na
666 \fB\fBenable-addr\fR [\fB-t\fR] \fIaddrobj\fR\fR
667 .ad
668 .sp .6
669 .RS 4n
670 The \fBenable-addr\fR subcommand is used to enable the address.
671 .sp
672 The \fB-t\fR option (also \fB--temporary\fR) means
673 that the enable is temporary and will not be persistent across reboots.
674 .sp

676 .RE

678 .sp
679 .ne 2
680 .na
681 \fB\fBrefresh-addr\fR [\fB-i\fR] \fIaddrobj\fR\fR
682 .ad
683 .sp .6
684 .RS 4n
685 The \fBrefresh-addr\fR subcommand is used to extend the lease for DHCP
686 addresses. It also restarts duplicate address detection for Static addresses.
687 .sp
688 The \fB-i\fR option (also \fB--inform\fR) means
689 that the network configuration will be obtained from DHCP without taking
690 a lease on the address.
691 .sp

693 .RE

695 .sp
696 .ne 2
697 .na
698 \fB\fBdelete-addr\fR [\fB-r\fR] \fIaddrobj\fR\fR
699 .ad
700 .sp .6
701 .RS 4n
702 The \fBdelete-addr\fR subcommand deletes the given address.
703 .sp
704 The \fB-r\fR option (also \fB--release\fR) is used for DHCP-assigned
705 addresses to indicate that the address should be released.
706 .sp

708 .RE

710 .sp
711 .ne 2
712 .na
713 \fB\fBshow-addr\fR [[\fB-p\fR] \fB-o\fR \fIfield\fR[,...]] [\fIaddrobj\fR]\fR
714 .ad
715 .sp .6
716 .RS 4n
717 The \fBshow-addr\fR subcommand is used to show the current address properties.
718 .sp
719 The \fB-p\fR option (also \fB--parsable\fR) prints
720 the output in a parsable format.
```



```
721 .sp
722 The \fB-o\fR option (also \fB--output\fR) is used
723 to select which fields will be shown.  The field value can be one of the
724 following names:
725 .sp
726 .ne 2
727 .na
728 .RS 4n
729 \fBALL\fR
730 .ad
731 .RS 4n
732 Display all fields
733 .RE

735 .sp
736 .ne 2
737 .na
738 \fBADDROBJ\fR
739 .ad
740 .RS 4n
741 The name of the address
742 .RE

744 .sp
745 .ne 2
746 .na
747 \fBTYPE\fR
748 .ad
749 .RS 4n
750 The type of the address.  It can be "static", "dhcp" or "addrconf".
751 .RE

753 .sp
754 .ne 2
755 .na
756 \fBSTATE\fR
757 .ad
758 .RS 4n
759 The state of the address.  It can be one of the following values:
760 .sp
761 .ne 2
762 .na
763 .RS 4n
764 disabled s see the \fBdisable-addr\fR subcommand
765 .sp
766 down - see the \fBdown-addr\fR subcommand
767 .sp
768 duplicate - the address is a duplicate
769 .sp
770 inaccessible - the interface for this address has failed
771 .sp
772 ok - the address is up
773 .sp
774 tentative - duplicate address detection in progress
775 .RE
776 .RE

778 .sp
779 .ne 2
780 .na
781 \fBCURRENT\fR
782 .ad
783 .RS 4n
784 A set of single character flags indicating the following:
785 .sp
786 .ne 2
```

```
.na
.RS 4n
U - up
.br
u - unnumbered (matches another local address)
.br
p - private, not advertised to routing
.br
t - temporary IPv6 address
.br
d - deprecated (not used for outgoing packets)
.RE
.RE

.sp
.ne 2
.na
\fBPERSISTENT\fR
.ad
.RS 4n
A set of single character flags showing the configuration which will be used
when the address is enabled.
.sp
.ne 2
.na
.RS 4n
U - up
.br
p - private, not advertised to routing
.br
d - deprecated (not used for outgoing packets)
.RE
.RE

.sp
.ne 2
.na
\fBADDR\fR
.ad
.RS 4n
The address
.RE
.RE

.RE

.sp
.ne 2
.na
\fB\fBset-addrprop\fR [\fB-t\fR] \fB-p\fR \fIprop\fR=<\fIvalue\fR[,...]> \fIaddr
.ad
.sp .6
.RS 4n
The \fBset-addrprop\fR subcommand is used to set a property’s value(s) on the
addrobj.
.sp
The \fB-t\fR option (also \fB--temporary\fR) means
that the setting is temporary and will not be persistent across reboots.
.sp
The \fB-p\fR option (also \fB--prop\fR) specifies the property name and
value(s).  The property name can be one of the following:
.sp
.ne 2
.na

.RS 4n
```



```
\fBbroadcast\fR
.ad
.RS 4n
The broadcast address (read-only)
.RE

.sp
.ne 2
.na
\fBdeprecated\fR
.ad
.RS 4n
The address should not be used to send packets but can still receive packets.
Can be "on" or "off".
.RE

.sp
.ne 2
.na
\fBprefixlen\fR
.ad
.RS 4n
The number of bits in the IPv4 netmask or IPv6 prefix.
.RE

.sp
.ne 2
.na
\fBprivate\fR
.ad
.RS 4n
The address is not advertised to routing.
Can be "on" or "off".
.RE

.sp
.ne 2
.na
\fBtransmit\fR
.ad
.RS 4n
Packets can be transmitted.
Can be "on" or "off".
.RE

.sp
.ne 2
.na
\fBzone\fR
.ad
.RS 4n
The zone the addrobj is in.
.RE

.RE
.RE

.sp
.ne 2
.na
\fB\fBreset-addrprop\fR [\fB-t\fR] \fB-p\fR \fIprop\fR \fIaddrobj\fR\fR
.ad
.sp .6
.RS 4n
The \fBreset-addrprop\fR subcommand is used to reset an addrobj’s property
```

```
 919 value to the default.
 920 .sp
 921 The \fB-t\fR option (also \fB--temporary\fR) means
 922 that the disable is temporary and will not be persistent across reboots.
 923 .sp
 924 The \fB-p\fR option (also \fB--prop\fR) specifies the property name.
 925 See the \fBset-addrprop\fR subcommand for the list of property names.
 926 .sp

 928 .RE

 930 .sp
 931 .ne 2
 932 .na
 933 \fB\fBshow-addrprop\fR [[\fB-c\fR]\fB-o\fR \fIfield\fR[,...]] [\fB-p\fR \fIprop\
 934 .ad
 935 .sp .6
 936 .RS 4n
 937 The \fBshow-addrprop\fR subcommand is used to display the property values
 938 for one or all of the addrobjs.
 939 .sp
 940 The \fB-c\fR option (also \fB--parsable\fR) prints
 941 the output in a parsable format.
 942 .sp
 943 The \fB-o\fR option (also \fB--output\fR) is used
 944 to select which fields will be shown.  The field value can be one of the
 945 following names:
 946 .sp
 947 .ne 2
 948 .na
 949 .RS 4n
 950 \fBALL\fR
 951 .ad
 952 .RS 4n
 953 Display all fields
 954 .RE

 956 .sp
 957 .ne 2
 958 .na
 959 \fBADDROBJ\fR
 960 .ad
 961 .RS 4n
 962 The name of the addrobj
 963 .RE

 965 .sp
 966 .ne 2
 967 .na
 968 \fBPROPERTY\fR
 969 .ad
 970 .RS 4n
 971 The name of the property
 972 .RE

 974 .sp
 975 .ne 2
 976 .na
 977 \fBPERM\fR
 978 .ad
 979 .RS 4n
 980 If the property is readable ("r") and/or writable ("w").
 981 .RE

 983 .sp
 984 .ne 2
```



```
 985 .na
 986 \fBCURRENT\fR
 987 .ad
 988 .RS 4n
 989 The value of the property
 990 .RE

 992 .sp
 993 .ne 2
 994 .na
 995 \fBPERSISTENT\fR
 996 .ad
 997 .RS 4n
 998 The persistent value of the property
 999 .RE

1001 .sp
1002 .ne 2
1003 .na
1004 \fBDEFAULT\fR
1005 .ad
1006 .RS 4n
1007 The default value of the property
1008 .RE

1010 .sp
1011 .ne 2
1012 .na
1013 \fBPOSSIBLE\fR
1014 .ad
1015 .RS 4n
1016 The possible values for the property
1017 .RE
1018 .RE

1020 .sp
1021 The \fB-p\fR option (also \fB--prop\fR) is used
1022 to specify which properties to display.  See the \fBset-addrprop\fR
1023 subcommand for the list of property names.
1024 .sp

1026 .RE

1028 .sp
1029 .ne 2
1030 .na
1031 \fB\fBset-prop\fR [\fB-t\fR] \fB-p\fR \fIprop\fR[+|-]=<\fIvalue\fR[,...]> \fIpro
1032 .ad
1033 .sp .6
1034 .RS 4n
1035 The \fBset-prop\fR subcommand is used to set a property’s value(s) on the
1036 protocol.
1037 .sp
1038 The \fB-t\fR option (also \fB--temporary\fR) means
1039 that the setting is temporary and will not be persistent across reboots.
1040 .sp
1041 The \fB-p\fR option (also \fB--prop\fR) specifies the property name and
1042 value(s). The optional [+|-] syntax can be used to add/remove values from the
1043 current list of values on the property.
1044 The property name can be one of the following:
1045 .sp
1046 .ne 2
1047 .na

1049 .RS 4n
```

```
\fBecn\fR
.ad
.RS 4n
Explicit congestion control (TCP-only)
Can be "never", "passive" or "active".
.RE

\fBextra_priv_ports\fR
.ad
.RS 4n
Additional privileged ports (SCTP, TCP or UDP)
.RE

\fBforwarding\fR
.ad
.RS 4n
Packet forwarding is enabled.
Can be "on" or "off".
.RE

\fBhoplimit\fR
.ad
.RS 4n
The IPv6 hoplimit.
.RE

\fBlargest_anon_port\fR
.ad
.RS 4n
Largest ephemeral port (SCTP, TCP or UDP)
.RE

\fBmax_buf\fR
.ad
.RS 4n
Maximum receive or send buffer size (ICMP, SCTP, TCP, or UDP).  This also
sets the upper limit for the \fBrecv_buf\fB and \fBsend_buf\fB properties.
.RE

\fBrecv_buf\fR
.ad
.RS 4n
Default receive buffer size (ICMP, SCTP, TCP, or UDP).  The maximum value for
this property is controlled by the \fBmax_buf\fR property.
.RE

\fBsack\fR
.ad
.RS 4n
Selective acknowledgement (TCP).
Can be "active", "passive" or "never".
.RE

\fBsend_buf\fR
.ad
.RS 4n
Default send buffer size (ICMP, SCTP, TCP, or UDP).  The maximum value for
this property is controlled by the \fBmax_buf\fR property.
.RE

\fBsmallest_anon_port\fR
.ad
.RS 4n
Smallest ephemeral port (SCTP, TCP or UDP)
.RE
```



```
\fBsmallest_nonpriv_port\fR
.ad
.RS 4n
Smallest non-privileged port (SCTP, TCP or UDP)
.RE

\fBttl\fR
.ad
.RS 4n
The IPv4 time-to-live.
.RE

.RE
.RE

.sp
.ne 2
.na
\fB\fBreset-prop\fR [\fB-t\fR] \fB-p\fR \fIprop\fR \fIprotocol\fR\fR
.ad
.sp .6
.RS 4n
The \fBreset-prop\fR subcommand is used to reset a protocol's property
value to the default.
.sp
The \fB-t\fR option (also \fB--temporary\fR) means
that the disable is temporary and will not be persistent across reboots.
.sp
The \fB-p\fR option (also \fB--prop\fR) specifies the property name.
See the \fBset-prop\fR subcommand for the list of property names.
.sp

.RE

.sp
.ne 2
.na
\fB\fBshow-prop\fR [[\fB-c\fR]\fB-o\fR \fIfield\fR[,...]] [\fB-p\fR \fIprop\fR,.
.ad
.sp .6
.RS 4n
The \fBshow-prop\fR subcommand is used to display the property values
for one or all of the protocols.
.sp
The \fB-c\fR option (also \fB--parsable\fR) prints
the output in a parsable format.
.sp
The \fB-o\fR option (also \fB--output\fR) is used
to select which fields will be shown.  The field value can be one of the
following names:
.sp
.ne 2
.na
.RS 4n
\fBALL\fR
.ad
.RS 4n
Display all fields
.RE

.sp
.ne 2
.na
\fBPROTO\fR
.ad
.RS 4n
```

```
1183 The name of the protocol
1184 .RE

1186 .sp
1187 .ne 2
1188 .na
1189 \fBPROPERTY\fR
1190 .ad
1191 .RS 4n
1192 The name of the property
1193 .RE

1195 .sp
1196 .ne 2
1197 .na
1198 \fBPERM\fR
1199 .ad
1200 .RS 4n
1201 If the property is readable ("r") and/or writable ("w").
1202 .RE

1204 .sp
1205 .ne 2
1206 .na
1207 \fBCURRENT\fR
1208 .ad
1209 .RS 4n
1210 The value of the property
1211 .RE

1213 .sp
1214 .ne 2
1215 .na
1216 \fBPERSISTENT\fR
1217 .ad
1218 .RS 4n
1219 The persistent value of the property
1220 .RE

1222 .sp
1223 .ne 2
1224 .na
1225 \fBDEFAULT\fR
1226 .ad
1227 .RS 4n
1228 The default value of the property
1229 .RE

1231 .sp
1232 .ne 2
1233 .na
1234 \fBPOSSIBLE\fR
1235 .ad
1236 .RS 4n
1237 The possible values for the property
1238 .RE
1239 .RE

1241 .sp
1242 The \fB-p\fR option (also \fB--prop\fR) is used
1243 to specify which properties to display.  See the \fBset-prop\fR
1244 subcommand for the list of property names.
1245 .sp

1247 .RE
```



```
1249 .SH SEE ALSO
1250 .sp
1251 .LP
1252 \fBifconfig\fR(1M), \fBdladm\fR(1M), \fBndd\fR(1M), \fBzonecfg\fR(1M),
1253 \fBarp\fR(1M), \fBcfgadm\fR(1M), \fBif_mpadm\fR(1M), \fBnsswitch.conf\fR(4),
1254 and \fBdhcp\fR(5).
```

```
**********************************************************
   18373 Thu Jul 17 00:50:40 2014
new/usr/src/man/man1m/lofiadm.1m
manpage lint.
**********************************************************
'\" te
.\" Copyright 2013 Nexenta Systems, Inc. All rights reserved.
.\" Copyright (c) 2008, Sun Microsystems, Inc. All Rights Reserved
.\" The contents of this file are subject to the terms of the Common Development
.\"  See the License for the specific language governing permissions and limitat
.\" the fields enclosed by brackets "[]" replaced with your own identifying info
.TH LOFIADM 1M "Aug 28, 2013"
.SH NAME
lofiadm \- administer files available as block devices through lofi
.SH SYNOPSIS
.LP
.nf
\fBlofiadm\fR [\fB-r\fR] \fB-a\fR \fIfile\fR [\fIdevice\fR]
.fi

.LP
.nf
\fBlofiadm\fR [\fB-r\fR] \fB-c\fR \fIcrypto_algorithm\fR \fB-a\fR \fIfile\fR [\f
.fi

.LP
.nf
\fBlofiadm\fR [\fB-r\fR] \fB-c\fR \fIcrypto_algorithm\fR \fB-k\fR \fIraw_key_fil
.fi

.LP
.nf
\fBlofiadm\fR [\fB-r\fR] \fB-c\fR \fIcrypto_algorithm\fR \fB-T\fR \fItoken_key\f
.fi

.LP
.nf
\fBlofiadm\fR [\fB-r\fR] \fB-c\fR \fIcrypto_algorithm\fR \fB-T\fR \fItoken_key\f
     \fB-k\fR \fIwrapped_key_file\fR \fB-a\fR \fIfile\fR [\fIdevice\fR]
.fi

.LP
.nf
\fBlofiadm\fR [\fB-r\fR] \fB-c\fR \fIcrypto_algorithm\fR \fB-e\fR \fB-a\fR \fIfi
.fi

.LP
.nf
\fBlofiadm\fR \fB-C\fR \fIalgorithm\fR [\fB-s\fR \fIsegment_size\fR] \fIfile\fR
.fi

.LP
.nf
\fBlofiadm\fR \fB-d\fR \fIfile\fR | \fIdevice\fR
.fi

.LP
.nf
\fBlofiadm\fR \fB-U\fR \fIfile\fR
.fi

.LP
.nf
\fBlofiadm\fR [ \fIfile\fR | \fIdevice\fR]
.fi
```



```
.SH DESCRIPTION
.sp
.LP
\fBlofiadm\fR administers \fBlofi\fR, the loopback file driver. \fBlofi\fR
allows a file to be associated with a block device. That file can then be
accessed through the block device. This is useful when the file contains an
image of some filesystem (such as a floppy or \fBCD-ROM\fR image), because the
block device can then be used with the normal system utilities for mounting,
checking or repairing filesystems. See \fBfsck\fR(1M) and \fBmount\fR(1M).
.sp
.LP
Use \fBlofiadm\fR to add a file as a loopback device, remove such an
association, or print information about the current associations.
.sp
.LP
Encryption and compression options are mutually exclusive on the command line.
Further, an encrypted file cannot be compressed later, nor can a compressed
file be encrypted later.

In the global zone, \fBlofiadm\fR can be used on both the global
zone devices and all devices owned by other non-global zones on the system.
.sp
.LP
.SH OPTIONS
.sp
.LP
The following options are supported:
.sp
.ne 2
.na
\fB\fB-a\fR \fIfile\fR [\fIdevice\fR]\fR
.ad
.sp .6
.RS 4n
Add \fIfile\fR as a block device.
.sp
If \fIdevice\fR is not specified, an available device is picked.
.sp
If \fIdevice\fR is specified, \fBlofiadm\fR attempts to assign it to
\fIfile\fR. \fIdevice\fR must be available or \fBlofiadm\fR will fail. The
ability to specify a device is provided for use in scripts that wish to
reestablish a particular set of associations.
.RE

.sp
.ne 2
.na
\fB\fB-C\fR {\fIgzip\fR | \fIgzip-N\fR | \fIlzma\fR}\fR
.ad
.sp .6
.RS 4n
Compress the file with the specified compression algorithm.
.sp
The \fBgzip\fR compression algorithm uses the same compression as the
open-source \fBgzip\fR command. You can specify the \fBgzip\fR level by using
the value \fBgzip-\fR\fIN\fR where \fIN\fR is 6 (fast) or 9 (best compression
ratio). Currently, \fBgzip\fR, without a number, is equivalent to \fBgzip-6\fR
(which is also the default for the \fBgzip\fR command).
.sp
\fIlzma\fR stands for the LZMA (Lempel-Ziv-Markov) compression algorithm.
.sp
Note that you cannot write to a compressed file, nor can you mount a compressed
file read/write.
.RE

.sp
```

```
.ne 2
.na
\fB\fB-d\fR \fIfile\fR | \fIdevice\fR\fR
.ad
.sp .6
.RS 4n
Remove an association by \fIfile\fR or \fIdevice\fR name, if the associated
block device is not busy, and deallocates the block device.
.RE

.sp
.ne 2
.na
\fB\fB-r\fR
.ad
.sp .6
.RS 4n
If the \fB-r\fR option is specified before the \fB-a\fR option, the
\fIdevice\fR will be opened read-only.
.RE

.sp
.ne 2
.na
\fB\fB-s\fR \fIsegment_size\fR\fR
.ad
.sp .6
.RS 4n
The segment size to use to divide the file being compressed. \fIsegment_size\fR
can be an integer multiple of 512.
.RE

.sp
.ne 2
.na
\fB\fB-U\fR \fIfile\fR\fR
.ad
.sp .6
.RS 4n
Uncompress a compressed file.
.RE

.sp
.LP
The following options are used when the file is encrypted:
.sp
.ne 2
.na
\fB\fB-c\fR \fIcrypto_algorithm\fR\fR
.ad
.sp .6
.RS 4n
Select the encryption algorithm. The algorithm must be specified when
encryption is enabled because the algorithm is not stored in the disk image.
.sp
If none of \fB-e\fR, \fB-k\fR, or \fB-T\fR is specified, \fBlofiadm\fR prompts
for a passphrase, with a minimum length of eight characters, to be entered .
The passphrase is used to derive a symmetric encryption key using PKCS#5 PBKD2.
.RE

.sp
.ne 2
.na
\fB\fB-k\fR \fIraw_key_file\fR | \fIwrapped_key_file\fR\fR
.ad
.sp .6
```



```
.RS 4n
Path to raw or wrapped symmetric encryption key. If a PKCS#11 object is also
given with the \fB-T\fR option, then the key is wrapped by that object. If
\fB-T\fR is not specified, the key is used raw.
.RE

.sp
.ne 2
.na
\fB\fB-T\fR \fItoken_key\fR\fR
.ad
.sp .6
.RS 4n
The key in a PKCS#11 token to use for the encryption or for unwrapping the key
file.
.sp
If \fB-k\fR is also specified, \fB-T\fR identifies the unwrapping key, which
must be an RSA private key.
.RE

.sp
.ne 2
.na
\fB\fB-e\fR\fR
.ad
.sp .6
.RS 4n
Generate an ephemeral symmetric encryption key.
.RE

.SH OPERANDS
.sp
.LP
The following operands are supported:
.sp
.ne 2
.na
\fB\fIcrypto_algorithm\fR\fR
.ad
.sp .6
.RS 4n
One of: \fBaes-128-cbc\fR, \fBaes-192-cbc\fR, \fBaes-256-cbc\fR,
\fBdes3-cbc\fR, \fBblowfish-cbc\fR.
.RE

.sp
.ne 2
.na
\fB\fIdevice\fR\fR
.ad
.sp .6
.RS 4n
Display the file name associated with the block device \fIdevice\fR.
.sp
Without arguments, print a list of the current associations. Filenames must be
valid absolute pathnames.
.sp
When a file is added, it is opened for reading or writing by root. Any
restrictions apply (such as restricted root access over \fBNFS\fR). The file is
held open until the association is removed. It is not actually accessed until
the block device is used, so it will never be written to if the block device is
only opened read-only.

Note that the filename may appear as "?" if it is not possible to resolve the
path in the current context (for example, if it’s an NFS path in a non-global
zone).
```

```
.RE

.sp
.ne 2
.na
\fB\fIfile\fR\fR
.ad
.sp .6
.RS 4n
Display the block device associated with \fIfile\fR.
.RE

.sp
.ne 2
.na
\fB\fIraw_key_file\fR\fR
.ad
.sp .6
.RS 4n
Path to a file of the appropriate length, in bits, to use as a raw symmetric
encryption key.
.RE

.sp
.ne 2
.na
\fB\fItoken_key\fR\fR
.ad
.sp .6
.RS 4n
PKCS#11 token object in the format:
.sp
.in +2
.nf
\fItoken_name\fR:\fImanufacturer_id\fR:\fIserial_number\fR:\fIkey_label\fR
.fi
.in -2
.sp

All but the key label are optional and can be empty. For example, to specify a
token object with only its key label \fBMylofiKey\fR, use:
.sp
.in +2
.nf
-T :::MylofiKey
.fi
.in -2
.sp

.RE

.sp
.ne 2
.na
\fB\fIwrapped_key_file\fR\fR
.ad
.sp .6
.RS 4n
Path to file containing a symmetric encryption key wrapped by the RSA private
key specified by \fB-T\fR.
.RE

.SH EXAMPLES
.LP
\fBExample 1 \fRMounting an Existing CD-ROM Image
.sp
```



```
.LP
You should ensure that Solaris understands the image before creating the
\fBCD\fR. \fBlofi\fR allows you to mount the image and see if it works.

.sp
.LP
This example mounts an existing \fBCD-ROM\fR image (\fBsparc.iso\fR), of the
\fBRed Hat 6.0 CD\fR which was downloaded from the Internet. It was created
with the \fBmkisofs\fR utility from the Internet.

.sp
.LP
Use \fBlofiadm\fR to attach a block device to it:

.sp
.in +2
.nf
# \fBlofiadm -a /home/mike_s/RH6.0/sparc.iso\fR
/dev/lofi/1
.fi
.in -2
.sp

.sp
.LP
\fBlofiadm\fR picks the device and prints the device name to the standard
output. You can run \fBlofiadm\fR again by issuing the following command:

.sp
.in +2
.nf
# \fBlofiadm\fR
Block Device     File                           Options
/dev/lofi/1      /home/mike_s/RH6.0/sparc.iso   -
.fi
.in -2
.sp

.sp
.LP
Or, you can give it one name and ask for the other, by issuing the following
command:

.sp
.in +2
.nf
# \fBlofiadm /dev/lofi/1\fR
/home/mike_s/RH6.0/sparc.iso
.fi
.in -2
.sp

.sp
.LP
Use the \fBmount\fR command to mount the image:

.sp
.in +2
.nf
# \fBmount -F hsfs -o ro /dev/lofi/1 /mnt\fR
.fi
.in -2
.sp

.sp
.LP
```

```
391 Check to ensure that Solaris understands the image:

393 .sp
394 .in +2
395 .nf
396 # \fBdf -k /mnt\fR
397 Filesystem            kbytes    used   avail capacity  Mounted on
398 /dev/lofi/1           512418  512418       0   100%    /mnt
399 # \fBls /mnt\fR
400 \&./            RedHat/       doc/          ls-lR         rr_moved/
401 \&../           TRANS.TBL     dosutils/     ls-lR.gz      sbin@
402 \&.buildlog     bin@          etc@          misc/         tmp/
403 COPYING       boot/         images/       mnt/          usr@
404 README        boot.cat*     kernels/      modules/
405 RPM-PGP-KEY   dev@          lib@          proc/
406 .fi
407 .in -2
408 .sp

410 .sp
411 .LP
412 Solaris can mount the CD-ROM image, and understand the filenames. The image was
413 created properly, and you can now create the \fBCD-ROM\fR with confidence.

415 .sp
416 .LP
417 As a final step, unmount and detach the images:

419 .sp
420 .in +2
421 .nf
422 # \fBumount /mnt\fR
423 # \fBlofiadm -d /dev/lofi/1\fR
424 # \fBlofiadm\fR
425 Block Device             File             Options
426 .fi
427 .in -2
428 .sp

430 .LP
431 \fBExample 2 \fRMounting a Floppy Image
432 .sp
433 .LP
434 This is similar to the first example.

436 .sp
437 .LP
438 Using \fBlofi\fR to help you mount files that contain floppy images is helpful
439 if a floppy disk contains a file that you need, but the machine which you are
440 on does not have a floppy drive. It is also helpful if you do not want to take
441 the time to use the \fBdd\fR command to copy the image to a floppy.

443 .sp
444 .LP
445 This is an example of getting to \fBMDB\fR floppy for Solaris on an x86
446 platform:

448 .sp
449 .in +2
450 .nf
451 # \fBlofiadm -a /export/s28/MDB_s28x_wos/latest/boot.3\fR
452 /dev/lofi/1
453 # \fBmount -F pcfs /dev/lofi/1 /mnt\fR
454 # \fBls /mnt\fR
455 \&./            COMMENT.BAT*  RC.D/         SOLARIS.MAP*
456 \&../           IDENT*        REPLACE.BAT*  X/
```



```
457 APPEND.BAT*   MAKEDIR.BAT*  SOLARIS/
458 # \fBumount /mnt\fR
459 # \fBlofiadm -d /export/s28/MDB_s28x_wos/latest/boot.3\fR
460 .fi
461 .in -2
462 .sp

464 .LP
465 \fBExample 3 \fRMaking a \fBUFS\fR Filesystem on a File
466 .sp
467 .LP
468 Making a \fBUFS\fR filesystem on a file can be useful, particularly if a test
469 suite requires a scratch filesystem. It can be painful (or annoying) to have to
470 repartition a disk just for the test suite, but you do not have to. You can
471 \fBnewfs\fR a file with \fBlofi\fR

473 .sp
474 .LP
475 Create the file:

477 .sp
478 .in +2
479 .nf
480 # \fBmkfile 35m /export/home/test\fR
481 .fi
482 .in -2
483 .sp

485 .sp
486 .LP
487 Attach it to a block device. You also get the character device that \fBnewfs\fR
488 requires, so \fBnewfs\fR that:

490 .sp
491 .in +2
492 .nf
493 # \fBlofiadm -a /export/home/test\fR
494 /dev/lofi/1
495 # \fBnewfs /dev/rlofi/1\fR
496 newfs: construct a new file system /dev/rlofi/1: (y/n)? \fBy\fR
497 /dev/rlofi/1:   71638 sectors in 119 cylinders of 1 tracks, 602 sectors
498         35.0MB in 8 cyl groups (16 c/g, 4.70MB/g, 2240 i/g)
499 super-block backups (for fsck -F ufs -o b=#) at:
500  32, 9664, 19296, 28928, 38560, 48192, 57824, 67456,
501 .fi
502 .in -2
503 .sp

505 .sp
506 .LP
507 Note that \fBufs\fR might not be able to use the entire file. Mount and use the
508 filesystem:

510 .sp
511 .in +2
512 .nf
513 # \fBmount /dev/lofi/1 /mnt\fR
514 # \fBdf -k /mnt\fR
515 Filesystem            kbytes    used   avail capacity  Mounted on
516 /dev/lofi/1            33455       9   30101     1%    /mnt
517 # \fBls /mnt\fR
518 \&./            ../           lost+found/
519 # \fBumount /mnt\fR
520 # \fBlofiadm -d /dev/lofi/1\fR
521 .fi
522 .in -2
```

```
523 .sp

525 .LP
526 \fBExample 4 \fRCreating a PC (FAT) File System on a Unix File
527 .sp
528 .LP
529 The following series of commands creates a \fBFAT\fR file system on a Unix
530 file. The file is associated with a block device created by \fBlofiadm\fR.

532 .sp
533 .in +2
534 .nf
535 # \fBmkfile 10M /export/test/testfs\fR
536 # \fBlofiadm -a /export/test testfs\fR
537 /dev/lofi/1
538 \fBNote use of\fR rlofi\fB, not\fR lofi\fB, in following command.\fR
539 # \fBmkfs -F pcfs -o nofdisk,size=20480 /dev/rlofi/1\fR
540 \fBConstruct a new FAT file system on /dev/rlofi/1: (y/n)?\fR y
541 # \fBmount -F pcfs /dev/lofi/1 /mnt\fR
542 # \fBcd /mnt\fR
543 # \fBdf -k .\fR
544 Filesystem            kbytes    used   avail capacity  Mounted on
545 /dev/lofi/1            10142       0   10142     0%    /mnt
546 .fi
547 .in -2
548 .sp

550 .LP
551 \fBExample 5 \fRCompressing an Existing CD-ROM Image
552 .sp
553 .LP
554 The following example illustrates compressing an existing CD-ROM image
555 (\fBsolaris.iso\fR), verifying that the image is compressed, and then
556 uncompressing it.

558 .sp
559 .in +2
560 .nf
561 # \fBlofiadm -C gzip /export/home/solaris.iso\fR
562 .fi
563 .in -2
564 .sp

566 .sp
567 .LP
568 Use \fBlofiadm\fR to attach a block device to it:

570 .sp
571 .in +2
572 .nf
573 # \fBlofiadm -a /export/home/solaris.iso\fR
574   /dev/lofi/1
575 .fi
576 .in -2
577 .sp

579 .sp
580 .LP
581 Check if the mapped image is compressed:

583 .sp
584 .in +2
585 .nf
586 # \fBlofiadm\fR
587 Block Device      File                            Options
588 /dev/lofi/1       /export/home/solaris.iso        Compressed(gzip)
```



```
589 /dev/lofi/2       /export/home/regular.iso        -
590 .fi
591 .in -2
592 .sp

594 .sp
595 .LP
596 Unmap the compressed image and uncompress it:

598 .sp
599 .in +2
600 .nf
601 # \fBlofiadm -d /dev/lofi/1\fR
602 # \fBlofiadm -U /export/home/solaris.iso\fR
603 .fi
604 .in -2
605 .sp

607 .LP
608 \fBExample 6 \fRCreating an Encrypted UFS File System on a File
609 .sp
610 .LP
611 This example is similar to the example of making a UFS filesystem on a file,
612 above.

614 .sp
615 .LP
616 Create the file:

618 .sp
619 .in +2
620 .nf
621 # \fBmkfile 35m /export/home/test\fR
622 .fi
623 .in -2
624 .sp

626 .sp
627 .LP
628 Attach the file to a block device and specify that the file image is encrypted.
629 As a result of this command, you obtain the character device, which is
630 subsequently used by \fBnewfs\fR:

632 .sp
633 .in +2
634 .nf
635 # \fBlofiadm -c aes-256-cbc -a /export/home/secrets\fR
636 Enter passphrase: \fBMy-M0th3r;l0v3s_m3+4lw4ys!\fR          (\fBnot echoed\fR)
637 Re-enter passphrase: \fBMy-M0th3r;l0v3s_m3+4lw4ys!\fR       (\fBnot echoed\fR)
638 /dev/lofi/1

640 # \fBnewfs /dev/rlofi/1\fR
641 newfs: construct a new file system /dev/rlofi/1: (y/n)? \fBy\fR
642 /dev/rlofi/1:   71638 sectors in 119 cylinders of 1 tracks, 602 sectors
643        35.0MB in 8 cyl groups (16 c/g, 4.70MB/g, 2240 i/g)
644 super-block backups (for fsck -F ufs -o b=#) at:
645 32, 9664, 19296, 28928, 38560, 48192, 57824, 67456,
646 .fi
647 .in -2
648 .sp

650 .sp
651 .LP
652 The mapped file system shows that encryption is enabled:

654 .sp
```

```
655 .in +2
656 .nf
657 # \fBlofiadm\fR
658 Block Device    File                    Options
659 /dev/lofi/1     /export/home/secrets    Encrypted
660 .fi
661 .in -2
662 .sp

664 .sp
665 .LP
666 Mount and use the filesystem:

668 .sp
669 .in +2
670 .nf
671 # \fBmount /dev/lofi/1 /mnt\fR
672 # \fBcp moms_secret_*_recipe /mnt\fR
673 # \fBls /mnt\fR
674 \&./           moms_secret_cookie_recipe    moms_secret_soup_recipe
675 \&../          moms_secret_fudge_recipe     moms_secret_stuffing_recipe
676 lost+found/  moms_secret_meatloaf_recipe  moms_secret_waffle_recipe
677 # \fBumount /mnt\fR
678 # \fBlofiadm -d /dev/lofi/1\fR
679 .fi
680 .in -2
681 .sp

683 .sp
684 .LP
685 Subsequent attempts to map the filesystem with the wrong key or the wrong
686 encryption algorithm will fail:

688 .sp
689 .in +2
690 .nf
691 # \fBlofiadm -c blowfish-cbc -a /export/home/secrets\fR
692 Enter passphrase: \fBmommy\fR                              (\fInot echoed\fR)
693 Re-enter passphrase: \fBmommy\fR                           (\fInot echoed\fR)
694 lofiadm: could not map file /root/lofi: Invalid argument
695 # \fBlofiadm\fR
696 Block Device    File                    Options
697 #
698 .fi
699 .in -2
700 .sp

702 .sp
703 .LP
704 Attempts to map the filesystem without encryption will succeed, however
705 attempts to mount and use the filesystem will fail:

707 .sp
708 .in +2
709 .nf
710 # \fBlofiadm -a /export/home/secrets\fR
711 /dev/lofi/1
712 # \fBlofiadm\fR
713 Block Device    File                    Options
714 /dev/lofi/1     /export/home/secrets    -
715 # \fBmount /dev/lofi/1 /mnt\fR
716 mount: /dev/lofi/1 is not this fstype
717 #
718 .fi
719 .in -2
720 .sp
```



```
722 .SH ENVIRONMENT VARIABLES
723 .sp
724 .LP
725 See \fBenviron\fR(5) for descriptions of the following environment variables
726 that affect the execution of \fBlofiadm\fR: \fBLC_CTYPE\fR, \fBLC_MESSAGES\fR
727 and \fBNLSPATH\fR.
728 .SH EXIT STATUS
729 .sp
730 .LP
731 The following exit values are returned:
732 .sp
733 .ne 2
734 .na
735 \fB\fB0\fR\fR
736 .ad
737 .sp .6
738 .RS 4n
739 Successful completion.
740 .RE

742 .sp
743 .ne 2
744 .na
745 \fB\fB>0\fR\fR
746 .ad
747 .sp .6
748 .RS 4n
749 An error occurred.
750 .RE

752 .SH SEE ALSO
753 .sp
754 .LP
755 \fBfsck\fR(1M), \fBmount\fR(1M), \fBmount_ufs\fR(1M), \fBnewfs\fR(1M),
756 \fBattributes\fR(5), \fBlofi\fR(7D), \fBlofs\fR(7FS)
757 .SH NOTES
758 .sp
759 .LP
760 Just as you would not directly access a disk device that has mounted file
761 systems, you should not access a file associated with a block device except
762 through the \fBlofi\fR file driver. It might also be appropriate to ensure that
763 the file has appropriate permissions to prevent such access.
764 .sp
765 .LP
766 The abilities of \fBlofiadm\fR, and who can use them, are controlled by the
767 permissions of \fB/dev/lofictl\fR. Read-access allows query operations, such as
768 listing all the associations. Write-access is required to do any state-changing
769 operations, like adding an association. As shipped, \fB/dev/lofictl\fR is owned
770 by \fBroot\fR, in group \fBsys\fR, and mode \fB0644\fR, so all users can do
771 query operations but only root can change anything. The administrator can give
772 users write-access, allowing them to add or delete associations, but that is
773 very likely a security hole and should probably only be given to a trusted
774 group.
775 .sp
776 .LP
777 When mounting a filesystem image, take care to use appropriate mount options.
778 In particular, the \fBnosuid\fR mount option might be appropriate for \fBUFS\fR
779 images whose origin is unknown. Also, some options might not be useful or
780 appropriate, like \fBlogging\fR or \fBforcedirectio\fR for \fBUFS\fR. For
781 compatibility purposes, a raw device is also exported along with the block
782 device. For example, \fBnewfs\fR(1M) requires one.
783 .sp
784 .LP
785 The output of \fBlofiadm\fR (without arguments) might change in future
786 releases.
```

```
**********************************************************
    2466 Thu Jul 17 00:50:40 2014
new/usr/src/man/man1m/root_archive.1m
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
   43446 Thu Jul 17 00:50:40 2014
new/usr/src/man/man1m/zonecfg.1m
manpage lint.
**********************************************************
   1 '\" te
   2 .\" Copyright (c) 2004, 2009 Sun Microsystems, Inc. All Rights Reserved.
   3 .\" Copyright 2013 Joyent, 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 limitati
   6 .\" fields enclosed by brackets "[]" replaced with your own identifying informat
   7 .TH ZONECFG 1M "Feb 28, 2014"
   8 .SH NAME
   9 zonecfg \- set up zone configuration
  10 .SH SYNOPSIS
  11 .LP
  12 .nf
  13 \fBzonecfg\fR \fB-z\fR \fIzonename\fR
  14 .fi

  16 .LP
  17 .nf
  18 \fBzonecfg\fR \fB-z\fR \fIzonename\fR \fIsubcommand\fR
  19 .fi

  21 .LP
  22 .nf
  23 \fBzonecfg\fR \fB-z\fR \fIzonename\fR \fB-f\fR \fIcommand_file\fR
  24 .fi

  26 .LP
  27 .nf
  28 \fBzonecfg\fR help
  29 .fi

  31 .SH DESCRIPTION
  32 .sp
  33 .LP
  34 The \fBzonecfg\fR utility creates and modifies the configuration of a zone.
  35 Zone configuration consists of a number of resources and properties.
  36 .sp
  37 .LP
  38 To simplify the user interface, \fBzonecfg\fR uses the concept of a scope. The
  39 default scope is global.
  40 .sp
  41 .LP
  42 The following synopsis of the \fBzonecfg\fR command is for interactive usage:
  43 .sp
  44 .in +2
  45 .nf
  46 zonecfg \fB-z\fR \fIzonename subcommand\fR
  47 .fi
  48 .in -2
  49 .sp

  51 .sp
  52 .LP
  53 Parameters changed through \fBzonecfg\fR do not affect a running zone. The zone
  54 must be rebooted for the changes to take effect.
  55 .sp
  56 .LP
  57 In addition to creating and modifying a zone, the \fBzonecfg\fR utility can
  58 also be used to persistently specify the resource management settings for the
  59 global zone.
  60 .sp
  61 .LP
```



```
  62 In the following text, "rctl" is used as an abbreviation for "resource
  63 control". See \fBresource_controls\fR(5).
  64 .sp
  65 .LP
  66 Every zone is configured with an associated brand. The brand determines the
  67 user-level environment used within the zone, as well as various behaviors for
  68 the zone when it is installed, boots, or is shutdown. Once a zone has been
  69 installed the brand cannot be changed. The default brand is determined by the
  70 installed distribution in the global zone. Some brands do not support all of
  71 the \fBzonecfg\fR properties and resources. See the brand-specific man page for
  72 more details on each brand. For an overview of brands, see the \fBbrands\fR(5)
  73 man page.
  74 .SS "Resources"
  75 .sp
  76 .LP
  77 The following resource types are supported:
  78 .sp
  79 .ne 2
  80 .na
  81 \fB\fBattr\fR\fR
  82 .ad
  83 .sp .6
  84 .RS 4n
  85 Generic attribute.
  86 .RE

  88 .sp
  89 .ne 2
  90 .na
  91 \fB\fBcapped-cpu\fR\fR
  92 .ad
  93 .sp .6
  94 .RS 4n
  95 Limits for CPU usage.
  96 .RE

  98 .sp
  99 .ne 2
 100 .na
 101 \fB\fBcapped-memory\fR\fR
 102 .ad
 103 .sp .6
 104 .RS 4n
 105 Limits for physical, swap, and locked memory.
 106 .RE

 108 .sp
 109 .ne 2
 110 .na
 111 \fB\fBdataset\fR\fR
 112 .ad
 113 .sp .6
 114 .RS 4n
 115 \fBZFS\fR dataset.
 116 .RE

 118 .sp
 119 .ne 2
 120 .na
 121 \fB\fBdedicated-cpu\fR\fR
 122 .ad
 123 .sp .6
 124 .RS 4n
 125 Subset of the system's processors dedicated to this zone while it is running.
 126 .RE
```

```
 128 .sp
 129 .ne 2
 130 .na
 131 \fB\fBdevice\fR\fR
 132 .ad
 133 .sp .6
 134 .RS 4n
 135 Device.
 136 .RE

 138 .sp
 139 .ne 2
 140 .na
 141 \fB\fBfs\fR\fR
 142 .ad
 143 .sp .6
 144 .RS 4n
 145 file-system
 146 .RE

 148 .sp
 149 .ne 2
 150 .na
 151 \fB\fBnet\fR\fR
 152 .ad
 153 .sp .6
 154 .RS 4n
 155 Network interface.
 156 .RE

 158 .sp
 159 .ne 2
 160 .na
 161 \fB\fBrctl\fR\fR
 162 .ad
 163 .sp .6
 164 .RS 4n
 165 Resource control.
 166 .RE

 168 .SS "Properties"
 169 .sp
 170 .LP
 171 Each resource type has one or more properties. There are also some global
 172 properties, that is, properties of the configuration as a whole, rather than of
 173 some particular resource.
 174 .sp
 175 .LP
 176 The following properties are supported:
 177 .sp
 178 .ne 2
 179 .na
 180 \fB(global)\fR
 181 .ad
 182 .sp .6
 183 .RS 4n
 184 \fBzonename\fR
 185 .RE

 187 .sp
 188 .ne 2
 189 .na
 190 \fB(global)\fR
 191 .ad
 192 .sp .6
 193 .RS 4n
```



```
 194 \fBzonepath\fR
 195 .RE

 197 .sp
 198 .ne 2
 199 .na
 200 \fB(global)\fR
 201 .ad
 202 .sp .6
 203 .RS 4n
 204 \fBautoboot\fR
 205 .RE

 207 .sp
 208 .ne 2
 209 .na
 210 \fB(global)\fR
 211 .ad
 212 .sp .6
 213 .RS 4n
 214 \fBbootargs\fR
 215 .RE

 217 .sp
 218 .ne 2
 219 .na
 220 \fB(global)\fR
 221 .ad
 222 .sp .6
 223 .RS 4n
 224 \fBpool\fR
 225 .RE

 227 .sp
 228 .ne 2
 229 .na
 230 \fB(global)\fR
 231 .ad
 232 .sp .6
 233 .RS 4n
 234 \fBlimitpriv\fR
 235 .RE

 237 .sp
 238 .ne 2
 239 .na
 240 \fB(global)\fR
 241 .ad
 242 .sp .6
 243 .RS 4n
 244 \fBbrand\fR
 245 .RE

 247 .sp
 248 .ne 2
 249 .na
 250 \fB(global)\fR
 251 .ad
 252 .sp .6
 253 .RS 4n
 254 \fBcpu-shares\fR
 255 .RE

 257 .sp
 258 .ne 2
 259 .na
```

```
260 \fB(global)\fR
261 .ad
262 .sp .6
263 .RS 4n
264 \fBhostid\fR
265 .RE

267 .sp
268 .ne 2
269 .na
270 \fB(global)\fR
271 .ad
272 .sp .6
273 .RS 4n
274 \fBmax-lwps\fR
275 .RE

277 .sp
278 .ne 2
279 .na
280 \fB(global)\fR
281 .ad
282 .sp .6
283 .RS 4n
284 \fBmax-msg-ids\fR
285 .RE

287 .sp
288 .ne 2
289 .na
290 \fB(global)\fR
291 .ad
292 .sp .6
293 .RS 4n
294 \fBmax-sem-ids\fR
295 .RE

297 .sp
298 .ne 2
299 .na
300 \fB(global)\fR
301 .ad
302 .sp .6
303 .RS 4n
304 \fBmax-shm-ids\fR
305 .RE

307 .sp
308 .ne 2
309 .na
310 \fB(global)\fR
311 .ad
312 .sp .6
313 .RS 4n
314 \fBmax-shm-memory\fR
315 .RE

317 .sp
318 .ne 2
319 .na
320 \fB(global)\fR
321 .ad
322 .sp .6
323 .RS 4n
324 \fBscheduling-class\fR
325 .RE
```



```
327 .sp
328 .ne 2
329 .na
330 .B (global)
331 .ad
332 .sp .6
333 .RS 4n
334 .B fs-allowed
335 .RE

337 .sp
338 .ne 2
339 .na
340 \fB\fBfs\fR\fR
341 .ad
342 .sp .6
343 .RS 4n
344 \fBdir\fR, \fBspecial\fR, \fBraw\fR, \fBtype\fR, \fBoptions\fR
345 .RE

347 .sp
348 .ne 2
349 .na
350 \fB\fBnet\fR\fR
351 .ad
352 .sp .6
353 .RS 4n
354 \fBaddress\fR, \fBphysical\fR, \fBdefrouter\fR
355 .RE

357 .sp
358 .ne 2
359 .na
360 \fB\fBdevice\fR\fR
361 .ad
362 .sp .6
363 .RS 4n
364 \fBmatch\fR
365 .RE

367 .sp
368 .ne 2
369 .na
370 \fB\fBrctl\fR\fR
371 .ad
372 .sp .6
373 .RS 4n
374 \fBname\fR, \fBvalue\fR
375 .RE

377 .sp
378 .ne 2
379 .na
380 \fB\fBattr\fR\fR
381 .ad
382 .sp .6
383 .RS 4n
384 \fBname\fR, \fBtype\fR, \fBvalue\fR
385 .RE

387 .sp
388 .ne 2
389 .na
390 \fB\fBdataset\fR\fR
391 .ad
```

```
392 .sp .6
393 .RS 4n
394 \fBname\fR
395 .RE

397 .sp
398 .ne 2
399 .na
400 \fB\fBdedicated-cpu\fR\fR
401 .ad
402 .sp .6
403 .RS 4n
404 \fBncpus\fR, \fBimportance\fR
405 .RE

407 .sp
408 .ne 2
409 .na
410 \fB\fBcapped-memory\fR\fR
411 .ad
412 .sp .6
413 .RS 4n
414 \fBphysical\fR, \fBswap\fR, \fBlocked\fR
415 .RE

417 .sp
418 .ne 2
419 .na
420 \fB\fBcapped-cpu\fR\fR
421 .ad
422 .sp .6
423 .RS 4n
424 \fBncpus\fR
425 .RE

427 .sp
428 .LP
429 As for the property values which are paired with these names, they are either
430 simple, complex, or lists. The type allowed is property-specific. Simple values
431 are strings, optionally enclosed within quotation marks. Complex values have
432 the syntax:
433 .sp
434 .in +2
435 .nf
436 (<\fIname\fR>=<\fIvalue\fR>,<\fIname\fR>=<\fIvalue\fR>,...)
437 .fi
438 .in -2
439 .sp

441 .sp
442 .LP
443 where each <\fIvalue\fR> is simple, and the <\fIname\fR> strings are unique
444 within a given property. Lists have the syntax:
445 .sp
446 .in +2
447 .nf
448 [<\fIvalue\fR>,...]
449 .fi
450 .in -2
451 .sp

453 .sp
454 .LP
455 where each <\fIvalue\fR> is either simple or complex. A list of a single value
456 (either simple or complex) is equivalent to specifying that value without the
457 list syntax. That is, "foo" is equivalent to "[foo]". A list can be empty
```



```
458 (denoted by "[]").
459 .sp
460 .LP
461 In interpreting property values, \fBzonecfg\fR accepts regular expressions as
462 specified in \fBfnmatch\fR(5). See \fBEXAMPLES\fR.
463 .sp
464 .LP
465 The property types are described as follows:
466 .sp
467 .ne 2
468 .na
469 \fBglobal: \fBzonename\fR\fR
470 .ad
471 .sp .6
472 .RS 4n
473 The name of the zone.
474 .RE

476 .sp
477 .ne 2
478 .na
479 \fBglobal: \fBzonepath\fR\fR
480 .ad
481 .sp .6
482 .RS 4n
483 Path to zone’s file system.
484 .RE

486 .sp
487 .ne 2
488 .na
489 \fBglobal: \fBautoboot\fR\fR
490 .ad
491 .sp .6
492 .RS 4n
493 Boolean indicating that a zone should be booted automatically at system boot.
494 Note that if the zones service is disabled, the zone will not autoboot,
495 regardless of the setting of this property. You enable the zones service with a
496 \fBsvcadm\fR command, such as:
497 .sp
498 .in +2
499 .nf
500 # \fBsvcadm enable svc:/system/zones:default\fR
501 .fi
502 .in -2
503 .sp

505 Replace \fBenable\fR with \fBdisable\fR to disable the zones service. See
506 \fBsvcadm\fR(1M).
507 .RE

509 .sp
510 .ne 2
511 .na
512 \fBglobal: \fBbootargs\fR\fR
513 .ad
514 .sp .6
515 .RS 4n
516 Arguments (options) to be passed to the zone bootup, unless options are
517 supplied to the "\fBzoneadm boot\fR" command, in which case those take
518 precedence. The valid arguments are described in \fBzoneadm\fR(1M).
519 .RE

521 .sp
522 .ne 2
523 .na
```

```
524 \fBglobal: \fBpool\fR\fR
525 .ad
526 .sp .6
527 .RS 4n
528 Name of the resource pool that this zone must be bound to when booted. This
529 property is incompatible with the \fBdedicated-cpu\fR resource.
530 .RE

532 .sp
533 .ne 2
534 .na
535 \fBglobal: \fBlimitpriv\fR\fR
536 .ad
537 .sp .6
538 .RS 4n
539 The maximum set of privileges any process in this zone can obtain. The property
540 should consist of a comma-separated privilege set specification as described in
541 \fBpriv_str_to_set\fR(3C). Privileges can be excluded from the resulting set by
542 preceding their names with a dash (-) or an exclamation point (!). The special
543 privilege string "zone" is not supported in this context. If the special string
544 "default" occurs as the first token in the property, it expands into a safe set
545 of privileges that preserve the resource and security isolation described in
546 \fBzones\fR(5). A missing or empty property is equivalent to this same set of
547 safe privileges.
548 .sp
549 The system administrator must take extreme care when configuring privileges for
550 a zone. Some privileges cannot be excluded through this mechanism as they are
551 required in order to boot a zone. In addition, there are certain privileges
552 which cannot be given to a zone as doing so would allow processes inside a zone
553 to unduly affect processes in other zones. \fBzoneadm\fR(1M) indicates when an
554 invalid privilege has been added or removed from a zone's privilege set when an
555 attempt is made to either "boot" or "ready" the zone.
556 .sp
557 See \fBprivileges\fR(5) for a description of privileges. The command "\fBppriv
558 -l\fR" (see \fBppriv\fR(1)) produces a list of all Solaris privileges. You can
559 specify privileges as they are displayed by \fBppriv\fR. In
560 \fBprivileges\fR(5), privileges are listed in the form
561 PRIV_\fIprivilege_name\fR. For example, the privilege \fIsys_time\fR, as you
562 would specify it in this property, is listed in \fBprivileges\fR(5) as
563 \fBPRIV_SYS_TIME\fR.
564 .RE

566 .sp
567 .ne 2
568 .na
569 \fBglobal: \fBbrand\fR\fR
570 .ad
571 .sp .6
572 .RS 4n
573 The zone's brand type.
574 .RE

576 .sp
577 .ne 2
578 .na
579 \fBglobal: \fBip-type\fR\fR
580 .ad
581 .sp .6
582 .RS 4n
583 A zone can either share the IP instance with the global zone, which is the
584 default, or have its own exclusive instance of IP.
585 .sp
586 This property takes the values \fBshared\fR and \fBexclusive\fR.
587 .RE

589 .sp
```



```
590 .ne 2
591 .na
592 \fBglobal: \fBhostid\fR\fR
593 .ad
594 .sp .6
595 .RS 4n
596 A zone can emulate a 32-bit host identifier to ease system consolidation. A
597 zone's \fBhostid\fR property is empty by default, meaning that the zone does
598 not emulate a host identifier. Zone host identifiers must be hexadecimal values
599 between 0 and FFFFFFFE. A \fB0x\fR or \fB0X\fR prefix is optional. Both
600 uppercase and lowercase hexadecimal digits are acceptable.
601 .RE

603 .sp
604 .ne 2
605 .na
606 \fB\fBfs\fR: dir, special, raw, type, options\fR
607 .ad
608 .sp .6
609 .RS 4n
610 Values needed to determine how, where, and so forth to mount file systems. See
611 \fBmount\fR(1M), \fBmount\fR(2), \fBfsck\fR(1M), and \fBvfstab\fR(4).
612 .RE

614 .sp
615 .ne 2
616 .na
617 \fB\fBnet\fR: address, physical, defrouter\fR
618 .ad
619 .sp .6
620 .RS 4n
621 The network address and physical interface name of the network interface. The
622 network address is one of:
623 .RS +4
624 .TP
625 .ie t \(bu
626 .el o
627 a valid IPv4 address, optionally followed by "\fB/\fR" and a prefix length;
628 .RE
629 .RS +4
630 .TP
631 .ie t \(bu
632 .el o
633 a valid IPv6 address, which must be followed by "\fB/\fR" and a prefix length;
634 .RE
635 .RS +4
636 .TP
637 .ie t \(bu
638 .el o
639 a host name which resolves to an IPv4 address.
640 .RE
641 Note that host names that resolve to IPv6 addresses are not supported.
642 .sp
643 The physical interface name is the network interface name.
644 .sp
645 The default router is specified similarly to the network address except that it
646 must not be followed by a \fB/\fR (slash) and a network prefix length.
647 .sp
648 A zone can be configured to be either exclusive-IP or shared-IP. For a
649 shared-IP zone, you must set both the physical and address properties; setting
650 the default router is optional. The interface specified in the physical
651 property must be plumbed in the global zone prior to booting the non-global
652 zone. However, if the interface is not used by the global zone, it should be
653 configured \fBdown\fR in the global zone, and the default router for the
654 interface should be specified here.
655 .sp
```

```
656 For an exclusive-IP zone, the physical property must be set and the address and
657 default router properties cannot be set.
658 .RE

660 .sp
661 .ne 2
662 .na
663 \fB\fBdevice\fR: match\fR
664 .ad
665 .sp .6
666 .RS 4n
667 Device name to match.
668 .RE

670 .sp
671 .ne 2
672 .na
673 \fB\fBrctl\fR: name, value\fR
674 .ad
675 .sp .6
676 .RS 4n
677 The name and \fIpriv\fR/\fIlimit\fR/\fIaction\fR triple of a resource control.
678 See \fBprctl\fR(1) and \fBrctladm\fR(1M). The preferred way to set rctl values
679 is to use the global property name associated with a specific rctl.
680 .RE

682 .sp
683 .ne 2
684 .na
685 \fB\fBattr\fR: name, type, value\fR
686 .ad
687 .sp .6
688 .RS 4n
689 The name, type and value of a generic attribute. The \fBtype\fR must be one of
690 \fBint\fR, \fBuint\fR, \fBboolean\fR or \fBstring\fR, and the value must be of
691 that type. \fBuint\fR means unsigned , that is, a non-negative integer.
692 .RE

694 .sp
695 .ne 2
696 .na
697 \fB\fBdataset\fR: name\fR
698 .ad
699 .sp .6
700 .RS 4n
701 The name of a \fBZFS\fR dataset to be accessed from within the zone. See
702 \fBzfs\fR(1M).
703 .RE

705 .sp
706 .ne 2
707 .na
708 \fBglobal: \fBcpu-shares\fR\fR
709 .ad
710 .sp .6
711 .RS 4n
712 The number of Fair Share Scheduler (FSS) shares to allocate to this zone. This
713 property is incompatible with the \fBdedicated-cpu\fR resource. This property
714 is the preferred way to set the \fBzone.cpu-shares\fR rctl.
715 .RE

717 .sp
718 .ne 2
719 .na
720 \fBglobal: \fBmax-lwps\fR\fR
721 .ad
```



```
722 .sp .6
723 .RS 4n
724 The maximum number of LWPs simultaneously available to this zone. This property
725 is the preferred way to set the \fBzone.max-lwps\fR rctl.
726 .RE

728 .sp
729 .ne 2
730 .na
731 \fBglobal: \fBmax-msg-ids\fR\fR
732 .ad
733 .sp .6
734 .RS 4n
735 The maximum number of message queue IDs allowed for this zone. This property is
736 the preferred way to set the \fBzone.max-msg-ids\fR rctl.
737 .RE

739 .sp
740 .ne 2
741 .na
742 \fBglobal: \fBmax-sem-ids\fR\fR
743 .ad
744 .sp .6
745 .RS 4n
746 The maximum number of semaphore IDs allowed for this zone. This property is the
747 preferred way to set the \fBzone.max-sem-ids\fR rctl.
748 .RE

750 .sp
751 .ne 2
752 .na
753 \fBglobal: \fBmax-shm-ids\fR\fR
754 .ad
755 .sp .6
756 .RS 4n
757 The maximum number of shared memory IDs allowed for this zone. This property is
758 the preferred way to set the \fBzone.max-shm-ids\fR rctl.
759 .RE

761 .sp
762 .ne 2
763 .na
764 \fBglobal: \fBmax-shm-memory\fR\fR
765 .ad
766 .sp .6
767 .RS 4n
768 The maximum amount of shared memory allowed for this zone. This property is the
769 preferred way to set the \fBzone.max-shm-memory\fR rctl. A scale (K, M, G, T)
770 can be applied to the value for this number (for example, 1M is one megabyte).
771 .RE

773 .sp
774 .ne 2
775 .na
776 \fBglobal: \fBscheduling-class\fR\fR
777 .ad
778 .sp .6
779 .RS 4n
780 Specifies the scheduling class used for processes running in a zone. When this
781 property is not specified, the scheduling class is established as follows:
782 .RS +4
783 .TP
784 .ie t \(bu
785 .el o
786 If the \fBcpu-shares\fR property or equivalent rctl is set, the scheduling
787 class FSS is used.
```

```
.RE
.RS +4
.TP
.ie t \(bu
.el o
If neither \fBcpu-shares\fR nor the equivalent rctl is set and the zone's pool
property references a pool that has a default scheduling class, that class is
used.
.RE
.RS +4
.TP
.ie t \(bu
.el o
Under any other conditions, the system default scheduling class is used.
.RE
.RE



.sp
.ne 2
.na
\fB\fBdedicated-cpu\fR: ncpus, importance\fR
.ad
.sp .6
.RS 4n
The number of CPUs that should be assigned for this zone's exclusive use. The
zone will create a pool and processor set when it boots. See \fBpooladm\fR(1M)
and \fBpoolcfg\fR(1M) for more information on resource pools. The \fBncpu\fR
property can specify a single value or a range (for example, 1-4) of
processors. The \fBimportance\fR property is optional; if set, it will specify
the \fBpset.importance\fR value for use by \fBpoold\fR(1M). If this resource is
used, there must be enough free processors to allocate to this zone when it
boots or the zone will not boot. The processors assigned to this zone will not
be available for the use of the global zone or other zones. This resource is
incompatible with both the \fBpool\fR and \fBcpu-shares\fR properties. Only a
single instance of this resource can be added to the zone.
.RE

.sp
.ne 2
.na
\fB\fBcapped-memory\fR: physical, swap, locked\fR
.ad
.sp .6
.RS 4n
The caps on the memory that can be used by this zone. A scale (K, M, G, T) can
be applied to the value for each of these numbers (for example, 1M is one
megabyte). Each of these properties is optional but at least one property must
be set when adding this resource. Only a single instance of this resource can
be added to the zone. The \fBphysical\fR property sets the \fBmax-rss\fR for
this zone. This will be enforced by \fBrcapd\fR(1M) running in the global zone.
The \fBswap\fR property is the preferred way to set the \fBzone.max-swap\fR
rctl. The \fBlocked\fR property is the preferred way to set the
\fBzone.max-locked-memory\fR rctl.
.RE

.sp
.ne 2
.na
\fB\fBcapped-cpu\fR: ncpus\fR
.ad
.sp .6
.RS 4n
Sets a limit on the amount of CPU time that can be used by a zone. The unit
used translates to the percentage of a single CPU that can be used by all user
```



```
threads in a zone, expressed as a fraction (for example, \fB\&.75\fR) or a
mixed number (whole number and fraction, for example, \fB1.25\fR). An
\fBncpu\fR value of \fB1\fR means 100% of a CPU, a value of \fB1.25\fR means
125%, \fB\&.75\fR mean 75%, and so forth. When projects within a capped zone
have their own caps, the minimum value takes precedence.
.sp
The \fBcapped-cpu\fR property is an alias for \fBzone.cpu-cap\fR resource
control and is related to the \fBzone.cpu-cap\fR resource control. See
\fBresource_controls\fR(5).
.RE

.sp
.ne 2
.mk
.na
\fBglobal: \fBfs-allowed\fR\fR
.ad
.sp .6
.RS 4n
A comma-separated list of additional filesystems that may be mounted within
the zone; for example "ufs,pcfs". By default, only hsfs(7fs) and network
filesystems can be mounted. If the first entry in the list is "-" then
that disables all of the default filesystems. If any filesystems are listed
after "-" then only those filesystems can be mounted.

This property does not apply to filesystems mounted into the zone via "add fs"
or "add dataset".

WARNING: allowing filesystem mounts other than the default may allow the zone
administrator to compromise the system with a malicious filesystem image, and
is not supported.
.RE

.sp
.LP
The following table summarizes resources, property-names, and types:
.sp
.in +2
.nf
resource          property-name   type
(global)          zonename        simple
(global)          zonepath        simple
(global)          autoboot        simple
(global)          bootargs        simple
(global)          pool            simple
(global)          limitpriv       simple
(global)          brand           simple
(global)          ip-type         simple
(global)          hostid          simple
(global)          cpu-shares      simple
(global)          max-lwps        simple
(global)          max-msg-ids     simple
(global)          max-sem-ids     simple
(global)          max-shm-ids     simple
(global)          max-shm-memory  simple
(global)          scheduling-class simple
fs                dir             simple
                   special         simple
                   raw             simple
                   type            simple
                   options         list of simple
net               address         simple
                   physical        simple
device            match           simple
rctl              name            simple
                   value           list of complex
```

```
 919 attr              name            simple
 920                    type            simple
 921                    value           simple
 922 dataset           name            simple
 923 dedicated-cpu     ncpus           simple or range
 924                    importance      simple

 926 capped-memory     physical        simple with scale
 927                    swap            simple with scale
 928                    locked          simple with scale

 930 capped-cpu        ncpus           simple
 931 .fi
 932 .in -2
 933 .sp

 935 .sp
 936 .LP
 937 To further specify things, the breakdown of the complex property "value" of the
 938 "rctl" resource type, it consists of three name/value pairs, the names being
 939 "priv", "limit" and "action", each of which takes a simple value. The "name"
 940 property of an "attr" resource is syntactically restricted in a fashion similar
 941 but not identical to zone names: it must begin with an alphanumeric, and can
 942 contain alphanumerics plus the hyphen (\fB-\fR), underscore (\fB_\fR), and dot
 943 (\fB\&.\fR) characters. Attribute names beginning with "zone" are reserved for
 944 use by the system. Finally, the "autoboot" global property must have a value of
 945 "true" or "false".
 946 .SS "Using Kernel Statistics to Monitor CPU Caps"
 947 .sp
 948 .LP
 949 Using the kernel statistics (\fBkstat\fR(3KSTAT)) module \fBcaps\fR, the system
 950 maintains information for all capped projects and zones. You can access this
 951 information by reading kernel statistics (\fBkstat\fR(3KSTAT)), specifying
 952 \fBcaps\fR as the \fBkstat\fR module name. The following command displays
 953 kernel statistics for all active CPU caps:
 954 .sp
 955 .in +2
 956 .nf
 957 # \fBkstat caps::'/cpucaps/'\fR
 958 .fi
 959 .in -2
 960 .sp

 962 .sp
 963 .LP
 964 A \fBkstat\fR(1M) command running in a zone displays only CPU caps relevant for
 965 that zone and for projects in that zone. See \fBEXAMPLES\fR.
 966 .sp
 967 .LP
 968 The following are cap-related arguments for use with \fBkstat\fR(1M):
 969 .sp
 970 .ne 2
 971 .na
 972 \fB\fBcaps\fR\fR
 973 .ad
 974 .sp .6
 975 .RS 4n
 976 The \fBkstat\fR module.
 977 .RE

 979 .sp
 980 .ne 2
 981 .na
 982 \fB\fBproject_caps\fR or \fBzone_caps\fR\fR
 983 .ad
 984 .sp .6
```



```
 985 .RS 4n
 986 \fBkstat\fR class, for use with the \fBkstat\fR \fB-c\fR option.
 987 .RE

 989 .sp
 990 .ne 2
 991 .na
 992 \fB\fBcpucaps_project_\fR\fIid\fR or \fBcpucaps_zone_\fR\fIid\fR\fR
 993 .ad
 994 .sp .6
 995 .RS 4n
 996 \fBkstat\fR name, for use with the \fBkstat\fR \fB-n\fR option. \fIid\fR is the
 997 project or zone identifier.
 998 .RE

1000 .sp
1001 .LP
1002 The following fields are displayed in response to a \fBkstat\fR(1M) command
1003 requesting statistics for all CPU caps.
1004 .sp
1005 .ne 2
1006 .na
1007 \fB\fBmodule\fR\fR
1008 .ad
1009 .sp .6
1010 .RS 4n
1011 In this usage of \fBkstat\fR, this field will have the value \fBcaps\fR.
1012 .RE

1014 .sp
1015 .ne 2
1016 .na
1017 \fB\fBname\fR\fR
1018 .ad
1019 .sp .6
1020 .RS 4n
1021 As described above, \fBcpucaps_project_\fR\fIid\fR or
1022 \fBcpucaps_zone_\fR\fIid\fR
1023 .RE

1025 .sp
1026 .ne 2
1027 .na
1028 \fB\fBabove_sec\fR\fR
1029 .ad
1030 .sp .6
1031 .RS 4n
1032 Total time, in seconds, spent above the cap.
1033 .RE

1035 .sp
1036 .ne 2
1037 .na
1038 \fB\fBbelow_sec\fR\fR
1039 .ad
1040 .sp .6
1041 .RS 4n
1042 Total time, in seconds, spent below the cap.
1043 .RE

1045 .sp
1046 .ne 2
1047 .na
1048 \fB\fBmaxusage\fR\fR
1049 .ad
1050 .sp .6
```

```
.RS 4n
Maximum observed CPU usage.
.RE

.sp
.ne 2
.na
\fB\fBnwait\fR\fR
.ad
.sp .6
.RS 4n
Number of threads on cap wait queue.
.RE

.sp
.ne 2
.na
\fB\fBusage\fR\fR
.ad
.sp .6
.RS 4n
Current aggregated CPU usage for all threads belonging to a capped project or
zone, in terms of a percentage of a single CPU.
.RE

.sp
.ne 2
.na
\fB\fBvalue\fR\fR
.ad
.sp .6
.RS 4n
The cap value, in terms of a percentage of a single CPU.
.RE

.sp
.ne 2
.na
\fB\fBzonename\fR\fR
.ad
.sp .6
.RS 4n
Name of the zone for which statistics are displayed.
.RE

.sp
.LP
See \fBEXAMPLES\fR for sample output from a \fBkstat\fR command.
.SH OPTIONS
.sp
.LP
The following options are supported:
.sp
.ne 2
.na
\fB\fB-f\fR \fIcommand_file\fR\fR
.ad
.sp .6
.RS 4n
Specify the name of \fBzonecfg\fR command file. \fIcommand_file\fR is a text
file of \fBzonecfg\fR subcommands, one per line.
.RE

.sp
.ne 2
.na
```



```
\fB\fB-z\fR \fIzonename\fR\fR
.ad
.sp .6
.RS 4n
Specify the name of a zone. Zone names are case sensitive. Zone names must
begin with an alphanumeric character and can contain alphanumeric characters,
the underscore (\fB_\fR) the hyphen (\fB-\fR), and the dot (\fB\&.\fR). The
name \fBglobal\fR and all names beginning with \fBSUNW\fR are reserved and
cannot be used.
.RE

.SH SUBCOMMANDS
.sp
.LP
You can use the \fBadd\fR and \fBselect\fR subcommands to select a specific
resource, at which point the scope changes to that resource. The \fBend\fR and
\fBcancel\fR subcommands are used to complete the resource specification, at
which time the scope is reverted back to global. Certain subcommands, such as
\fBadd\fR, \fBremove\fR and \fBset\fR, have different semantics in each scope.
.sp
.LP
\fBzonecfg\fR supports a semicolon-separated list of subcommands. For example:
.sp
.in +2
.nf
# \fBzonecfg -z myzone "add net; set physical=myvnic; end"\fR
.fi
.in -2
.sp

.sp
.LP
Subcommands which can result in destructive actions or loss of work have an
\fB-F\fR option to force the action. If input is from a terminal device, the
user is prompted when appropriate if such a command is given without the
\fB-F\fR option otherwise, if such a command is given without the \fB-F\fR
option, the action is disallowed, with a diagnostic message written to standard
error.
.sp
.LP
The following subcommands are supported:
.sp
.ne 2
.na
\fB\fBadd\fR \fIresource-type\fR (global scope)\fR
.ad
.br
.na
\fB\fBadd\fR \fIproperty-name property-value\fR (resource scope)\fR
.ad
.sp .6
.RS 4n
In the global scope, begin the specification for a given resource type. The
scope is changed to that resource type.
.sp
In the resource scope, add a property of the given name with the given value.
The syntax for property values varies with different property types. In
general, it is a simple value or a list of simple values enclosed in square
brackets, separated by commas (\fB[foo,bar,baz]\fR). See \fBPROPERTIES\fR.
.RE

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

```
1183 .sp .6
1184 .RS 4n
1185 End the resource specification and reset scope to global. Abandons any
1186 partially specified resources. \fBcancel\fR is only applicable in the resource
1187 scope.
1188 .RE

1190 .sp
1191 .ne 2
1192 .na
1193 \fB\fBclear\fR \fIproperty-name\fR\fR
1194 .ad
1195 .sp .6
1196 .RS 4n
1197 Clear the value for the property.
1198 .RE

1200 .sp
1201 .ne 2
1202 .na
1203 \fB\fBcommit\fR\fR
1204 .ad
1205 .sp .6
1206 .RS 4n
1207 Commit the current configuration from memory to stable storage. The
1208 configuration must be committed to be used by \fBzoneadm\fR. Until the
1209 in-memory configuration is committed, you can remove changes with the
1210 \fBrevert\fR subcommand. The \fBcommit\fR operation is attempted automatically
1211 upon completion of a \fBzonecfg\fR session. Since a configuration must be
1212 correct to be committed, this operation automatically does a verify.
1213 .RE

1215 .sp
1216 .ne 2
1217 .na
1218 \fB\fBcreate [\fR\fB-F\fR\fB] [\fR \fB-a\fR \fIpath\fR |\fB-b\fR \fB|\fR
1219 \fB-t\fR \fItemplate\fR\fB]\fR\fR
1220 .ad
1221 .sp .6
1222 .RS 4n
1223 Create an in-memory configuration for the specified zone. Use \fBcreate\fR to
1224 begin to configure a new zone. See \fBcommit\fR for saving this to stable
1225 storage.
1226 .sp
1227 If you are overwriting an existing configuration, specify the \fB-F\fR option
1228 to force the action. Specify the \fB-t\fR \fItemplate\fR option to create a
1229 configuration identical to \fItemplate\fR, where \fItemplate\fR is the name of
1230 a configured zone.
1231 .sp
1232 Use the \fB-a\fR \fIpath\fR option to facilitate configuring a detached zone on
1233 a new host. The \fIpath\fR parameter is the zonepath location of a detached
1234 zone that has been moved on to this new host. Once the detached zone is
1235 configured, it should be installed using the "\fBzoneadm attach\fR" command
1236 (see \fBzoneadm\fR(1M)). All validation of the new zone happens during the
1237 \fBattach\fR process, not during zone configuration.
1238 .sp
1239 Use the \fB-b\fR option to create a blank configuration. Without arguments,
1240 \fBcreate\fR applies the Sun default settings.
1241 .RE

1243 .sp
1244 .ne 2
1245 .na
1246 \fB\fBdelete [\fR\fB-F\fR\fB]\fR\fR
1247 .ad
1248 .sp .6
```



```
1249 .RS 4n
1250 Delete the specified configuration from memory and stable storage. This action
1251 is instantaneous, no commit is necessary. A deleted configuration cannot be
1252 reverted.
1253 .sp
1254 Specify the \fB-F\fR option to force the action.
1255 .RE

1257 .sp
1258 .ne 2
1259 .na
1260 \fB\fBend\fR\fR
1261 .ad
1262 .sp .6
1263 .RS 4n
1264 End the resource specification. This subcommand is only applicable in the
1265 resource scope. \fBzonecfg\fR checks to make sure the current resource is
1266 completely specified. If so, it is added to the in-memory configuration (see
1267 \fBcommit\fR for saving this to stable storage) and the scope reverts to
1268 global. If the specification is incomplete, it issues an appropriate error
1269 message.
1270 .RE

1272 .sp
1273 .ne 2
1274 .na
1275 \fB\fBexport [\fR\fB-f\fR \fIoutput-file\fR\fB]\fR\fR
1276 .ad
1277 .sp .6
1278 .RS 4n
1279 Print configuration to standard output. Use the \fB-f\fR option to print the
1280 configuration to \fIoutput-file\fR. This option produces output in a form
1281 suitable for use in a command file.
1282 .RE

1284 .sp
1285 .ne 2
1286 .na
1287 \fB\fBhelp [usage] [\fIsubcommand\fR] [syntax] [\fR\fIcommand-name\fR\fB]\fR\fR
1288 .ad
1289 .sp .6
1290 .RS 4n
1291 Print general help or help about given topic.
1292 .RE

1294 .sp
1295 .ne 2
1296 .na
1297 \fB\fBinfo zonename | zonepath | autoboot | brand | pool | limitpriv\fR\fR
1298 .ad
1299 .br
1300 .na
1301 \fB\fBinfo [\fR\fIresource-type\fR
1302 \fB[\fR\fIproperty-name\fR\fB=\fR\fIproperty-value\fR\fB]*]\fR\fR
1303 .ad
1304 .sp .6
1305 .RS 4n
1306 Display information about the current configuration. If \fIresource-type\fR is
1307 specified, displays only information about resources of the relevant type. If
1308 any \fIproperty-name\fR value pairs are specified, displays only information
1309 about resources meeting the given criteria. In the resource scope, any
1310 arguments are ignored, and \fBinfo\fR displays information about the resource
1311 which is currently being added or modified.
1312 .RE

1314 .sp
```

```
.ne 2
.na
\fB\fBremove\fR \fIresource-type\fR\fB{\fR\fIproperty-name\fR\fB=\fR\fIproperty
-value\fR\fB}\fR(global scope)\fR
.ad
.sp .6
.RS 4n
In the global scope, removes the specified resource. The \fB[]\fR syntax means
0 or more of whatever is inside the square braces. If you want only to remove a
single instance of the resource, you must specify enough property name-value
pairs for the resource to be uniquely identified. If no property name-value
pairs are specified, all instances will be removed. If there is more than one
pair is specified, a confirmation is required, unless you use the \fB-F\fR
option.
.RE

.sp
.ne 2
.na
\fB\fBselect\fR \fIresource-type\fR
\fB{\fR\fIproperty-name\fR\fB=\fR\fIproperty-value\fR\fB}\fR\fR
.ad
.sp .6
.RS 4n
Select the resource of the given type which matches the given
\fIproperty-name\fR \fIproperty-value\fR pair criteria, for modification. This
subcommand is applicable only in the global scope. The scope is changed to that
resource type. The \fB{}\fR syntax means 1 or more of whatever is inside the
curly braces. You must specify enough \fIproperty -name property-value\fR pairs
for the resource to be uniquely identified.
.RE

.sp
.ne 2
.na
\fB\fBset\fR \fIproperty-name\fR\fB=\fR\fIproperty\fR\fB-\fR\fIvalue\fR\fR
.ad
.sp .6
.RS 4n
Set a given property name to the given value. Some properties (for example,
\fBzonename\fR and \fBzonepath\fR) are global while others are
resource-specific. This subcommand is applicable in both the global and
resource scopes.
.RE

.sp
.ne 2
.na
\fB\fBverify\fR\fR
.ad
.sp .6
.RS 4n
Verify the current configuration for correctness:
.RS +4
.TP
.ie t \(bu
.el o
All resources have all of their required properties specified.
.RE
.RS +4
.TP
.ie t \(bu
.el o
A \fBzonepath\fR is specified.
.RE
.RE
```



```
.sp
.ne 2
.na
\fB\fBrevert\fR \fB[\fR\fB-F\fR\fB]\fR\fR
.ad
.sp .6
.RS 4n
Revert the configuration back to the last committed state. The \fB-F\fR option
can be used to force the action.
.RE

.sp
.ne 2
.na
\fB\fBexit [\fR\fB-F\fR\fB]\fR\fR
.ad
.sp .6
.RS 4n
Exit the \fBzonecfg\fR session. A commit is automatically attempted if needed.
You can also use an \fBEOF\fR character to exit \fBzonecfg\fR. The \fB-F\fR
option can be used to force the action.
.RE

.SH EXAMPLES
.LP
\fBExample 1 \fRCreating the Environment for a New Zone
.sp
.LP
In the following example, \fBzonecfg\fR creates the environment for a new zone.
\fB/usr/local\fR is loopback mounted from the global zone into
\fB/opt/local\fR. \fB/opt/sfw\fR is loopback mounted from the global zone,
three logical network interfaces are added, and a limit on the number of
fair-share scheduler (FSS) CPU shares for a zone is set using the \fBrctl\fR
resource type. The example also shows how to select a given resource for
modification.

.sp
.in +2
.nf
example# \fBzonecfg -z myzone3\fR
my-zone3: No such zone configured
Use 'create' to begin configuring a new zone.
zonecfg:myzone3> \fBcreate\fR
zonecfg:myzone3> \fBset zonepath=/export/home/my-zone3\fR
zonecfg:myzone3> \fBset autoboot=true\fR
zonecfg:myzone3> \fBadd fs\fR
zonecfg:myzone3:fs> \fBset dir=/usr/local\fR
zonecfg:myzone3:fs> \fBset special=/opt/local\fR
zonecfg:myzone3:fs> \fBset type=lofs\fR
zonecfg:myzone3:fs> \fBadd options [ro,nodevices]\fR
zonecfg:myzone3:fs> \fBend\fR
zonecfg:myzone3> \fBadd fs\fR
zonecfg:myzone3:fs> \fBset dir=/mnt\fR
zonecfg:myzone3:fs> \fBset special=/dev/dsk/c0t0d0s7\fR
zonecfg:myzone3:fs> \fBset raw=/dev/rdsk/c0t0d0s7\fR
zonecfg:myzone3:fs> \fBset type=ufs\fR
zonecfg:myzone3:fs> \fBend\fR
zonecfg:myzone3> \fBadd net\fR
zonecfg:myzone3:net> \fBset address=192.168.0.1/24\fR
zonecfg:myzone3:net> \fBset physical=eri0\fR
zonecfg:myzone3:net> \fBend\fR
zonecfg:myzone3> \fBadd net\fR
zonecfg:myzone3:net> \fBset address=192.168.1.2/24\fR
zonecfg:myzone3:net> \fBset physical=eri0\fR
zonecfg:myzone3:net> \fBend\fR
```

```
zonecfg:myzone3> \fBadd net\fR
zonecfg:myzone3:net> \fBset address=192.168.2.3/24\fR
zonecfg:myzone3:net> \fBset physical=eri0\fR
zonecfg:myzone3:net> \fBend\fR
zonecfg:my-zone3> \fBset cpu-shares=5\fR
zonecfg:my-zone3> \fBadd capped-memory\fR
zonecfg:my-zone3:capped-memory> \fBset physical=50m\fR
zonecfg:my-zone3:capped-memory> \fBset swap=100m\fR
zonecfg:my-zone3:capped-memory> \fBend\fR
zonecfg:myzone3> \fBexit\fR
.fi
.in -2
.sp

.LP
\fBExample 2 \fRCreating a Non-Native Zone
.sp
.LP
The following example creates a new Linux zone:

.sp
.in +2
.nf
example# \fBzonecfg -z lxzone\fR
lxzone: No such zone configured
Use ’create’ to begin configuring a new zone
zonecfg:lxzone> \fBcreate -t SUNWlx\fR
zonecfg:lxzone> \fBset zonepath=/export/zones/lxzone\fR
zonecfg:lxzone> \fBset autoboot=true\fR
zonecfg:lxzone> \fBexit\fR
.fi
.in -2
.sp

.LP
\fBExample 3 \fRCreating an Exclusive-IP Zone
.sp
.LP
The following example creates a zone that is granted exclusive access to
\fBbge1\fR and \fBbge33000\fR and that is isolated at the IP layer from the
other zones configured on the system.

.sp
.LP
The IP addresses and routing is configured inside the new zone using
\fBsysidtool\fR(1M).

.sp
.in +2
.nf
example# \fBzonecfg -z excl\fR
excl: No such zone configured
Use ’create’ to begin configuring a new zone
zonecfg:excl> \fBcreate\fR
zonecfg:excl> \fBset zonepath=/export/zones/excl\fR
zonecfg:excl> \fBset ip-type=exclusive\fR
zonecfg:excl> \fBadd net\fR
zonecfg:excl:net> \fBset physical=bge1\fR
zonecfg:excl:net> \fBend\fR
zonecfg:excl> \fBadd net\fR
zonecfg:excl:net> \fBset physical=bge33000\fR
zonecfg:excl:net> \fBend\fR
zonecfg:excl> \fBexit\fR
.fi
.in -2
.sp
```



```
.LP
\fBExample 4 \fRAssociating a Zone with a Resource Pool
.sp
.LP
The following example shows how to associate an existing zone with an existing
resource pool:

.sp
.in +2
.nf
example# \fBzonecfg -z myzone\fR
zonecfg:myzone> \fBset pool=mypool\fR
zonecfg:myzone> \fBexit\fR
.fi
.in -2
.sp

.sp
.LP
For more information about resource pools, see \fBpooladm\fR(1M) and
\fBpoolcfg\fR(1M).

.LP
\fBExample 5 \fRChanging the Name of a Zone
.sp
.LP
The following example shows how to change the name of an existing zone:

.sp
.in +2
.nf
example# \fBzonecfg -z myzone\fR
zonecfg:myzone> \fBset zonename=myzone2\fR
zonecfg:myzone2> \fBexit\fR
.fi
.in -2
.sp

.LP
\fBExample 6 \fRChanging the Privilege Set of a Zone
.sp
.LP
The following example shows how to change the set of privileges an existing
zone’s processes will be limited to the next time the zone is booted. In this
particular case, the privilege set will be the standard safe set of privileges
a zone normally has along with the privilege to change the system date and
time:

.sp
.in +2
.nf
example# \fBzonecfg -z myzone\fR
zonecfg:myzone> \fBset limitpriv="default,sys_time"\fR
zonecfg:myzone2> \fBexit\fR
.fi
.in -2
.sp

.LP
\fBExample 7 \fRSetting the \fBzone.cpu-shares\fR Property for the Global Zone
.sp
.LP
The following command sets the \fBzone.cpu-shares\fR property for the global
zone:
```

```
.sp
.in +2
.nf
example# \fBzonecfg -z global\fR
zonecfg:global> \fBset cpu-shares=5\fR
zonecfg:global> \fBexit\fR
.fi
.in -2
.sp

.LP
\fBExample 8 \fRUsing Pattern Matching
.sp
.LP
The following commands illustrate \fBzonecfg\fR support for pattern matching.
In the zone \fBflexlm\fR, enter:

.sp
.in +2
.nf
zonecfg:flexlm> \fBadd device\fR
zonecfg:flexlm:device> \fBset match="/dev/cua/a00[2-5]"\fR
zonecfg:flexlm:device> \fBend\fR
.fi
.in -2
.sp

.sp
.LP
In the global zone, enter:

.sp
.in +2
.nf
global# \fBls /dev/cua\fR
a     a000  a001  a002  a003  a004  a005  a006  a007  b
.fi
.in -2
.sp

.sp
.LP
In the zone \fBflexlm\fR, enter:

.sp
.in +2
.nf
flexlm# \fBls /dev/cua\fR
a002  a003  a004  a005
.fi
.in -2
.sp

.LP
\fBExample 9 \fRSetting a Cap for a Zone to Three CPUs
.sp
.LP
The following sequence uses the \fBzonecfg\fR command to set the CPU cap for a
zone to three CPUs.

.sp
.in +2
.nf
zonecfg:myzone> \fBadd capped-cpu\fR
zonecfg:myzone>capped-cpu> \fBset ncpus=3\fR
zonecfg:myzone>capped-cpu>capped-cpu> \fBend\fR
```



```
.fi
.in -2
.sp

.sp
.LP
The preceding sequence, which uses the capped-cpu property, is equivalent to
the following sequence, which makes use of the \fBzone.cpu-cap\fR resource
control.

.sp
.in +2
.nf
zonecfg:myzone> \fBadd rctl\fR
zonecfg:myzone:rctl> \fBset name=zone.cpu-cap\fR
zonecfg:myzone:rctl> \fBadd value (priv=privileged,limit=300,action=none)\fR
zonecfg:myzone:rctl> \fBend\fR
.fi
.in -2
.sp

.LP
\fBExample 10 \fRUsing \fBkstat\fR to Monitor CPU Caps
.sp
.LP
The following command displays information about all CPU caps.

.sp
.in +2
.nf
# \fBkstat -n /cpucaps/\fR
module: caps                            instance: 0
name:   cpucaps_project_0               class:    project_caps
        above_sec                       0
        below_sec                       2157
        crtime                          821.048183159
        maxusage                        2
        nwait                           0
        snaptime                        235885.637253027
        usage                           0
        value                           18446743151372347932
        zonename                        global

module: caps                            instance: 0
name:   cpucaps_project_1               class:    project_caps
        above_sec                       0
        below_sec                       0
        crtime                          225339.192787265
        maxusage                        5
        nwait                           0
        snaptime                        235885.637591677
        usage                           5
        value                           18446743151372347932
        zonename                        global

module: caps                            instance: 0
name:   cpucaps_project_201             class:    project_caps
        above_sec                       0
        below_sec                       235105
        crtime                          780.37961782
        maxusage                        100
        nwait                           0
        snaptime                        235885.637789687
        usage                           43
        value                           100
        zonename                        global
```

```
module: caps                    instance: 0
name:   cpucaps_project_202     class:    project_caps
        above_sec               0
        below_sec               235094
        crtime                  791.72983782
        maxusage                100
        nwait                   0
        snaptime                235885.637967512
        usage                   48
        value                   100
        zonename                global

module: caps                    instance: 0
name:   cpucaps_project_203     class:    project_caps
        above_sec               0
        below_sec               235034
        crtime                  852.104401481
        maxusage                75
        nwait                   0
        snaptime                235885.638144304
        usage                   47
        value                   100
        zonename                global

module: caps                    instance: 0
name:   cpucaps_project_86710   class:    project_caps
        above_sec               22
        below_sec               235166
        crtime                  698.441717859
        maxusage                101
        nwait                   0
        snaptime                235885.638319871
        usage                   54
        value                   100
        zonename                global

module: caps                    instance: 0
name:   cpucaps_zone_0          class:    zone_caps
        above_sec               100733
        below_sec               134332
        crtime                  821.048177123
        maxusage                207
        nwait                   2
        snaptime                235885.638497731
        usage                   199
        value                   200
        zonename                global

module: caps                    instance: 1
name:   cpucaps_project_0       class:    project_caps
        above_sec               0
        below_sec               0
        crtime                  225360.256448422
        maxusage                7
        nwait                   0
        snaptime                235885.638714404
        usage                   7
        value                   18446743151372347932
        zonename                test_001

module: caps                    instance: 1
name:   cpucaps_zone_1          class:    zone_caps
        above_sec               2
        below_sec               10524
        crtime                  225360.256440278
```



```
        maxusage                106
        nwait                   0
        snaptime                235885.638896443
        usage                   7
        value                   100
        zonename                test_001
.fi
.in -2
.sp

.LP
\fBExample 11 \fRDisplaying CPU Caps for a Specific Zone or Project
.sp
.LP
Using the \fBkstat\fR \fB-c\fR and \fB-i\fR options, you can display CPU caps
for a specific zone or project, as below. The first command produces a display
for a specific project, the second for the same project within zone 1.

.sp
.in +2
.nf
# \fBkstat -c project_caps\fR

# \fBkstat -c project_caps -i 1\fR
.fi
.in -2
.sp

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

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

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

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

```
1843 .sp

1845 .sp
1846 .TS
1847 box;
1848 c | c
1849 l | l .
1850 ATTRIBUTE TYPE  ATTRIBUTE VALUE
1851 _
1852 Interface Stability     Volatile
1853 .TE

1855 .SH SEE ALSO
1856 .sp
1857 .LP
1858 \fBppriv\fR(1), \fBprctl\fR(1), \fBzlogin\fR(1), \fBkstat\fR(1M),
1859 \fBmount\fR(1M), \fBpooladm\fR(1M), \fBpoolcfg\fR(1M), \fBpoold\fR(1M),
1860 \fBrcapd\fR(1M), \fBrctladm\fR(1M), \fBsvcadm\fR(1M), \fBsysidtool\fR(1M),
1861 \fBzfs\fR(1M), \fBzoneadm\fR(1M), \fBpriv_str_to_set\fR(3C),
1862 \fBkstat\fR(3KSTAT), \fBvfstab\fR(4), \fBattributes\fR(5), \fBbrands\fR(5),
1863 \fBfnmatch\fR(5), \fBlx\fR(5), \fBprivileges\fR(5), \fBresource_controls\fR(5),
1864 \fBzones\fR(5)
1865 .sp
1866 .LP
1867 \fISystem Administration Guide: Solaris Containers-Resource Management, and
1868 Solaris Zones\fR
1869 .SH NOTES
1870 .sp
1871 .LP
1872 All character data used by \fBzonecfg\fR must be in US-ASCII encoding.
```

```
**********************************************************
    5997 Thu Jul 17 00:50:40 2014
new/usr/src/man/man2/pipe.2
manpage lint.
**********************************************************
'\" te
.\" Copyright (c) 2002, Sun Microsystems, Inc.  All Rights Reserved.
.\" Copyright 1989 AT&T
.\" Portions Copyright (c) 2001, the Institute of Electrical and Electronics Eng
.\" Portions Copyright (c) 2013, OmniTI Computer Consulting, Inc. All Rights Res
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for permission
.\" http://www.opengroup.org/bookstore/.
.\" The Institute of Electrical and Electronics Engineers and The Open Group, ha
.\"  This notice shall appear on any product containing this material.
.\" 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 PIPE 2 "Apr 19, 2013"
.SH NAME
pipe \- create an interprocess channel
.SH SYNOPSIS
.LP
.nf
#include <unistd.h>

\fBint\fR \fBpipe\fR(\fBint\fR \fIfildes\fR[2]);

\fBint\fR \fBpipe2\fR(\fBint\fR \fIfildes\fR[2], \fBint\fR \fIflags\fR);
.fi

.SH DESCRIPTION
.sp
.LP
The \fBpipe()\fR and pipe2() functions create an I/O mechanism called a
pipe and returns two file descriptors, \fIfildes\fR[\fB0\fR] and
\fIfildes\fR[\fB1\fR]. The files associated with \fIfildes\fR[\fB0\fR]
and \fIfildes\fR[\fB1\fR] are streams and are both opened for reading and
writing.  The \fBpipe()\fR call will clear the \fBO_NDELAY\fR,
\fBO_NONBLOCK\fR, and \fBFD_CLOEXEC\fR flags on both file descriptors. The
\fBfcntl\fR(2) function can be used to set these flags.
.sp
.LP
The \fBpipe2()\fR call will clear the \fBO_NDELAY\fR on both filedescriptors.
The \fIflags\fR argument may be used to specify attributes on both file
descriptors.  \fBpipe2()\fR called with a \fIflags\fR value of 0 will
behave identically to \fBpipe()\fR.  Values for \fIflags\fR are constructed
by a bitwise-inclusive-OR of flags from the following list, defined in
<\fBfcntl.h\fR>.
.RE

.sp
.ne 2
.na
\fB\fBO_NONBLOCK\fR\fR
.ad
.RS 12n
Both file descriptors will be placed in non-blocking mode. This corresponds
to the \fBO_NONBLOCK\fR flag to \fBfcntl\fR(2).
.RE

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



```
Both file descriptors will be opened with the FD_CLOEXEC flag set. Both file
descriptors will be closed prior to any future exec() calls.
.RE

.sp
.LP
A read from \fIfildes\fR[\fB0\fR] accesses the data written to
\fIfildes\fR[\fB1\fR] on a first-in-first-out (FIFO) basis and a read from
\fIfildes\fR[\fB1\fR] accesses the data written to \fIfildes\fR[\fB0\fR] also
on a \fBFIFO\fR basis.
.sp
.LP
Upon successful completion \fBpipe()\fR marks for update the \fBst_atime\fR,
\fBst_ctime\fR, and \fBst_mtime\fR fields of the pipe.
.SH RETURN VALUES
.sp
.LP
Upon successful completion, \fB0\fR is returned. Otherwise, \fB\(mi1\fR is
returned and \fBerrno\fR is set to indicate the error.
.SH ERRORS
.sp
.LP
The \fBpipe()\fR and \fBpipe2()\fR functions will fail if:
.sp
.ne 2
.na
\fB\fBEMFILE\fR\fR
.ad
.RS 10n
More than {\fBOPEN_MAX\fR} file descriptors are already in use by this process.
.RE

.sp
.ne 2
.na
\fB\fBENFILE\fR\fR
.ad
.RS 10n
The number of simultaneously open files in the system would exceed a
system-imposed limit.
.RE

.sp
.ne 2
.na
\fB\fBEFAULT\fR\fR
.ad
.RS 10n
The \fIfildes[2]\fR argument points to an illegal address.
.RE

.sp
.LP
The \fBpipe2()\fR function will also fail if:
.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 10n
The \fIflags\fR argument is illegal. Valid \fIflags\fR are zero or a
bitwise inclusive-OR of \fBO_CLOEXEC\fR and \fBO_NONBLOCK\fR.
.RE


.SH ATTRIBUTES
```

```
 127 .sp
 128 .LP
 129 See \fBattributes\fR(5) for descriptions of the following attributes:
 130 .sp

 132 .sp
 133 .TS
 134 box;
 135 c | c
 136 l | l .
 137 ATTRIBUTE TYPE  ATTRIBUTE VALUE
 138 _
 139 Interface Stability     Standard
 140 _
 141 MT-Level        Async-Signal-Safe
 142 .TE

 144 .SH SEE ALSO
 145 .sp
 146 .LP
 147 \fBsh\fR(1), \fBfcntl\fR(2), \fBfstat\fR(2), \fBgetmsg\fR(2), \fBpoll\fR(2),
 148 \fBputmsg\fR(2), \fBread\fR(2), \fBwrite\fR(2), \fBopen\fR(2),
 149 \fBattributes\fR(5), \fBstandards\fR(5), \fBstreamio\fR(7I)
 150 .SH NOTES
 151 .sp
 152 .LP
 153 Since a pipe is bi-directional, there are two separate flows of data.
 154 Therefore, the size (\fBst_size\fR) returned by a call to \fBfstat\fR(2) with
 155 argument  \fIfildes\fR[\fB0\fR] or \fIfildes\fR[\fB1\fR] is the number of bytes
 156 available for reading from \fIfildes\fR[\fB0\fR] or \fIfildes\fR[\fB1\fR]
 157 respectively. Previously, the size (\fBst_size\fR) returned by a call to
 158 \fBfstat()\fR with argument \fIfildes\fR[\fB1\fR] (the write-end) was the
 159 number of bytes available for reading from \fIfildes\fR[\fB0\fR] (the
 160 read-end).
```

```
**********************************************************
    3682 Thu Jul 17 00:50:40 2014
new/usr/src/man/man3c/btowc.3c
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
    6351 Thu Jul 17 00:50:40 2014
new/usr/src/man/man3c/fgetwc.3c
manpage lint.
**********************************************************
```

```
   1 '\" te
   2 .\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
   3 .\" Copyright (c) 2003, X/Open Company Limited.  All Rights Reserved.  Portions
   4 .\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for permission
   5 .\" http://www.opengroup.org/bookstore/.
   6 .\" The Institute of Electrical and Electronics Engineers and The Open Group, ha
   7 .\"  This notice shall appear on any product containing this material.
   8 .\" The contents of this file are subject to the terms of the Common Development
   9 .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE or http:
  10 .\" When distributing Covered Code, include this CDDL HEADER in each file and in
  11 .TH FGETWC 3C "Jun 24, 2014"
  12 .SH NAME
  13 fgetwc, fgetwc_l \- get a wide-character code from a stream
  14 .SH SYNOPSIS
  15 .LP
  16 .nf
  17 #include <stdio.h>
  18 #include <wchar.h>

  20 \fBwint_t\fR \fBfgetwc\fR(\fBFILE *\fR\fIstream\fR);
  21 .fi
  22 .LP
  23 .nf
  24 #include <stdio.h>
  25 #include <wchar.h>
  26 #include <xlocale.h>

  28 \fBwint_t\fR \fBfgetwc_l\fR(\fBFILE *\fR\fIstream\fR, \fBlocale_t\fR, \fIloc\fR)
  29 .fi

  31 .SH DESCRIPTION
  32 .sp
  33 .LP
  34 The \fBfgetwc()\fR and \fBfgetwc_l()\fR functions obtain the next
  35 character (if present) from the input stream pointed to by \fIstream\fR,
  36 convert that to the corresponding wide-character code and advance the
  37 associated file position indicator for the stream (if defined).
  38 Whereas \fBfgetwc()\fR uses the current locale, \fBfgetwc_l()\fR uses the
  39 locale specified by \fIloc\fR.
  39 locale specified by \fIloc\R.
  40 .LP
  41 If an error occurs, the resulting value of the file position indicator for the
  42 stream is indeterminate.
  43 .LP
  44 The \fBfgetwc()\fR and \fBfgetwc_l()\fR functions may mark the \fBst_atime\fR
  45 field of the file
  46 associated with \fIstream\fR for update. The \fBst_atime\fR field will be
  47 marked for update by the first successful execution of \fBfgetwc()\fR,
  48 \fBfgetc\fR(3C), \fBfgets\fR(3C), \fBfgetws\fR(3C), \fBfread\fR(3C),
  49 \fBfscanf\fR(3C), \fBgetc\fR(3C), \fBgetchar\fR(3C), \fBgets\fR(3C), or
  50 \fBscanf\fR(3C) using \fIstream\fR that returns data not supplied by a prior
  51 call to \fBungetc\fR(3C) or \fBungetwc\fR(3C).
  52 .SH RETURN VALUES
  53 .LP
  54 Upon successful completion both functions return the
  55 wide-character code of the character read from the input stream pointed to by
  56 \fIstream\fR converted to a type \fBwint_t\fR.
  57 .LP
  58 For standard-conforming (see \fBstandards\fR(5)) applications, if the
  59 end-of-file indicator for the stream is set, \fBfgetwc()\fR and
  60 \fBfgetwc_l()\fR return \fBWEOF\fR whether or not the stream is at
```



```
  61 end-of-file.
  62 .LP
  63 If a read error occurs, the error indicator for the stream is set,
  64 \fBfgetwc()\fR and \fBfgetwc_l()\fR returns \fBWEOF\fR and sets
  65 \fBerrno\fR to indicate the error.
  66 .LP
  67 If an encoding error occurs, the error indicator for the stream is set,
  68 \fBfgetwc()\fR and \fBfgetwc_l()\fR return \fBWEOF\fR, and \fBerrno\fR is
  69 set to indicate the error.
  70 .SH ERRORS
  71 .LP
  72 The \fBfgetwc()\fR and \fBfgetwc_l()\fR functions will fail if data needs to be
  73 read and:
  74 .TP
  75 .B EAGAIN
  76 The \fBO_NONBLOCK\fR flag is set for the file descriptor underlying
  77 \fIstream\fR and the process would be delayed in the \fBfgetwc()\fR or
  78 \fBfgetwc_l()\fR operation.
  79 .TP
  80 .B EBADF
  81 The file descriptor underlying \fIstream\fR is not a valid file descriptor open
  82 for reading.
  83 .TP
  84 .B EINTR
  85 The read operation was terminated due to the receipt of a signal, and no data
  86 was transferred.
  87 .TP
  88 .B EIO
  89 A physical I/O error has occurred, or the process is in a background process
  90 group attempting to read from its controlling terminal and either the process
  91 is ignoring or blocking the \fBSIGTTIN\fR signal or the process group is
  92 orphaned.
  93 .TP
  94 .B EOVERFLOW
  95 The file is a regular file and an attempt was made to read at or beyond the
  96 offset maximum associated with the corresponding  \fIstream\fR.
  97 .LP
  98 The \fBfgetwc()\fR and \fBfgetwc_l()\fR functions may fail if:
  99 .TP
 100 .B ENOMEM
 101 Insufficient memory is available.
 102 .TP
 103 .B ENXIO
 104 A request was made of a non-existent device, or the request was outside the
 105 capabilities of the device.
 106 .TP
 107 .B EILSEQ
 108 The data obtained from the input stream does not form a valid character.
 109 .SH USAGE
 110 .sp
 111 .LP
 112 The \fBferror\fR(3C) or \fBfeof\fR(3C) functions must be used to distinguish
 113 between an error condition and an end-of-file condition.
 114 .SH ATTRIBUTES
 115 .sp
 116 .LP
 117 See \fBattributes\fR(5) for descriptions of the following attributes:
 118 .TS
 119 box;
 120 c | c
 121 l | l .
 122 ATTRIBUTE TYPE  ATTRIBUTE VALUE
 123 _
 124 CSI     Enabled
 125 _
 126 Interface Stability     See below.
```

```
 127 _
 128 MT-Level        MT-Safe
 129 .TE

 131 .LP
 132 The
 133 .B fgetwc()
 134 function is Standard.  The
 135 .B fgetwc_l()
 136 function is Uncommitted.
 137 .SH SEE ALSO
 138 .LP
 139 \fBfeof\fR(3C), \fBferror\fR(3C), \fBfgetc\fR(3C), \fBfgets\fR(3C),
 140 \fBfgetws\fR(3C), \fBfopen\fR(3C), \fBfread\fR(3C), \fBfscanf\fR(3C),
 141 \fBgetc\fR(3C), \fBgetchar\fR(3C), \fBgets\fR(3C), \fBscanf\fR(3C),
 142 \fBnewlocale\fR(3C), \fBsetlocale\fR(3C), \fBungetc\fR(3C), \fBungetwc\fR(3C),
 143 \fBuselocale\fR(3C), \fBattributes\fR(5),
 144 \fBstandards\fR(5)
```

```
**********************************************************
   12307 Thu Jul 17 00:50:40 2014
new/usr/src/man/man3c/fopen.3c
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
    4950 Thu Jul 17 00:50:40 2014
new/usr/src/man/man3c/iswctype.3c
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
    6226 Thu Jul 17 00:50:41 2014
new/usr/src/man/man3c/mbrtowc.3c
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
    6940 Thu Jul 17 00:50:41 2014
new/usr/src/man/man3c/mbsrtowcs.3c
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
    4309 Thu Jul 17 00:50:41 2014
new/usr/src/man/man3c/newlocale.3c
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
    4621 Thu Jul 17 00:50:41 2014
new/usr/src/man/man3c/strcoll.3c
manpage lint.
**********************************************************
'\" te
.\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
.\" Copyright (c) 2003, Sun Microsystems, Inc.  All Rights Reserved.
.\" Copyright 1989 AT&T
.\" Portions Copyright (c) 2001, the Institute of Electrical and Electronics Eng
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for permission
.\" http://www.opengroup.org/bookstore/.
.\" The Institute of Electrical and Electronics Engineers and The Open Group, ha
.\"  This notice shall appear on any product containing this material.
.\" 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 STRCOLL 3C "Jun 23, 2014"
.SH NAME
strcoll, strcoll_l \- string collation
.SH SYNOPSIS
.LP
.nf
#include <string.h>

\fBint\fR \fBstrcoll\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fIs2\fR);
.fi
.LP
.nf
\fBint\fR \fBstrcoll_l\fR(\fBconst char *\fR\fIs1\fR, \fBconst char *\fR\fIs2\fR
.fi

.SH DESCRIPTION
.LP
Both \fBstrcoll()\fR and \fBstrxfrm\fR(3C) provide for locale-specific string
sorting.  \fBstrcoll()\fR is intended for applications in which the number of
comparisons per string is small. When strings are to be compared a number of
times, \fBstrxfrm\fR(3C) is a more appropriate function because the
transformation process occurs only once.
.LP
The \fBstrcoll_l()\fR function behaves
identically to \fBstrcoll()\fR, except instead of operating in the current
locale, it operates in the locale specified by \fIloc\fR.
.LP
The \fBstrcoll()\fR function does not change the setting of \fBerrno\fR if
successful.
.LP
Since no return value is reserved to indicate an error, an application wishing
to check for error situations should set \fBerrno\fR to 0, then call
\fBstrcoll()\fR, then check \fBerrno\fR.
.SH RETURN VALUES
.LP
Upon successful completion, \fBstrcoll()\fR returns an integer greater than,
equal to, or less than zero in direct correlation to whether string \fIs1\fR is
greater than, equal to, or less than the string \fIs2\fR.  The comparison is
based on strings interpreted as appropriate to the locale
category \fBLC_COLLATE\fR (see \fBsetlocale\fR(3C)).
.LP
On error, \fBstrcoll()\fR may set \fBerrno\fR, but no return value is reserved
to indicate an error.
.SH ERRORS
.sp
.LP
The \fBstrcoll()\fR and \fBstrcoll_l()\fR functions may fail if:
.sp
.ne 2
```



```
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 10n
The \fIs1\fR or \fIs2\fR arguments contain characters outside the domain of the
collating sequence.
.RE
.SH FILES
.IP \fB/usr/lib/locale/\fR\fIlocale\fR\fB/LC_COLLATE/*\fR
collation database for \fIlocale\fR
.RE

.SH ATTRIBUTES
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
CSI     Enabled
_
Interface Stability     Standard
_
MT-Level        MT-Safe
.TE

.SH SEE ALSO
.sp
.LP
\fBlocaledef\fR(1), \fBnewlocale\fR(3C), \fBsetlocale\fR(3C), \fBstring\fR(3C),
\fBstrxfrm\fR(3C), \fBuselocale\fR(3C),
\fBwsxfrm\fR(3C), \fBattributes\fR(5), \fBenviron\fR(5), \fBstandards\fR(5)
```

```
**********************************************************
   21322 Thu Jul 17 00:50:41 2014
new/usr/src/man/man3c/string.3c
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
    1767 Thu Jul 17 00:50:41 2014
new/usr/src/man/man3c/towlower.3c
manpage lint.
**********************************************************
   1 '\" te
   2 .\"
   3 .\" This file and its contents are supplied under the terms of the
   4 .\" Common Development and Distribution License ("CDDL"), version 1.0.
   5 .\" You may only use this file in accordance with the terms of version
   6 .\" 1.0 of the CDDL.
   7 .\"
   8 .\" A full copy of the text of the CDDL should have accompanied this
   9 .\" source.  A copy of the CDDL is also available via the Internet at
  10 .\" http://www.illumos.org/license/CDDL.
  11 .\"
  12 .\"
  13 .\" Copyright (c) 2014 Joyent, Inc.  All rights reserved.
  14 .\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
  15 .\"
  16 .TH TOWLOWER 3C "Jun 21, 2014"
  17 .SH NAME
  18 towlower, towlower_l \- transliterate upper-case wide characters to lower-case
  19 .SH SYNOPSIS
  20 .LP
  21 .nf
  22 #include <wctype.h>

  24 \fBwint_t\fR \fBtowlower\fR(\fBwint_t\fR \fIwc\fR);
  25 .fi
  26 .LP
  27 .nf
  28 \fBwint_t\fR \fBtowlower_l\fR(\fBwint_t\fR \fIwc\fR, \fBlocale_t\fR \fIloc\fR);
  29 .fi
  29 .nf
  30 .SH DESCRIPTION
  31 The function
  32 .BR towlower()
  33 is the wide character equivalent of the function
  34 .BR tolower (3C).
  35 It converts the upper-case wide character
  36 .I wc
  37 to the equivalent lower-case
  38 wide character, if one exists. If one does not exist, it returns
  39 .I wc
  40 unchanged.
  41 .LP
  42 The function
  43 .B towlower_l()
  44 is equivalent to the function
  45 .BR towlower() ,
  46 but instead of operating in the current locale, operates in the
  47 locale specified by
  48 .IR loc .
  49 .SH RETURN VALUES
  50 On successful completion,
  51 .B towlower()
  52 and
  53 .B towlower_l()
  54 return the lower-case character that corresponds to the argument passed.
  55 Otherwise, they return the argument unchanged.
  56 .SH ERRORS
  57 No errors are defined.
  58 .SH ATTRIBUTES
  59 .TS
  60 box;
```



```
  61 c | c
  62 l | l .
  63 ATTRIBUTE TYPE  ATTRIBUTE VALUE
  64 _
  65 Interface Stability     Standard
  66 _
  67 MT-Level        MT-Safe
  68 .TE

  70 .SH SEE ALSO
  71 .BR newlocale (3C),
  72 .BR setlocale (3C),
  73 .BR towupper (3C),
  74 .BR uselocale (3C),
  75 .BR locale (5)
```

```
**********************************************************
    2323 Thu Jul 17 00:50:41 2014
new/usr/src/man/man3c/uselocale.3c
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
    4735 Thu Jul 17 00:50:41 2014
new/usr/src/man/man3c/wcscoll.3c
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
    6901 Thu Jul 17 00:50:41 2014
new/usr/src/man/man3c/wcsrtombs.3c
manpage lint.
**********************************************************
'\" te
.\" Copyright 2014 Garrett D'Amore <garrett@damore.org>
.\" Copyright (c) 1992, X/Open Company Limited. All Rights Reserved.  Portions C
.\" Sun Microsystems, Inc. gratefully acknowledges The Open Group for permission
.\" http://www.opengroup.org/bookstore/.
.\" The Institute of Electrical and Electronics Engineers and The Open Group, ha
.\"  This notice shall appear on any product containing this material.
.\" 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 WCSRTOMBS 3C "Jul 13, 2014"
.SH NAME
wcsnrtombs, wcsnrtombs_l, wcsrtombs, wcsrtombs_l \- convert a wide-character str
(restartable)
.SH SYNOPSIS
.LP
.nf
#include <wchar.h>

\fBsize_t\fR \fBwcsrtombs\fR(\fBchar *restrict\fR \fIdst\fR, \fBconst wchar_t **
     \fBsize_t\fR \fIlen\fR, \fBmbstate_t *restrict\fR \fIps\fR);
.fi
.LP
.nf
\fBsize_t\fR \fBwcsnrtombs\fR(\fBchar *restrict\fR \fIdst\fR, \fBconst wchar_t *
     \fBsize_t\fR \fInwc\fR, \fBsize_t\fR \fIlen\fR, \fBmbstate_t *restrict\fR \
.fi
.LP
.nf
#include <wchar.h>
#include <xlocale.h>

\fBsize_t\fR \fBwcsrtombs_l\fR(\fBchar *restrict\fR \fIdst\fR, \fBconst wchar_t
     \fBsize_t\fR \fIlen\fR, \fBmbstate_t *restrict\fR \fIps\fR, \fBlocale_t\fR
.fi
.LP
.nf
\fBsize_t\fR \fBwcsnrtombs_l\fR(\fBchar *restrict\fR \fIdst\fR, \fBconst wchar_t
     \fBsize_t\fR \fInwc\fR, \fBsize_t\fR \fIlen\fR, \fBmbstate_t *restrict\fR \
.fi
.SH DESCRIPTION
.LP
The \fBwcsrtombs()\fR function converts a sequence of wide-characters from the
array indirectly pointed to by \fIsrc\fR into a sequence of corresponding
characters, beginning in the conversion state described by the object pointed
to by \fIps\fR. If \fIdst\fR is not a null pointer, the converted characters
are then stored into the array pointed to by \fIdst\fR. Conversion continues up
to and including a terminating null  wide-character, which is also stored.
Conversion stops earlier in the following cases:
.RS +4
.TP
.ie t \(bu
.el o
When a code is reached that does not correspond to a valid character.
.RE
.RS +4
.TP
.ie t \(bu
.el o
When the next character would exceed the limit of \fIlen\fR total bytes to be
stored in the array pointed to by \fIdst\fR (and \fIdst\fR is not a null
```



```
pointer).
.RE
.RS +4
.TP
.ie t \(bu
.el o
In the case of \fBwcsnrtombs()\fR and \fBwcsnrtombs_l()\fR, when \fInwc\fR
wide characters have been completely converted.
.RE
.LP
Each conversion takes place as if by a call to the \fBwcrtomb()\fR function.
.LP
If \fIdst\fR is not a null pointer, the pointer object pointed to by \fIsrc\fR
is assigned either a null pointer (if conversion stopped due to reaching a
terminating null wide-character) or the address just  past the last
wide-character converted (if any). If conversion stopped due to reaching a
terminating null wide-character, the resulting state described is the initial
conversion state.
.LP
If \fIps\fR is a null pointer, these functions uses their own
internal \fBmbstate_t\fR object, which is initialized at program startup to the
initial conversion state.  Otherwise, the  \fBmbstate_t\fR object pointed to by
\fIps\fR is used to completely describe the current conversion state of the
associated character sequence. The system will behave as if no function defined
in the Reference Manual calls any of these functions.
.LP
The behavior of \fBwcsrtombs()\fR and \fBwcsnrtombs()\fR are affected by the
\fBLC_CTYPE\fR category of the current locale.  See \fBenviron\fR(5).
.LP
The \fBwcsrtombs_l()\fR and \fBwcsnrtombs_l()\fR functions behave identically
to \fBwcsrtombs()\fR and \fBwcsnrtombs()\fR respectively, except
that instead of operating in the current locale, they operate in the locale
specified by \fIloc\fR.
.SH RETURN VALUES
.LP
If conversion stops because a code is reached that does not correspond to a
valid character, an encoding error occurs.  In this case, these
functions store the value of the macro \fBEILSEQ\fR in \fBerrno\fR and return
\fB(size_t)\(mi1\fR; the conversion state is undefined.  Otherwise, they return
the number of bytes in the resulting character sequence, not including the
terminating null (if any).
.SH ERRORS
.LP
These functions may fail if:
.sp
.ne 2
.na
\fB\fBEINVAL\fR\fR
.ad
.RS 10n
The \fIps\fR argument points to an object that contains an invalid conversion
state.
.RE

.sp
.ne 2
.na
\fB\fBEILSEQ\fR\fR
.ad
.RS 10n
A wide-character code does not correspond to a valid character.
.RE
.SH ATTRIBUTES
.LP
See \fBattributes\fR(5) for descriptions of the following attributes:
.TS
```

```
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
Interface Stability     See below.
_
MT-Level        See below.
.TE

.LP
The \fBwcsrtombs()\fR and \fBwcsnrtombs()\fR functions are Standard.  The
\fBwcsrtombs_l()\fR and \fBwcsnrtombs_l()\fR functions are Uncommitted.
.LP
If \fIps\fR is a null pointer, these functions should be considered Unsafe
for use in multithreaded applications.  Otherwise, they are MT-Safe.
.SH SEE ALSO
.LP
\fBmbsinit\fR(3C), \fBnewlocale\fR(3C), \fBsetlocale\fR(3C), \fBuselocale\fR(3C)
\fBwcrtomb\fR(3C), \fBattributes\fR(5),
\fBenviron\fR(5), \fBstandards\fR(5)
```

```
**********************************************************
   34015 Thu Jul 17 00:50:42 2014
new/usr/src/man/man3lib/libc.3lib
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
    2499 Thu Jul 17 00:50:42 2014
new/usr/src/man/man4/init.4
manpage lint.
**********************************************************
'\" te
.\" Copyright 2014 Garrett D'Amore
.\" Copyright (c) 2003, 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 INIT 4 "Mar 15, 2014"
.TH init 4 "Mar 15, 2014"
.SH NAME
init \- set default system time zone and locale
.SH SYNOPSIS
.LP
.nf
\fB/etc/default/init\fR
.fi

.SH DESCRIPTION
.sp
.LP
This file sets the time zone environment variable \fBTZ\fR, and the
locale-related environment variables \fBLANG\fR, \fBLC_COLLATE\fR,
\fBLC_CTYPE\fR, \fBLC_MESSAGES\fR, \fBLC_MONETARY\fR, \fBLC_NUMERIC\fR, and
\fBLC_TIME\fR.
.sp
.LP
\fB/etc/TIMEZONE\fR is a symbolic link to \fB/etc/default/init\fR.  This
link exists for compatibility with legacy software, is obsolete, and may
be removed in a future release.
.sp
.LP
The number of environment variables that can be set from
\fB/etc/default/init\fR is limited to 20.
.sp
.LP
The format of the file is:
.sp
.in +2
.nf
\fIVAR\fR\fB=\fR\fIvalue\fR
.fi
.in -2
.sp

.sp
.LP
where \fIVAR\fR is a timezone environment variable and \fIvalue\fR is the value
assigned to the variable. \fIvalue\fR can be enclosed in double quotes (") or
single quotes (\&'). The double or single quotes cannot be part of the value.
.SH SEE ALSO
.sp
.LP
\fBinit\fR(1M), \fBrtc\fR(1M), \fBctime\fR(3C), \fBenviron\fR(5)
.SH NOTES
.sp
.LP
When changing the \fBTZ\fR setting on x86 systems, you must make a
corresponding change to the \fB/etc/rtc_config\fR file to account for the new
timezone setting. This can be accomplished by executing the following commands,
followed by a reboot, to make the changes take effect:
.sp
```



```
.in +2
.nf
# rtc \fB-z\fR \fIzone-name\fR
# rtc \fB-c\fR

.fi
.in -2
.sp

.sp
.LP
where \fIzone-name\fR is the same name as the \fBTZ\fR variable setting.
.sp
.LP
See \fBrtc\fR(1M) for information on the \fBrtc\fR command.
```

```
**********************************************************
   76160 Thu Jul 17 00:50:42 2014
new/usr/src/man/man5/mdoc.5
manpage lint.
**********************************************************
```

```
.\"
.\" Permission to use, copy, modify, and distribute this software for any
.\" purpose with or without fee is hereby granted, provided that the above
.\" copyright notice and this permission notice appear in all copies.
.\"
.\" THE SOFTWARE IS PROVIDED "AS IS" AND THE AUTHOR DISCLAIMS ALL WARRANTIES
.\" WITH REGARD TO THIS SOFTWARE INCLUDING ALL IMPLIED WARRANTIES OF
.\" MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR
.\" ANY SPECIAL, DIRECT, INDIRECT, OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES
.\" WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN
.\" ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF
.\" OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.
.\"
.\"
.\" Copyright (c) 2009, 2010, 2011 Kristaps Dzonsons <kristaps@bsd.lv>
.\" Copyright (c) 2010, 2011 Ingo Schwarze <schwarze@openbsd.org>
.\" Copyright 2012 Nexenta Systems, Inc. All rights reserved.
.\" Copyright 2014 Garrett D'Amore <garrett@dmaore.org>
.\"
.Dd Jul 16, 2014
.Dt MDOC 5
.Os
.Sh NAME
.Nm mdoc
.Nd semantic markup language for formatting manual pages
.Sh DESCRIPTION
The
.Nm mdoc
language supports authoring of manual pages for the
.Xr man 1
utility by allowing semantic annotations of words, phrases,
page sections and complete manual pages.
Such annotations are used by formatting tools to achieve a uniform
presentation across all manuals written in
.Nm ,
and to support hyperlinking if supported by the output medium.
.Pp
This reference document describes the structure of manual pages
and the syntax and usage of the
.Nm
language.
The reference implementation of a parsing and formatting tool is
.Xr mandoc 1 ;
the
.Sx COMPATIBILITY
section describes compatibility with other implementations.
.Pp
In an
.Nm
document, lines beginning with the control character
.Sq \&.
are called
.Dq macro lines .
The first word is the macro name.
It consists of two or three letters.
Most macro names begin with a capital letter.
For a list of available macros, see
.Sx MACRO OVERVIEW .
The words following the macro name are arguments to the macro, optionally
including the names of other, callable macros; see
.Sx MACRO SYNTAX
```



```
for details.
.Pp
Lines not beginning with the control character are called
.Dq text lines .
They provide free-form text to be printed; the formatting of the text
depends on the respective processing context:
.Bd -literal -offset indent
\&.Sh Macro lines change control state.
Text lines are interpreted within the current state.
.Ed
.Pp
Many aspects of the basic syntax of the
.Nm
language are based on the
.Xr roff 5
language; see the
.Em LANGUAGE SYNTAX
and
.Em MACRO SYNTAX
sections in the
.Xr roff 5
manual for details, in particular regarding
comments, escape sequences, whitespace, and quoting.
However, using
.Xr roff 5
requests in
.Nm
documents is discouraged;
.Xr mandoc 1
supports some of them merely for backward compatibility.
.Sh MANUAL STRUCTURE
A well-formed
.Nm
document consists of a document prologue followed by one or more
sections.
.Pp
The prologue, which consists of the
.Sx \&Dd ,
.Sx \&Dt ,
and
.Sx \&Os
macros in that order, is required for every document.
.Pp
The first section (sections are denoted by
.Sx \&Sh )
must be the NAME section, consisting of at least one
.Sx \&Nm
followed by
.Sx \&Nd .
.Pp
Following that, convention dictates specifying at least the
.Em SYNOPSIS
and
.Em DESCRIPTION
sections, although this varies between manual sections.
.Pp
The following is a well-formed skeleton
.Nm
file for a utility
.Qq progname :
.Bd -literal -offset indent
\&.Dd Jan 1, 1970
\&.Dt PROGNAME section
\&.Os
\&.Sh NAME
\&.Nm progname
```

```
\&.Nd one line description
\&.\e\(dq .Sh LIBRARY
\&.\e\(dq For sections 2, 3, & 9 only.
\&.Sh SYNOPSIS
\&.Nm progname
\&.Op Fl options
\&.Ar
\&.Sh DESCRIPTION
The
\&.Nm
utility processes files ...
\&.\e\(dq .Sh IMPLEMENTATION NOTES
\&.\e\(dq .Sh RETURN VALUES
\&.\e\(dq For sections 2, 3, & 9 only.
\&.\e\(dq .Sh ENVIRONMENT
\&.\e\(dq For sections 1, 1M, 5, & 6 only.
\&.\e\(dq .Sh FILES
\&.\e\(dq .Sh EXIT STATUS
\&.\e\(dq For sections 1, 1M, & 6 only.
\&.\e\(dq .Sh EXAMPLES
\&.\e\(dq .Sh DIAGNOSTICS
\&.\e\(dq For sections 1, 1M, 5, 6, & 7 only.
\&.\e\(dq .Sh ERRORS
\&.\e\(dq For sections 2, 3, & 9 only.
\&.\e\(dq .Sh ARCHITECTURE
\&.\e\(dq .Sh CODE SET INDEPENDENCE
\&.\e\(dq For sections 1, 1M, & 3 only.
\&.\e\(dq .Sh INTERFACE STABILITY
\&.\e\(dq .Sh MT-LEVEL
\&.\e\(dq For sections 2 & 3 only.
\&.\e\(dq .Sh SEE ALSO
\&.\e\(dq .Xr foobar 1
\&.\e\(dq .Sh STANDARDS
\&.\e\(dq .Sh HISTORY
\&.\e\(dq .Sh AUTHORS
\&.\e\(dq .Sh CAVEATS
\&.\e\(dq .Sh BUGS
\&.\e\(dq .Sh SECURITY CONSIDERATIONS
\&.\e\(dq Not used in OpenBSD.
.Ed
.Pp
The sections in an
.Nm
document are conventionally ordered as they appear above.
Sections should be composed as follows:
.Bl -ohang -offset Ds
.It Em NAME
The name(s) and a one line description of the documented material.
The syntax for this as follows:
.Bd -literal -offset indent
\&.Nm name0 ,
\&.Nm name1 ,
\&.Nm name2
\&.Nd a one line description
.Ed
.Pp
Multiple
.Sq \&Nm
names should be separated by commas.
.Pp
The
.Sx \&Nm
macro(s) must precede the
.Sx \&Nd
macro.
.Pp
```



```
See
.Sx \&Nm
and
.Sx \&Nd .
.It Em LIBRARY
The name of the library containing the documented material, which is
assumed to be a function in a section 2, 3, or 9 manual.
The syntax for this is as follows:
.Bd -literal -offset indent
\&.Lb libarm
.Ed
.Pp
See
.Sx \&Lb .
.It Em SYNOPSIS
Documents the utility invocation syntax, function call syntax, or device
configuration.
.Pp
For the first, utilities (sections 1, 1M, and 6), this is
generally structured as follows:
.Bd -literal -offset indent
\&.Nm bar
\&.Op Fl v
\&.Op Fl o Ar file
\&.Op Ar
\&.Nm foo
\&.Op Fl v
\&.Op Fl o Ar file
\&.Op Ar
.Ed
.Pp
Commands should be ordered alphabetically.
.Pp
For the second, function calls (sections 2, 3, 9):
.Bd -literal -offset indent
\&.In header.h
\&.Vt extern const char *global;
\&.Ft "char *"
\&.Fn foo "const char *src"
\&.Ft "char *"
\&.Fn bar "const char *src"
.Ed
.Pp
Ordering of
.Sx \&In ,
.Sx \&Vt ,
.Sx \&Fn ,
and
.Sx \&Fo
macros should follow C header-file conventions.
.Pp
And for the third, configurations (section 7):
.Bd -literal -offset indent
\&.Cd \(dqit* at isa? port 0x2e\(dq
\&.Cd \(dqit* at isa? port 0x4e\(dq
.Ed
.Pp
Manuals not in these sections generally don't need a
.Em SYNOPSIS .
.Pp
Some macros are displayed differently in the
.Em SYNOPSIS
section, particularly
.Sx \&Nm ,
.Sx \&Cd ,
.Sx \&Fd ,
```

```
.Sx \&Fn ,
.Sx \&Fo ,
.Sx \&In ,
.Sx \&Vt ,
and
.Sx \&Ft .
All of these macros are output on their own line.
If two such dissimilar macros are pairwise invoked (except for
.Sx \&Ft
before
.Sx \&Fo
or
.Sx \&Fn ) ,
they are separated by a vertical space, unless in the case of
.Sx \&Fo ,
.Sx \&Fn ,
and
.Sx \&Ft ,
which are always separated by vertical space.
.Pp
When text and macros following an
.Sx \&Nm
macro starting an input line span multiple output lines,
all output lines but the first will be indented to align
with the text immediately following the
.Sx \&Nm
macro, up to the next
.Sx \&Nm ,
.Sx \&Sh ,
or
.Sx \&Ss
macro or the end of an enclosing block, whichever comes first.
.It Em DESCRIPTION
This begins with an expansion of the brief, one line description in
.Em NAME :
.Bd -literal -offset indent
The
\&.Nm
utility does this, that, and the other.
.Ed
.Pp
It usually follows with a breakdown of the options (if documenting a
command), such as:
.Bd -literal -offset indent
The arguments are as follows:
\&.Bl \-tag \-width Ds
\&.It Fl v
Print verbose information.
\&.El
.Ed
.Pp
Manuals not documenting a command won't include the above fragment.
.Pp
Since the
.Em DESCRIPTION
section usually contains most of the text of a manual, longer manuals
often use the
.Sx \&Ss
macro to form subsections.
In very long manuals, the
.Em DESCRIPTION
may be split into multiple sections, each started by an
.Sx \&Sh
macro followed by a non-standard section name, and each having
several subsections, like in the present
.Nm
```



```
manual.
.It Em IMPLEMENTATION NOTES
Implementation-specific notes should be kept here.
This is useful when implementing standard functions that may have side
effects or notable algorithmic implications.
.It Em RETURN VALUES
This section documents the
return values of functions in sections 2, 3, and 9.
.Pp
See
.Sx \&Rv .
.It Em ENVIRONMENT
Lists the environment variables used by the utility,
and explains the syntax and semantics of their values.
The
.Xr environ 5
manual provides examples of typical content and formatting.
.Pp
See
.Sx \&Ev .
.It Em FILES
Documents files used.
It's helpful to document both the file name and a short description of how
the file is used (created, modified, etc.).
.Pp
See
.Sx \&Pa .
.It Em EXIT STATUS
This section documents the
command exit status for section 1, 6, and 8 utilities.
Historically, this information was described in
.Em DIAGNOSTICS ,
a practise that is now discouraged.
.Pp
See
.Sx \&Ex .
.It Em EXAMPLES
Example usages.
This often contains snippets of well-formed, well-tested invocations.
Make sure that examples work properly!
.It Em DIAGNOSTICS
Documents error conditions.
This is most useful in section 4 manuals.
Historically, this section was used in place of
.Em EXIT STATUS
for manuals in sections 1, 6, and 8; however, this practise is
discouraged.
.Pp
See
.Sx \&Bl
.Fl diag .
.It Em ERRORS
Documents error handling in sections 2, 3, and 9.
.Pp
See
.Sx \&Er .
.It Em ARCHITECTURE
This section is usually absent, but will be present when the
interface is specific to one or more architectures.
.It Em CODE SET INDEPENDENCE
Indicates whether the interface operates correctly with various different
code sets.  True independent code sets will support not only ASCII and
Extended UNIX Codesets (EUC), but also other multi-byte encodings such as
UTF-8 and GB2312.
.Pp
Generally there will be some limitations that are fairly standard.  See
```

```
.Xr standards 5 for more information about some of these.  Most interfaces
should support at least UTF-8 in addition to ASCII.
.It Em INTERFACE STABILITY
Indicates the level of commitment to the interface. Interfaces can be described
with in the following ways:
.Bl -tag -width Ds
.Bl -tag
.It Nm Standard
Indicates that the interface is defined by one or more standards bodies.
Generally, changes to the interface will be carefully managed to conform
to the relevant standards.  These interfaces are generally the most suitable
for use in portable programs.
.It Nm Committed
Indicates that the interface is intended to be preserved for the long-haul, and
will rarely, if ever change, and never without notification (barring
extraordinary and extenuating circumstances). These interfaces are
preferred over other interfaces with the exeception of
.Nm Standard
interfaces.
.It Nm Uncommitted
Indicates that the interface may change.  Generally, changes to these interfaces
should be infrequent, and some effort will be made to address compatibility
considerations when changing or removing such interfaces.  However, there is
no firm commitment to the preservation of the interface.  Most often this
is applied to interfaces where operational experience with the interface
is still limited and some need to change may be anticipated.
.Pp
Consumers should expect to revalidate any
.Nm Uncommitted
interfaces when crossing release boundaries.  Products intended for
use on many releases or intended to support compatibility with future
releases should avoid these interfaces.
.It Nm Volatile
The interface can change at any time for any reason. Often this relates to
interfaces that are part of external software components that are still evolving
rapidly.  Consumers should not expect that the interface (either binary or
source level) will be unchanged from one release to the next.
.It Nm Not-an-Interface
Describes something that is specifically not intended for programmatic
consumption.  For example, specific human-readable output, or the layout
of graphical items on a user interface, may be described this way.  Generally
programmatic alternatives to these will be available, and should be used
when programmatic consumption is needed.
.It Nm Private
This is an internal interface.  Generally these interfaces should only be
used within the project, and should not be used by other programs or modules.
The interface can and will change without notice as the project needs, at
any time.
.Pp
Most often, Private interfaces will lack any documentation whatsoever, and
generally any undocumented interface can be assumed to be Private.
.It Nm Obsolete
The interface is not intended for use in new projects or programs, and may
be removed at a future date.  The
.Nm Obsolete
word is a modifier that can
be applied to other commitment levels. For example an
.Nm Obsolete Committed
interface is unlikely to be removed or changed, but nonetheless new use
is discouraged (perhaps a better newer alternative is present).
.El
.It Em MT-LEVEL
This section describes considerations for the interface when used within
programs that use multiple threads.  More discussion of these considerations
is made in the MT-Level section of
.Xr attributes 5 .
```



```
The interface can be described in the following ways.
.Bl -tag -width Ds
.Bl -tag
.It Nm Safe
Indicates the interface is safe for use within multiple threads.  There
may be additional caveats that apply, in which case those will be
described.  Note that some interfaces have semantics which may affect
other threads, but these should be an intrinsic part of the interface
rather than an unexpected side effect.  For example, closing a file in
one thread will cause that file to be closed in all threads.
.It Nm Unsafe
Indicates the interface is unsuitable for concurrent use within multiple
threads.  A threaded application may still make use of the interface, but
will be required to provide external synchronization means to ensure that
only a single thread calls the interface at a time.
.It Nm MT-Safe
Indicates that the interface is not only safe for concurrent use, but is
designed for such use.  For example, a
.Nm Safe
interface may make use of a global lock to provide safety, but at reduced
internal concurrency, whereas an
.Nm MT-Safe
interface will be designed to be efficient even when used concurrently.
.It Nm Async-Signal-Safe
Indicates that the library is safe for use within a signal handler.  An
.Nm MT-Safe
interface can be made
.Nm Async-Signal-Safe
by ensuring that it blocks signals when acquiring locks.
.It Nm Safe with Exections
As for
.Nm Safe
but with specific exceptions noted.
.It Nm MT-Safe with Exections
As for
.Nm MT-Safe
but with specific exceptions noted.
.El
.It Em SEE ALSO
References other manuals with related topics.
This section should exist for most manuals.
Cross-references should conventionally be ordered first by section, then
alphabetically.
.Pp
References to other documentation concerning the topic of the manual page,
for example authoritative books or journal articles, may also be
provided in this section.
.Pp
See
.Sx \&Rs
and
.Sx \&Xr .
.It Em STANDARDS
References any standards implemented or used.
If not adhering to any standards, the
.Em HISTORY
section should be used instead.
.Pp
See
.Sx \&St .
.It Em HISTORY
A brief history of the subject, including where it was first implemented,
and when it was ported to or reimplemented for the operating system at hand.
.It Em AUTHORS
Credits to the person or persons who wrote the code and/or documentation.
Authors should generally be noted by both name and email address.
```

```
.Pp
See
.Sx \&An .
.It Em CAVEATS
Common misuses and misunderstandings should be explained
in this section.
.It Em BUGS
Known bugs, limitations, and work-arounds should be described
in this section.
.It Em SECURITY CONSIDERATIONS
Documents any security precautions that operators should consider.
.El
.Sh MACRO OVERVIEW
This overview is sorted such that macros of similar purpose are listed
together, to help find the best macro for any given purpose.
Deprecated macros are not included in the overview, but can be found below
in the alphabetical
.Sx MACRO REFERENCE .
.Ss Document preamble and NAME section macros
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Dd Ta document date: Ar month day , year
.It Sx \&Dt Ta document title: Ar TITLE SECTION Op Ar volume | arch
.It Sx \&Os Ta operating system version: Op Ar system Op Ar version
.It Sx \&Nm Ta document name (one argument)
.It Sx \&Nd Ta document description (one line)
.El
.Ss Sections and cross references
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Sh Ta section header (one line)
.It Sx \&Ss Ta subsection header (one line)
.It Sx \&Sx Ta internal cross reference to a section or subsection
.It Sx \&Xr Ta cross reference to another manual page: Ar name section
.It Sx \&Pp , \&Lp Ta start a text paragraph (no arguments)
.El
.Ss Displays and lists
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Bd , \&Ed Ta display block:
.Fl Ar type
.Op Fl offset Ar width
.Op Fl compact
.It Sx \&D1 Ta indented display (one line)
.It Sx \&Dl Ta indented literal display (one line)
.It Sx \&Bl , \&El Ta list block:
.Fl Ar type
.Op Fl width Ar val
.Op Fl offset Ar val
.Op Fl compact
.It Sx \&It Ta list item (syntax depends on Fl Ar type )
.It Sx \&Ta Ta table cell separator in Sx \&Bl Fl column No lists
.It Sx \&Rs , \&%* , \&Re Ta bibliographic block (references)
.El
.Ss Spacing control
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Pf Ta prefix, no following horizontal space (one argument)
.It Sx \&Ns Ta roman font, no preceding horizontal space (no arguments)
.It Sx \&Ap Ta apostrophe without surrounding whitespace (no arguments)
.It Sx \&Sm Ta switch horizontal spacing mode: Cm on | off
.It Sx \&Bk , \&Ek Ta keep block: Fl words
.It Sx \&br Ta force output line break in text mode (no arguments)
.It Sx \&sp Ta force vertical space: Op Ar height
.El
.Ss Semantic markup for command line utilities:
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Nm Ta start a SYNOPSIS block with the name of a utility
.It Sx \&Fl Ta command line options (flags) (>=0 arguments)
.It Sx \&Cm Ta command modifier (>0 arguments)
```



```
.It Sx \&Ar Ta command arguments (>=0 arguments)
.It Sx \&Op , \&Oo , \&Oc Ta optional syntax elements (enclosure)
.It Sx \&Ic Ta internal or interactive command (>0 arguments)
.It Sx \&Ev Ta environmental variable (>0 arguments)
.It Sx \&Pa Ta file system path (>=0 arguments)
.El
.Ss Semantic markup for function libraries:
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Lb Ta function library (one argument)
.It Sx \&In Ta include file (one argument)
.It Sx \&Ft Ta function type (>0 arguments)
.It Sx \&Fo , \&Fc Ta function block: Ar funcname
.It Sx \&Fn Ta function name:
.Op Ar functype
.Ar funcname
.Oo
.Op Ar argtype
.Ar argname
.Oc
.It Sx \&Fa Ta function argument (>0 arguments)
.It Sx \&Vt Ta variable type (>0 arguments)
.It Sx \&Va Ta variable name (>0 arguments)
.It Sx \&Dv Ta defined variable or preprocessor constant (>0 arguments)
.It Sx \&Er Ta error constant (>0 arguments)
.It Sx \&Ev Ta environmental variable (>0 arguments)
.El
.Ss Various semantic markup:
.Bl -column "Brq, Bro, Brc" description
.It Sx \&An Ta author name (>0 arguments)
.It Sx \&Lk Ta hyperlink: Ar uri Op Ar name
.It Sx \&Mt Ta Do mailto Dc hyperlink: Ar address
.It Sx \&Cd Ta kernel configuration declaration (>0 arguments)
.It Sx \&Ad Ta memory address (>0 arguments)
.It Sx \&Ms Ta mathematical symbol (>0 arguments)
.It Sx \&Tn Ta tradename (>0 arguments)
.El
.Ss Physical markup
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Em Ta italic font or underline (emphasis) (>0 arguments)
.It Sx \&Sy Ta boldface font (symbolic) (>0 arguments)
.It Sx \&Li Ta typewriter font (literal) (>0 arguments)
.It Sx \&No Ta return to roman font (normal) (no arguments)
.It Sx \&Bf , \&Ef Ta font block:
.Op Fl Ar type | Cm \&Em | \&Li | \&Sy
.El
.Ss Physical enclosures
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Dq , \&Do , \&Dc Ta enclose in typographic double quotes: Dq text
.It Sx \&Qq , \&Qo , \&Qc Ta enclose in typewriter double quotes: Qq text
.It Sx \&Sq , \&So , \&Sc Ta enclose in single quotes: Sq text
.It Sx \&Ql Ta single-quoted literal text: Ql text
.It Sx \&Pq , \&Po , \&Pc Ta enclose in parentheses: Pq text
.It Sx \&Bq , \&Bo , \&Bc Ta enclose in square brackets: Bq text
.It Sx \&Brq , \&Bro , \&Brc Ta enclose in curly braces: Brq text
.It Sx \&Aq , \&Ao , \&Ac Ta enclose in angle brackets: Aq text
.It Sx \&Eo , \&Ec Ta generic enclosure
.El
.Ss Text production
.Bl -column "Brq, Bro, Brc" description
.It Sx \&Ex Fl std Ta standard command exit values: Op Ar utility ...
.It Sx \&Rv Fl std Ta standard function return values: Op Ar function ...
.It Sx \&St Ta reference to a standards document (one argument)
.It Sx \&Ux Ta Ux
.It Sx \&At Ta At
.It Sx \&Bx Ta Bx
.It Sx \&Bsx Ta Bsx
```

```
.It Sx \&Nx Ta Nx
.It Sx \&Fx Ta Fx
.It Sx \&Ox Ta Ox
.It Sx \&Dx Ta Dx
.El
.Sh MACRO REFERENCE
This section is a canonical reference of all macros, arranged
alphabetically.
For the scoping of individual macros, see
.Sx MACRO SYNTAX .
.Ss \&%A
Author name of an
.Sx \&Rs
block.
Multiple authors should each be accorded their own
.Sx \%%A
line.
Author names should be ordered with full or abbreviated forename(s)
first, then full surname.
.Ss \&%B
Book title of an
.Sx \&Rs
block.
This macro may also be used in a non-bibliographic context when
referring to book titles.
.Ss \&%C
Publication city or location of an
.Sx \&Rs
block.
.Ss \&%D
Publication date of an
.Sx \&Rs
block.
Recommended formats of arguments are
.Ar month day , year
or just
.Ar year .
.Ss \&%I
Publisher or issuer name of an
.Sx \&Rs
block.
.Ss \&%J
Journal name of an
.Sx \&Rs
block.
.Ss \&%N
Issue number (usually for journals) of an
.Sx \&Rs
block.
.Ss \&%O
Optional information of an
.Sx \&Rs
block.
.Ss \&%P
Book or journal page number of an
.Sx \&Rs
block.
.Ss \&%Q
Institutional author (school, government, etc.) of an
.Sx \&Rs
block.
Multiple institutional authors should each be accorded their own
.Sx \&%Q
line.
.Ss \&%R
Technical report name of an
```



```
.Sx \&Rs
block.
.Ss \&%T
Article title of an
.Sx \&Rs
block.
This macro may also be used in a non-bibliographical context when
referring to article titles.
.Ss \&%U
URI of reference document.
.Ss \&%V
Volume number of an
.Sx \&Rs
block.
.Ss \&Ac
Close an
.Sx \&Ao
block.
Does not have any tail arguments.
.Ss \&Ad
Memory address.
Do not use this for postal addresses.
.Pp
Examples:
.Dl \&.Ad [0,$]
.Dl \&.Ad 0x00000000
.Ss \&An
Author name.
Can be used both for the authors of the program, function, or driver
documented in the manual, or for the authors of the manual itself.
Requires either the name of an author or one of the following arguments:
.Pp
.Bl -tag -width "-nosplitX" -offset indent -compact
.It Fl split
Start a new output line before each subsequent invocation of
.Sx \&An .
.It Fl nosplit
The opposite of
.Fl split .
.El
.Pp
The default is
.Fl nosplit .
The effect of selecting either of the
.Fl split
modes ends at the beginning of the
.Em AUTHORS
section.
In the
.Em AUTHORS
section, the default is
.Fl nosplit
for the first author listing and
.Fl split
for all other author listings.
.Pp
Examples:
.Dl \&.An -nosplit
.Dl \&.An Kristaps Dzonsons \&Aq kristaps@bsd.lv
.Ss \&Ao
Begin a block enclosed by angle brackets.
Does not have any head arguments.
.Pp
Examples:
.Dl \&.Fl -key= \&Ns \&Ao \&Ar val \&Ac
.Pp
```

```
786 See also
787 .Sx \&Aq .
788 .Ss \&Ap
789 Inserts an apostrophe without any surrounding whitespace.
790 This is generally used as a grammatical device when referring to the verb
791 form of a function.
792 .Pp
793 Examples:
794 .Dl \&.Fn execve \&Ap d
795 .Ss \&Aq
796 Encloses its arguments in angle brackets.
797 .Pp
798 Examples:
799 .Dl \&.Fl -key= \&Ns \&Aq \&Ar val
800 .Pp
801 .Em Remarks :
802 this macro is often abused for rendering URIs, which should instead use
803 .Sx \&Lk
804 or
805 .Sx \&Mt ,
806 or to note pre-processor
807 .Dq Li #include
808 statements, which should use
809 .Sx \&In .
810 .Pp
811 See also
812 .Sx \&Ao .
813 .Ss \&Ar
814 Command arguments.
815 If an argument is not provided, the string
816 .Dq file ...\&
817 is used as a default.
818 .Pp
819 Examples:
820 .Dl ".Fl o Ar file"
821 .Dl ".Ar"
822 .Dl ".Ar arg1 , arg2 ."
823 .Pp
824 The arguments to the
825 .Sx \&Ar
826 macro are names and placeholders for command arguments;
827 for fixed strings to be passed verbatim as arguments, use
828 .Sx \&Fl
829 or
830 .Sx \&Cm .
831 .Ss \&At
832 Formats an AT&T version.
833 Accepts one optional argument:
834 .Pp
835 .Bl -tag -width "v[1-7] | 32vX" -offset indent -compact
836 .It Cm v[1-7] | 32v
837 A version of
838 .At .
839 .It Cm III
840 .At III .
841 .It Cm V[.[1-4]]?
842 A version of
843 .At V .
844 .El
845 .Pp
846 Note that these arguments do not begin with a hyphen.
847 .Pp
848 Examples:
849 .Dl \&.At
850 .Dl \&.At III
851 .Dl \&.At V.1
```



```
852 .Pp
853 See also
854 .Sx \&Bsx ,
855 .Sx \&Bx ,
856 .Sx \&Dx ,
857 .Sx \&Fx ,
858 .Sx \&Nx ,
859 .Sx \&Ox ,
860 and
861 .Sx \&Ux .
862 .Ss \&Bc
863 Close a
864 .Sx \&Bo
865 block.
866 Does not have any tail arguments.
867 .Ss \&Bd
868 Begin a display block.
869 Its syntax is as follows:
870 .Bd -ragged -offset indent
871 .Pf \. Sx \&Bd
872 .Fl Ns Ar type
873 .Op Fl offset Ar width
874 .Op Fl compact
875 .Ed
876 .Pp
877 Display blocks are used to select a different indentation and
878 justification than the one used by the surrounding text.
879 They may contain both macro lines and text lines.
880 By default, a display block is preceded by a vertical space.
881 .Pp
882 The
883 .Ar type
884 must be one of the following:
885 .Bl -tag -width 13n -offset indent
886 .It Fl centered
887 Produce one output line from each input line, and centre-justify each line.
888 Using this display type is not recommended; many
889 .Nm
890 implementations render it poorly.
891 .It Fl filled
892 Change the positions of line breaks to fill each line, and left- and
893 right-justify the resulting block.
894 .It Fl literal
895 Produce one output line from each input line,
896 and do not justify the block at all.
897 Preserve white space as it appears in the input.
898 Always use a constant-width font.
899 Use this for displaying source code.
900 .It Fl ragged
901 Change the positions of line breaks to fill each line, and left-justify
902 the resulting block.
903 .It Fl unfilled
904 The same as
905 .Fl literal ,
906 but using the same font as for normal text, which is a variable width font
907 if supported by the output device.
908 .El
909 .Pp
910 The
911 .Ar type
912 must be provided first.
913 Additional arguments may follow:
914 .Bl -tag -width 13n -offset indent
915 .It Fl offset Ar width
916 Indent the display by the
917 .Ar width ,
```

```
 918 which may be one of the following:
 919 .Bl -item
 920 .It
 921 One of the pre-defined strings
 922 .Cm indent ,
 923 the width of a standard indentation (six constant width characters);
 924 .Cm indent-two ,
 925 twice
 926 .Cm indent ;
 927 .Cm left ,
 928 which has no effect;
 929 .Cm right ,
 930 which justifies to the right margin; or
 931 .Cm center ,
 932 which aligns around an imagined centre axis.
 933 .It
 934 A macro invocation, which selects a predefined width
 935 associated with that macro.
 936 The most popular is the imaginary macro
 937 .Ar \&Ds ,
 938 which resolves to
 939 .Sy 6n .
 940 .It
 941 A width using the syntax described in
 942 .Sx Scaling Widths .
 943 .It
 944 An arbitrary string, which indents by the length of this string.
 945 .El
 946 .Pp
 947 When the argument is missing,
 948 .Fl offset
 949 is ignored.
 950 .It Fl compact
 951 Do not assert vertical space before the display.
 952 .El
 953 .Pp
 954 Examples:
 955 .Bd -literal -offset indent
 956 \&.Bd \-literal \-offset indent \-compact
 957    Hello       world.
 958 \&.Ed
 959 .Ed
 960 .Pp
 961 See also
 962 .Sx \&D1
 963 and
 964 .Sx \&Dl .
 965 .Ss \&Bf
 966 Change the font mode for a scoped block of text.
 967 Its syntax is as follows:
 968 .Bd -ragged -offset indent
 969 .Pf \. Sx \&Bf
 970 .Oo
 971 .Fl emphasis | literal | symbolic |
 972 .Cm \&Em | \&Li | \&Sy
 973 .Oc
 974 .Ed
 975 .Pp
 976 The
 977 .Fl emphasis
 978 and
 979 .Cm \&Em
 980 argument are equivalent, as are
 981 .Fl symbolic
 982 and
 983 .Cm \&Sy ,
```



```
 984 and
 985 .Fl literal
 986 and
 987 .Cm \&Li .
 988 Without an argument, this macro does nothing.
 989 The font mode continues until broken by a new font mode in a nested
 990 scope or
 991 .Sx \&Ef
 992 is encountered.
 993 .Pp
 994 See also
 995 .Sx \&Li ,
 996 .Sx \&Ef ,
 997 .Sx \&Em ,
 998 and
 999 .Sx \&Sy .
1000 .Ss \&Bk
1001 For each macro, keep its output together on the same output line,
1002 until the end of the macro or the end of the input line is reached,
1003 whichever comes first.
1004 Line breaks in text lines are unaffected.
1005 The syntax is as follows:
1006 .Pp
1007 .D1 Pf \. Sx \&Bk Fl words
1008 .Pp
1009 The
1010 .Fl words
1011 argument is required; additional arguments are ignored.
1012 .Pp
1013 The following example will not break within each
1014 .Sx \&Op
1015 macro line:
1016 .Bd -literal -offset indent
1017 \&.Bk \-words
1018 \&.Op Fl f Ar flags
1019 \&.Op Fl o Ar output
1020 \&.Ek
1021 .Ed
1022 .Pp
1023 Be careful in using over-long lines within a keep block!
1024 Doing so will clobber the right margin.
1025 .Ss \&Bl
1026 Begin a list.
1027 Lists consist of items specified using the
1028 .Sx \&It
1029 macro, containing a head or a body or both.
1030 The list syntax is as follows:
1031 .Bd -ragged -offset indent
1032 .Pf \. Sx \&Bl
1033 .Fl Ns Ar type
1034 .Op Fl width Ar val
1035 .Op Fl offset Ar val
1036 .Op Fl compact
1037 .Op HEAD ...
1038 .Ed
1039 .Pp
1040 The list
1041 .Ar type
1042 is mandatory and must be specified first.
1043 The
1044 .Fl width
1045 and
1046 .Fl offset
1047 arguments accept
1048 .Sx Scaling Widths
1049 or use the length of the given string.
```

```
1050 The
1051 .Fl offset
1052 is a global indentation for the whole list, affecting both item heads
1053 and bodies.
1054 For those list types supporting it, the
1055 .Fl width
1056 argument requests an additional indentation of item bodies,
1057 to be added to the
1058 .Fl offset .
1059 Unless the
1060 .Fl compact
1061 argument is specified, list entries are separated by vertical space.
1062 .Pp
1063 A list must specify one of the following list types:
1064 .Bl -tag -width 12n -offset indent
1065 .It Fl bullet
1066 No item heads can be specified, but a bullet will be printed at the head
1067 of each item.
1068 Item bodies start on the same output line as the bullet
1069 and are indented according to the
1070 .Fl width
1071 argument.
1072 .It Fl column
1073 A columnated list.
1074 The
1075 .Fl width
1076 argument has no effect; instead, each argument specifies the width
1077 of one column, using either the
1078 .Sx Scaling Widths
1079 syntax or the string length of the argument.
1080 If the first line of the body of a
1081 .Fl column
1082 list is not an
1083 .Sx \&It
1084 macro line,
1085 .Sx \&It
1086 contexts spanning one input line each are implied until an
1087 .Sx \&It
1088 macro line is encountered, at which point items start being interpreted as
1089 described in the
1090 .Sx \&It
1091 documentation.
1092 .It Fl dash
1093 Like
1094 .Fl bullet ,
1095 except that dashes are used in place of bullets.
1096 .It Fl diag
1097 Like
1098 .Fl inset ,
1099 except that item heads are not parsed for macro invocations.
1100 Most often used in the
1101 .Em DIAGNOSTICS
1102 section with error constants in the item heads.
1103 .It Fl enum
1104 A numbered list.
1105 No item heads can be specified.
1106 Formatted like
1107 .Fl bullet ,
1108 except that cardinal numbers are used in place of bullets,
1109 starting at 1.
1110 .It Fl hang
1111 Like
1112 .Fl tag ,
1113 except that the first lines of item bodies are not indented, but follow
1114 the item heads like in
1115 .Fl inset
```



```
1116 lists.
1117 .It Fl hyphen
1118 Synonym for
1119 .Fl dash .
1120 .It Fl inset
1121 Item bodies follow items heads on the same line, using normal inter-word
1122 spacing.
1123 Bodies are not indented, and the
1124 .Fl width
1125 argument is ignored.
1126 .It Fl item
1127 No item heads can be specified, and none are printed.
1128 Bodies are not indented, and the
1129 .Fl width
1130 argument is ignored.
1131 .It Fl ohang
1132 Item bodies start on the line following item heads and are not indented.
1133 The
1134 .Fl width
1135 argument is ignored.
1136 .It Fl tag
1137 Item bodies are indented according to the
1138 .Fl width
1139 argument.
1140 When an item head fits inside the indentation, the item body follows
1141 this head on the same output line.
1142 Otherwise, the body starts on the output line following the head.
1143 .El
1144 .Pp
1145 Lists may be nested within lists and displays.
1146 Nesting of
1147 .Fl column
1148 and
1149 .Fl enum
1150 lists may not be portable.
1151 .Pp
1152 See also
1153 .Sx \&El
1154 and
1155 .Sx \&It .
1156 .Ss \&Bo
1157 Begin a block enclosed by square brackets.
1158 Does not have any head arguments.
1159 .Pp
1160 Examples:
1161 .Bd -literal -offset indent -compact
1162 \&.Bo 1 ,
1163 \&.Dv BUFSIZ \&Bc
1164 .Ed
1165 .Pp
1166 See also
1167 .Sx \&Bq .
1168 .Ss \&Bq
1169 Encloses its arguments in square brackets.
1170 .Pp
1171 Examples:
1172 .Dl \&.Bq 1 , \&Dv BUFSIZ
1173 .Pp
1174 .Em Remarks :
1175 this macro is sometimes abused to emulate optional arguments for
1176 commands; the correct macros to use for this purpose are
1177 .Sx \&Op ,
1178 .Sx \&Oo ,
1179 and
1180 .Sx \&Oc .
1181 .Pp
```

```
1182 See also
1183 .Sx \&Bo .
1184 .Ss \&Brc
1185 Close a
1186 .Sx \&Bro
1187 block.
1188 Does not have any tail arguments.
1189 .Ss \&Bro
1190 Begin a block enclosed by curly braces.
1191 Does not have any head arguments.
1192 .Pp
1193 Examples:
1194 .Bd -literal -offset indent -compact
1195 \&.Bro 1 , ... ,
1196 \&.Va n \&Brc
1197 .Ed
1198 .Pp
1199 See also
1200 .Sx \&Brq .
1201 .Ss \&Brq
1202 Encloses its arguments in curly braces.
1203 .Pp
1204 Examples:
1205 .Dl \&.Brq 1 , ... , \&Va n
1206 .Pp
1207 See also
1208 .Sx \&Bro .
1209 .Ss \&Bsx
1210 Format the BSD/OS version provided as an argument, or a default value if
1211 no argument is provided.
1212 .Pp
1213 Examples:
1214 .Dl \&.Bsx 1.0
1215 .Dl \&.Bsx
1216 .Pp
1217 See also
1218 .Sx \&At ,
1219 .Sx \&Bx ,
1220 .Sx \&Dx ,
1221 .Sx \&Fx ,
1222 .Sx \&Nx ,
1223 .Sx \&Ox ,
1224 and
1225 .Sx \&Ux .
1226 .Ss \&Bt
1227 Prints
1228 .Dq is currently in beta test.
1229 .Ss \&Bx
1230 Format the BSD version provided as an argument, or a default value if no
1231 argument is provided.
1232 .Pp
1233 Examples:
1234 .Dl \&.Bx 4.3 Tahoe
1235 .Dl \&.Bx 4.4
1236 .Dl \&.Bx
1237 .Pp
1238 See also
1239 .Sx \&At ,
1240 .Sx \&Bsx ,
1241 .Sx \&Dx ,
1242 .Sx \&Fx ,
1243 .Sx \&Nx ,
1244 .Sx \&Ox ,
1245 and
1246 .Sx \&Ux .
1247 .Ss \&Cd
```



```
1248 Kernel configuration declaration.
1249 This denotes strings accepted by
1250 .Xr config 8 .
1251 It is most often used in section 4 manual pages.
1252 .Pp
1253 Examples:
1254 .Dl \&.Cd device le0 at scode?
1255 .Pp
1256 .Em Remarks :
1257 this macro is commonly abused by using quoted literals to retain
1258 whitespace and align consecutive
1259 .Sx \&Cd
1260 declarations.
1261 This practise is discouraged.
1262 .Ss \&Cm
1263 Command modifiers.
1264 Typically used for fixed strings passed as arguments, unless
1265 .Sx \&Fl
1266 is more appropriate.
1267 Also useful when specifying configuration options or keys.
1268 .Pp
1269 Examples:
1270 .Dl ".Nm mt Fl f Ar device Cm rewind"
1271 .Dl ".Nm ps Fl o Cm pid , Ns Cm command"
1272 .Dl ".Nm dd Cm if= Ns Ar file1 Cm of= Ns Ar file2"
1273 .Dl ".Cm IdentityFile Pa ~/.ssh/id_rsa"
1274 .Dl ".Cm LogLevel Dv DEBUG"
1275 .Ss \&D1
1276 One-line indented display.
1277 This is formatted by the default rules and is useful for simple indented
1278 statements.
1279 It is followed by a newline.
1280 .Pp
1281 Examples:
1282 .Dl \&.D1 \&Fl abcdefgh
1283 .Pp
1284 See also
1285 .Sx \&Bd
1286 and
1287 .Sx \&Dl .
1288 .Ss \&Db
1289 Switch debugging mode.
1290 Its syntax is as follows:
1291 .Pp
1292 .D1 Pf \. Sx \&Db Cm on | off
1293 .Pp
1294 This macro is ignored by
1295 .Xr mandoc 1 .
1296 .Ss \&Dc
1297 Close a
1298 .Sx \&Do
1299 block.
1300 Does not have any tail arguments.
1301 .Ss \&Dd
1302 Document date.
1303 This is the mandatory first macro of any
1304 .Nm
1305 manual.
1306 Its syntax is as follows:
1307 .Pp
1308 .D1 Pf \. Sx \&Dd Ar month day , year
1309 .Pp
1310 The
1311 .Ar month
1312 is the full English month name, the
1313 .Ar day
```

```
1314 is an optionally zero-padded numeral, and the
1315 .Ar year
1316 is the full four-digit year.
1317 .Pp
1318 Other arguments are not portable; the
1319 .Xr mandoc 1
1320 utility handles them as follows:
1321 .Bl -dash -offset 3n -compact
1322 .It
1323 To have the date automatically filled in by the
1324 .Ox
1325 version of
1326 .Xr cvs 1 ,
1327 the special string
1328 .Dq $\&Mdocdate$
1329 can be given as an argument.
1330 .It
1331 A few alternative date formats are accepted as well
1332 and converted to the standard form.
1333 .It
1334 If a date string cannot be parsed, it is used verbatim.
1335 .It
1336 If no date string is given, the current date is used.
1337 .El
1338 .Pp
1339 Examples:
1340 .Dl \&.Dd $\&Mdocdate$
1341 .Dl \&.Dd $\&Mdocdate: July 21 2007$
1342 .Dl \&.Dd July 21, 2007
1343 .Pp
1344 See also
1345 .Sx \&Dt
1346 and
1347 .Sx \&Os .
1348 .Ss \&Dl
1349 One-line intended display.
1350 This is formatted as literal text and is useful for commands and
1351 invocations.
1352 It is followed by a newline.
1353 .Pp
1354 Examples:
1355 .Dl \&.Dl % mandoc mdoc.5 \e(ba less
1356 .Pp
1357 See also
1358 .Sx \&Bd
1359 and
1360 .Sx \&D1 .
1361 .Ss \&Do
1362 Begin a block enclosed by double quotes.
1363 Does not have any head arguments.
1364 .Pp
1365 Examples:
1366 .Bd -literal -offset indent -compact
1367 \&.Do
1368 April is the cruellest month
1369 \&.Dc
1370 \e(em T.S. Eliot
1371 .Ed
1372 .Pp
1373 See also
1374 .Sx \&Dq .
1375 .Ss \&Dq
1376 Encloses its arguments in
1377 .Dq typographic
1378 double-quotes.
1379 .Pp
```



```
1380 Examples:
1381 .Bd -literal -offset indent -compact
1382 \&.Dq April is the cruellest month
1383 \e(em T.S. Eliot
1384 .Ed
1385 .Pp
1386 See also
1387 .Sx \&Qq ,
1388 .Sx \&Sq ,
1389 and
1390 .Sx \&Do .
1391 .Ss \&Dt
1392 Document title.
1393 This is the mandatory second macro of any
1394 .Nm
1395 file.
1396 Its syntax is as follows:
1397 .Bd -ragged -offset indent
1398 .Pf \. Sx \&Dt
1399 .Oo
1400 .Ar title
1401 .Oo
1402 .Ar section
1403 .Op Ar volume
1404 .Op Ar arch
1405 .Oc
1406 .Oc
1407 .Ed
1408 .Pp
1409 Its arguments are as follows:
1410 .Bl -tag -width Ds -offset Ds
1411 .It Ar title
1412 The document’s title (name), defaulting to
1413 .Dq UNKNOWN
1414 if unspecified.
1415 It should be capitalised.
1416 .It Ar section
1417 The manual section. It should correspond to the manual’s filename suffix
1418 and defaults to
1419 .Dq 1
1420 if unspecified.
1421 .It Ar volume
1422 This overrides the volume inferred from
1423 .Ar section .
1424 This field is optional.
1425 .It Ar arch
1426 This specifies the machine architecture a manual page applies to,
1427 where relevant.
1428 .El
1429 .Ss \&Dv
1430 Defined variables such as preprocessor constants, constant symbols,
1431 enumeration values, and so on.
1432 .Pp
1433 Examples:
1434 .Dl \&.Dv NULL
1435 .Dl \&.Dv BUFSIZ
1436 .Dl \&.Dv STDOUT_FILENO
1437 .Pp
1438 See also
1439 .Sx \&Er
1440 and
1441 .Sx \&Ev
1442 for special-purpose constants and
1443 .Sx \&Va
1444 for variable symbols.
1445 .Ss \&Dx
```

```
1446 Format the DragonFly BSD version provided as an argument, or a default
1447 value if no argument is provided.
1448 .Pp
1449 Examples:
1450 .Dl \&.Dx 2.4.1
1451 .Dl \&.Dx
1452 .Pp
1453 See also
1454 .Sx \&At ,
1455 .Sx \&Bsx ,
1456 .Sx \&Bx ,
1457 .Sx \&Fx ,
1458 .Sx \&Nx ,
1459 .Sx \&Ox ,
1460 and
1461 .Sx \&Ux .
1462 .Ss \&Ec
1463 Close a scope started by
1464 .Sx \&Eo .
1465 Its syntax is as follows:
1466 .Pp
1467 .D1 Pf \. Sx \&Ec Op Ar TERM
1468 .Pp
1469 The
1470 .Ar TERM
1471 argument is used as the enclosure tail, for example, specifying \e(rq
1472 will emulate
1473 .Sx \&Dc .
1474 .Ss \&Ed
1475 End a display context started by
1476 .Sx \&Bd .
1477 .Ss \&Ef
1478 End a font mode context started by
1479 .Sx \&Bf .
1480 .Ss \&Ek
1481 End a keep context started by
1482 .Sx \&Bk .
1483 .Ss \&El
1484 End a list context started by
1485 .Sx \&Bl .
1486 .Pp
1487 See also
1488 .Sx \&Bl
1489 and
1490 .Sx \&It .
1491 .Ss \&Em
1492 Denotes text that should be
1493 .Em emphasised .
1494 Note that this is a presentation term and should not be used for
1495 stylistically decorating technical terms.
1496 Depending on the output device, this is usually represented
1497 using an italic font or underlined characters.
1498 .Pp
1499 Examples:
1500 .Dl \&.Em Warnings!
1501 .Dl \&.Em Remarks :
1502 .Pp
1503 See also
1504 .Sx \&Bf ,
1505 .Sx \&Li ,
1506 .Sx \&No ,
1507 and
1508 .Sx \&Sy .
1509 .Ss \&En
1510 This macro is obsolete and not implemented in
1511 .Xr mandoc 1 .
```



```
1512 .Ss \&Eo
1513 An arbitrary enclosure.
1514 Its syntax is as follows:
1515 .Pp
1516 .D1 Pf \. Sx \&Eo Op Ar TERM
1517 .Pp
1518 The
1519 .Ar TERM
1520 argument is used as the enclosure head, for example, specifying \e(lq
1521 will emulate
1522 .Sx \&Do .
1523 .Ss \&Er
1524 Error constants for definitions of the
1525 .Va errno
1526 libc global variable.
1527 This is most often used in section 2 and 3 manual pages.
1528 .Pp
1529 Examples:
1530 .Dl \&.Er EPERM
1531 .Dl \&.Er ENOENT
1532 .Pp
1533 See also
1534 .Sx \&Dv
1535 for general constants.
1536 .Ss \&Es
1537 This macro is obsolete and not implemented.
1538 .Ss \&Ev
1539 Environmental variables such as those specified in
1540 .Xr environ 5 .
1541 .Pp
1542 Examples:
1543 .Dl \&.Ev DISPLAY
1544 .Dl \&.Ev PATH
1545 .Pp
1546 See also
1547 .Sx \&Dv
1548 for general constants.
1549 .Ss \&Ex
1550 Insert a standard sentence regarding command exit values of 0 on success
1551 and >0 on failure.
1552 This is most often used in section 1, 6, and 8 manual pages.
1553 Its syntax is as follows:
1554 .Pp
1555 .D1 Pf \. Sx \&Ex Fl std Op Ar utility ...
1556 .Pp
1557 If
1558 .Ar utility
1559 is not specified, the document’s name set by
1560 .Sx \&Nm
1561 is used.
1562 Multiple
1563 .Ar utility
1564 arguments are treated as separate utilities.
1565 .Pp
1566 See also
1567 .Sx \&Rv .
1568 .Ss \&Fa
1569 Function argument.
1570 Its syntax is as follows:
1571 .Bd -ragged -offset indent
1572 .Pf \. Sx \&Fa
1573 .Op Cm argtype
1574 .Cm argname
1575 .Ed
1576 .Pp
1577 This may be invoked for names with or without the corresponding type.
```

```
1578 It is also used to specify the field name of a structure.
1579 Most often, the
1580 .Sx \&Fa
1581 macro is used in the
1582 .Em SYNOPSIS
1583 within
1584 .Sx \&Fo
1585 section when documenting multi-line function prototypes.
1586 If invoked with multiple arguments, the arguments are separated by a
1587 comma.
1588 Furthermore, if the following macro is another
1589 .Sx \&Fa ,
1590 the last argument will also have a trailing comma.
1591 .Pp
1592 Examples:
1593 .Dl \&.Fa \(dqconst char *p\(dq
1594 .Dl \&.Fa \(dqint a\(dq \(dqint b\(dq \(dqint c\(dq
1595 .Dl \&.Fa foo
1596 .Pp
1597 See also
1598 .Sx \&Fo .
1599 .Ss \&Fc
1600 End a function context started by
1601 .Sx \&Fo .
1602 .Ss \&Fd
1603 Historically used to document include files.
1604 This usage has been deprecated in favour of
1605 .Sx \&In .
1606 Do not use this macro.
1607 .Pp
1608 See also
1609 .Sx MANUAL STRUCTURE
1610 and
1611 .Sx \&In .
1612 .Ss \&Fl
1613 Command-line flag or option.
1614 Used when listing arguments to command-line utilities.
1615 Prints a fixed-width hyphen
1616 .Sq \-
1617 directly followed by each argument.
1618 If no arguments are provided, a hyphen is printed followed by a space.
1619 If the argument is a macro, a hyphen is prefixed to the subsequent macro
1620 output.
1621 .Pp
1622 Examples:
1623 .Dl ".Fl R Op Fl H | L | P"
1624 .Dl ".Op Fl 1AaCcdFfgHhikLlmnopqRrSsTtux"
1625 .Dl ".Fl type Cm d Fl name Pa CVS"
1626 .Dl ".Fl Ar signal_number"
1627 .Dl ".Fl o Fl"
1628 .Pp
1629 See also
1630 .Sx \&Cm .
1631 .Ss \&Fn
1632 A function name.
1633 Its syntax is as follows:
1634 .Bd -ragged -offset indent
1635 .Pf \. Ns Sx \&Fn
1636 .Op Ar functype
1637 .Ar funcname
1638 .Op Oo Ar argtype Oc Ar argname
1639 .Ed
1640 .Pp
1641 Function arguments are surrounded in parenthesis and
1642 are delimited by commas.
1643 If no arguments are specified, blank parenthesis are output.
```



```
1644 In the
1645 .Em SYNOPSIS
1646 section, this macro starts a new output line,
1647 and a blank line is automatically inserted between function definitions.
1648 .Pp
1649 Examples:
1650 .Dl \&.Fn \(dqint funcname\(dq \(dqint arg0\(dq \(dqint arg1\(dq
1651 .Dl \&.Fn funcname \(dqint arg0\(dq
1652 .Dl \&.Fn funcname arg0
1653 .Pp
1654 .Bd -literal -offset indent -compact
1655 \&.Ft functype
1656 \&.Fn funcname
1657 .Ed
1658 .Pp
1659 When referring to a function documented in another manual page, use
1660 .Sx \&Xr
1661 instead.
1662 See also
1663 .Sx MANUAL STRUCTURE ,
1664 .Sx \&Fo ,
1665 and
1666 .Sx \&Ft .
1667 .Ss \&Fo
1668 Begin a function block.
1669 This is a multi-line version of
1670 .Sx \&Fn .
1671 Its syntax is as follows:
1672 .Pp
1673 .D1 Pf \. Sx \&Fo Ar funcname
1674 .Pp
1675 Invocations usually occur in the following context:
1676 .Bd -ragged -offset indent
1677 .Pf \. Sx \&Ft Ar functype
1678 .br
1679 .Pf \. Sx \&Fo Ar funcname
1680 .br
1681 .Pf \. Sx \&Fa Oo Ar argtype Oc Ar argname
1682 .br
1683 \&.\.\.
1684 .br
1685 .Pf \. Sx \&Fc
1686 .Ed
1687 .Pp
1688 A
1689 .Sx \&Fo
1690 scope is closed by
1691 .Sx \&Fc .
1692 .Pp
1693 See also
1694 .Sx MANUAL STRUCTURE ,
1695 .Sx \&Fa ,
1696 .Sx \&Fc ,
1697 and
1698 .Sx \&Ft .
1699 .Ss \&Fr
1700 This macro is obsolete and not implemented in
1701 .Xr mandoc 1 .
1702 .Pp
1703 It was used to show function return values.
1704 The syntax was:
1705 .Pp
1706 .Dl Pf . Sx \&Fr Ar value
1707 .Ss \&Ft
1708 A function type.
1709 Its syntax is as follows:
```

```
.Pp
.D1 Pf \. Sx \&Ft Ar functype
.Pp
In the
.Em SYNOPSIS
section, a new output line is started after this macro.
.Pp
Examples:
.Dl \&.Ft int
.Bd -literal -offset indent -compact
\&.Ft functype
\&.Fn funcname
.Ed
.Pp
See also
.Sx MANUAL STRUCTURE ,
.Sx \&Fn ,
and
.Sx \&Fo .
.Ss \&Fx
Format the
.Fx
version provided as an argument, or a default value
if no argument is provided.
.Pp
Examples:
.Dl \&.Fx 7.1
.Dl \&.Fx
.Pp
See also
.Sx \&At ,
.Sx \&Bsx ,
.Sx \&Bx ,
.Sx \&Dx ,
.Sx \&Nx ,
.Sx \&Ox ,
and
.Sx \&Ux .
.Ss \&Hf
This macro is not implemented in
.Xr mandoc 1 .
.Pp
It was used to include the contents of a (header) file literally.
The syntax was:
.Pp
.Dl Pf . Sx \&Hf Ar filename
.Ss \&Ic
Designate an internal or interactive command.
This is similar to
.Sx \&Cm
but used for instructions rather than values.
.Pp
Examples:
.Dl \&.Ic :wq
.Dl \&.Ic hash
.Dl \&.Ic alias
.Pp
Note that using
.Sx \&Bd Fl literal
or
.Sx \&D1
is preferred for displaying code; the
.Sx \&Ic
macro is used when referring to specific instructions.
.Ss \&In
An
```



```
.Dq include
file.
When invoked as the first macro on an input line in the
.Em SYNOPSIS
section, the argument is displayed in angle brackets
and preceded by
.Dq #include ,
and a blank line is inserted in front if there is a preceding
function declaration.
This is most often used in section 2, 3, and 9 manual pages.
.Pp
Examples:
.Dl \&.In sys/types.h
.Pp
See also
.Sx MANUAL STRUCTURE .
.Ss \&It
A list item.
The syntax of this macro depends on the list type.
.Pp
Lists
of type
.Fl hang ,
.Fl ohang ,
.Fl inset ,
and
.Fl diag
have the following syntax:
.Pp
.D1 Pf \. Sx \&It Ar args
.Pp
Lists of type
.Fl bullet ,
.Fl dash ,
.Fl enum ,
.Fl hyphen
and
.Fl item
have the following syntax:
.Pp
.D1 Pf \. Sx \&It
.Pp
with subsequent lines interpreted within the scope of the
.Sx \&It
until either a closing
.Sx \&El
or another
.Sx \&It .
.Pp
The
.Fl tag
list has the following syntax:
.Pp
.D1 Pf \. Sx \&It Op Cm args
.Pp
Subsequent lines are interpreted as with
.Fl bullet
and family.
The line arguments correspond to the list’s left-hand side; body
arguments correspond to the list’s contents.
.Pp
The
.Fl column
list is the most complicated.
Its syntax is as follows:
.Pp
```

```
.D1 Pf \. Sx \&It Ar cell Op <TAB> Ar cell ...
.D1 Pf \. Sx \&It Ar cell Op Sx \&Ta Ar cell ...
.Pp
The arguments consist of one or more lines of text and macros
representing a complete table line.
Cells within the line are delimited by tabs or by the special
.Sx \&Ta
block macro.
The tab cell delimiter may only be used within the
.Sx \&It
line itself; on following lines, only the
.Sx \&Ta
macro can be used to delimit cells, and
.Sx \&Ta
is only recognised as a macro when called by other macros,
not as the first macro on a line.
.Pp
Note that quoted strings may span tab-delimited cells on an
.Sx \&It
line.
For example,
.Pp
.Dl .It \(dqcol1 ; <TAB> col2 ;\(dq \&;
.Pp
will preserve the semicolon whitespace except for the last.
.Pp
See also
.Sx \&Bl .
.Ss \&Lb
Specify a library.
The syntax is as follows:
.Pp
.D1 Pf \. Sx \&Lb Ar library
.Pp
The
.Ar library
parameter may be a system library, such as
.Cm libz
or
.Cm libpam ,
in which case a small library description is printed next to the linker
invocation; or a custom library, in which case the library name is
printed in quotes.
This is most commonly used in the
.Em SYNOPSIS
section as described in
.Sx MANUAL STRUCTURE .
.Pp
Examples:
.Dl \&.Lb libz
.Dl \&.Lb mdoc
.Ss \&Li
Denotes text that should be in a
.Li literal
font mode.
Note that this is a presentation term and should not be used for
stylistically decorating technical terms.
.Pp
On terminal output devices, this is often indistinguishable from
normal text.
.Pp
See also
.Sx \&Bf ,
.Sx \&Em ,
.Sx \&No ,
and
```



```
.Sx \&Sy .
.Ss \&Lk
Format a hyperlink.
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&Lk Ar uri Op Ar name
.Pp
Examples:
.Dl \&.Lk http://bsd.lv \(dqThe BSD.lv Project\(dq
.Dl \&.Lk http://bsd.lv
.Pp
See also
.Sx \&Mt .
.Ss \&Lp
Synonym for
.Sx \&Pp .
.Ss \&Ms
Display a mathematical symbol.
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&Ms Ar symbol
.Pp
Examples:
.Dl \&.Ms sigma
.Dl \&.Ms aleph
.Ss \&Mt
Format a
.Dq mailto:
hyperlink.
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&Mt Ar address
.Pp
Examples:
.Dl \&.Mt discuss@manpages.bsd.lv
.Ss \&Nd
A one line description of the manual’s content.
This may only be invoked in the
.Em SYNOPSIS
section subsequent the
.Sx \&Nm
macro.
.Pp
Examples:
.Dl Pf . Sx \&Nd mdoc language reference
.Dl Pf . Sx \&Nd format and display UNIX manuals
.Pp
The
.Sx \&Nd
macro technically accepts child macros and terminates with a subsequent
.Sx \&Sh
invocation.
Do not assume this behaviour: some
.Xr whatis 1
database generators are not smart enough to parse more than the line
arguments and will display macros verbatim.
.Pp
See also
.Sx \&Nm .
.Ss \&Nm
The name of the manual page, or \(em in particular in section 1, 6,
and 8 pages \(em of an additional command or feature documented in
the manual page.
When first invoked, the
.Sx \&Nm
macro expects a single argument, the name of the manual page.
```

```
1974 Usually, the first invocation happens in the
1975 .Em NAME
1976 section of the page.
1977 The specified name will be remembered and used whenever the macro is
1978 called again without arguments later in the page.
1979 The
1980 .Sx \&Nm
1981 macro uses
1982 .Sx Block full-implicit
1983 semantics when invoked as the first macro on an input line in the
1984 .Em SYNOPSIS
1985 section; otherwise, it uses ordinary
1986 .Sx In-line
1987 semantics.
1988 .Pp
1989 Examples:
1990 .Bd -literal -offset indent
1991 \&.Sh SYNOPSIS
1992 \&.Nm cat
1993 \&.Op Fl benstuv
1994 \&.Op Ar
1995 .Ed
1996 .Pp
1997 In the
1998 .Em SYNOPSIS
1999 of section 2, 3 and 9 manual pages, use the
2000 .Sx \&Fn
2001 macro rather than
2002 .Sx \&Nm
2003 to mark up the name of the manual page.
2004 .Ss \&No
2005 Normal text.
2006 Closes the scope of any preceding in-line macro.
2007 When used after physical formatting macros like
2008 .Sx \&Em
2009 or
2010 .Sx \&Sy ,
2011 switches back to the standard font face and weight.
2012 Can also be used to embed plain text strings in macro lines
2013 using semantic annotation macros.
2014 .Pp
2015 Examples:
2016 .Dl ".Em italic , Sy bold , No and roman"
2017 .Pp
2018 .Bd -literal -offset indent -compact
2019 \&.Sm off
2020 \&.Cm :C No / Ar pattern No / Ar replacement No /
2021 \&.Sm on
2022 .Ed
2023 .Pp
2024 See also
2025 .Sx \&Em ,
2026 .Sx \&Li ,
2027 and
2028 .Sx \&Sy .
2029 .Ss \&Ns
2030 Suppress a space between the output of the preceding macro
2031 and the following text or macro.
2032 Following invocation, input is interpreted as normal text
2033 just like after an
2034 .Sx \&No
2035 macro.
2036 .Pp
2037 This has no effect when invoked at the start of a macro line.
2038 .Pp
2039 Examples:
```



```
2040 .Dl ".Ar name Ns = Ns Ar value"
2041 .Dl ".Cm :M Ns Ar pattern"
2042 .Dl ".Fl o Ns Ar output"
2043 .Pp
2044 See also
2045 .Sx \&No
2046 and
2047 .Sx \&Sm .
2048 .Ss \&Nx
2049 Format the
2050 .Nx
2051 version provided as an argument, or a default value if
2052 no argument is provided.
2053 .Pp
2054 Examples:
2055 .Dl \&.Nx 5.01
2056 .Dl \&.Nx
2057 .Pp
2058 See also
2059 .Sx \&At ,
2060 .Sx \&Bsx ,
2061 .Sx \&Bx ,
2062 .Sx \&Dx ,
2063 .Sx \&Fx ,
2064 .Sx \&Ox ,
2065 and
2066 .Sx \&Ux .
2067 .Ss \&Oc
2068 Close multi-line
2069 .Sx \&Oo
2070 context.
2071 .Ss \&Oo
2072 Multi-line version of
2073 .Sx \&Op .
2074 .Pp
2075 Examples:
2076 .Bd -literal -offset indent -compact
2077 \&.Oo
2078 \&.Op Fl flag Ns Ar value
2079 \&.Oc
2080 .Ed
2081 .Ss \&Op
2082 Optional part of a command line.
2083 Prints the argument(s) in brackets.
2084 This is most often used in the
2085 .Em SYNOPSIS
2086 section of section 1 and 8 manual pages.
2087 .Pp
2088 Examples:
2089 .Dl \&.Op \&Fl a \&Ar b
2090 .Dl \&.Op \&Ar a | b
2091 .Pp
2092 See also
2093 .Sx \&Oo .
2094 .Ss \&Os
2095 Document operating system version.
2096 This is the mandatory third macro of
2097 any
2098 .Nm
2099 file.
2100 Its syntax is as follows:
2101 .Pp
2102 .D1 Pf \. Sx \&Os Op Ar system Op Ar version
2103 .Pp
2104 The optional
2105 .Ar system
```

```
2106 parameter specifies the relevant operating system or environment.
2107 Left unspecified, it defaults to the local operating system version.
2108 This is the suggested form.
2109 .Pp
2110 Examples:
2111 .Dl \&.Os
2112 .Dl \&.Os KTH/CSC/TCS
2113 .Dl \&.Os BSD 4.3
2114 .Pp
2115 See also
2116 .Sx \&Dd
2117 and
2118 .Sx \&Dt .
2119 .Ss \&Ot
2120 This macro is obsolete and not implemented in
2121 .Xr mandoc 1 .
2122 .Pp
2123 Historical
2124 .Xr mdoc 5
2125 packages described it as
2126 .Dq "old function type (FORTRAN)" .
2127 .Ss \&Ox
2128 Format the
2129 .Ox
2130 version provided as an argument, or a default value
2131 if no argument is provided.
2132 .Pp
2133 Examples:
2134 .Dl \&.Ox 4.5
2135 .Dl \&.Ox
2136 .Pp
2137 See also
2138 .Sx \&At ,
2139 .Sx \&Bsx ,
2140 .Sx \&Bx ,
2141 .Sx \&Dx ,
2142 .Sx \&Fx ,
2143 .Sx \&Nx ,
2144 and
2145 .Sx \&Ux .
2146 .Ss \&Pa
2147 An absolute or relative file system path, or a file or directory name.
2148 If an argument is not provided, the character
2149 .Sq \(ti
2150 is used as a default.
2151 .Pp
2152 Examples:
2153 .Dl \&.Pa /usr/bin/mandoc
2154 .Dl \&.Pa /usr/share/man/man5/mdoc.5
2155 .Pp
2156 See also
2157 .Sx \&Lk .
2158 .Ss \&Pc
2159 Close parenthesised context opened by
2160 .Sx \&Po .
2161 .Ss \&Pf
2162 Removes the space between its argument
2163 .Pq Dq prefix
2164 and the following macro.
2165 Its syntax is as follows:
2166 .Pp
2167 .D1 .Pf Ar prefix macro arguments ...
2168 .Pp
2169 This is equivalent to:
2170 .Pp
2171 .D1 .No Ar prefix No \&Ns Ar macro arguments ...
```



```
2172 .Pp
2173 Examples:
2174 .Dl ".Pf $ Ar variable_name"
2175 .Dl ".Pf 0x Ar hex_digits"
2176 .Pp
2177 See also
2178 .Sx \&Ns
2179 and
2180 .Sx \&Sm .
2181 .Ss \&Po
2182 Multi-line version of
2183 .Sx \&Pq .
2184 .Ss \&Pp
2185 Break a paragraph.
2186 This will assert vertical space between prior and subsequent macros
2187 and/or text.
2188 .Pp
2189 Paragraph breaks are not needed before or after
2190 .Sx \&Sh
2191 or
2192 .Sx \&Ss
2193 macros or before displays
2194 .Pq Sx \&Bd
2195 or lists
2196 .Pq Sx \&Bl
2197 unless the
2198 .Fl compact
2199 flag is given.
2200 .Ss \&Pq
2201 Parenthesised enclosure.
2202 .Pp
2203 See also
2204 .Sx \&Po .
2205 .Ss \&Qc
2206 Close quoted context opened by
2207 .Sx \&Qo .
2208 .Ss \&Ql
2209 Format a single-quoted literal.
2210 See also
2211 .Sx \&Qq
2212 and
2213 .Sx \&Sq .
2214 .Ss \&Qo
2215 Multi-line version of
2216 .Sx \&Qq .
2217 .Ss \&Qq
2218 Encloses its arguments in
2219 .Qq typewriter
2220 double-quotes.
2221 Consider using
2222 .Sx \&Dq .
2223 .Pp
2224 See also
2225 .Sx \&Dq ,
2226 .Sx \&Sq ,
2227 and
2228 .Sx \&Qo .
2229 .Ss \&Re
2230 Close an
2231 .Sx \&Rs
2232 block.
2233 Does not have any tail arguments.
2234 .Ss \&Rs
2235 Begin a bibliographic
2236 .Pq Dq reference
2237 block.
```

```
Does not have any head arguments.
The block macro may only contain
.Sx \&%A ,
.Sx \&%B ,
.Sx \&%C ,
.Sx \&%D ,
.Sx \&%I ,
.Sx \&%J ,
.Sx \&%N ,
.Sx \&%O ,
.Sx \&%P ,
.Sx \&%Q ,
.Sx \&%R ,
.Sx \&%T ,
.Sx \&%U ,
and
.Sx \&%V
child macros (at least one must be specified).
.Pp
Examples:
.Bd -literal -offset indent -compact
\&.Rs
\&.%A J. E. Hopcroft
\&.%A J. D. Ullman
\&.%B Introduction to Automata Theory, Languages, and Computation
\&.%I Addison-Wesley
\&.%C Reading, Massachusettes
\&.%D 1979
\&.Re
.Ed
.Pp
If an
.Sx \&Rs
block is used within a SEE ALSO section, a vertical space is asserted
before the rendered output, else the block continues on the current
line.
.Ss \&Rv
Insert a standard sentence regarding a function call’s return value of 0
on success and \-1 on error, with the
.Va errno
libc global variable set on error.
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&Rv Fl std Op Ar function ...
.Pp
If
.Ar function
is not specified, the document’s name set by
.Sx \&Nm
is used.
Multiple
.Ar function
arguments are treated as separate functions.
.Pp
See also
.Sx \&Ex .
.Ss \&Sc
Close single-quoted context opened by
.Sx \&So .
.Ss \&Sh
Begin a new section.
For a list of conventional manual sections, see
.Sx MANUAL STRUCTURE .
These sections should be used unless it’s absolutely necessary that
custom sections be used.
.Pp
```



```
Section names should be unique so that they may be keyed by
.Sx \&Sx .
Although this macro is parsed, it should not consist of child node or it
may not be linked with
.Sx \&Sx .
.Pp
See also
.Sx \&Pp ,
.Sx \&Ss ,
and
.Sx \&Sx .
.Ss \&Sm
Switches the spacing mode for output generated from macros.
Its syntax is as follows:
.Pp
.D1 Pf \. Sx \&Sm Cm on | off
.Pp
By default, spacing is
.Cm on .
When switched
.Cm off ,
no white space is inserted between macro arguments and between the
output generated from adjacent macros, but text lines
still get normal spacing between words and sentences.
.Ss \&So
Multi-line version of
.Sx \&Sq .
.Ss \&Sq
Encloses its arguments in
.Sq typewriter
single-quotes.
.Pp
See also
.Sx \&Dq ,
.Sx \&Qq ,
and
.Sx \&So .
.Ss \&Ss
Begin a new subsection.
Unlike with
.Sx \&Sh ,
there is no convention for the naming of subsections.
Except
.Em DESCRIPTION ,
the conventional sections described in
.Sx MANUAL STRUCTURE
rarely have subsections.
.Pp
Sub-section names should be unique so that they may be keyed by
.Sx \&Sx .
Although this macro is parsed, it should not consist of child node or it
may not be linked with
.Sx \&Sx .
.Pp
See also
.Sx \&Pp ,
.Sx \&Sh ,
and
.Sx \&Sx .
.Ss \&St
Replace an abbreviation for a standard with the full form.
The following standards are recognised:
.Pp
.Bl -tag -width "-p1003.1g-2000X" -compact
.It \-p1003.1-88
.St -p1003.1-88
```

```
.It \-p1003.1-90
.St -p1003.1-90
.It \-p1003.1-96
.St -p1003.1-96
.It \-p1003.1-2001
.St -p1003.1-2001
.It \-p1003.1-2004
.St -p1003.1-2004
.It \-p1003.1-2008
.St -p1003.1-2008
.It \-p1003.1
.St -p1003.1
.It \-p1003.1b
.St -p1003.1b
.It \-p1003.1b-93
.St -p1003.1b-93
.It \-p1003.1c-95
.St -p1003.1c-95
.It \-p1003.1g-2000
.St -p1003.1g-2000
.It \-p1003.1i-95
.St -p1003.1i-95
.It \-p1003.2-92
.St -p1003.2-92
.It \-p1003.2a-92
.St -p1003.2a-92
.It \-p1387.2-95
.St -p1387.2-95
.It \-p1003.2
.St -p1003.2
.It \-p1387.2
.St -p1387.2
.It \-isoC
.St -isoC
.It \-isoC-90
.St -isoC-90
.It \-isoC-amd1
.St -isoC-amd1
.It \-isoC-tcor1
.St -isoC-tcor1
.It \-isoC-tcor2
.St -isoC-tcor2
.It \-isoC-99
.St -isoC-99
.It \-isoC-2011
.St -isoC-2011
.It \-iso9945-1-90
.St -iso9945-1-90
.It \-iso9945-1-96
.St -iso9945-1-96
.It \-iso9945-2-93
.St -iso9945-2-93
.It \-ansiC
.St -ansiC
.It \-ansiC-89
.St -ansiC-89
.It \-ansiC-99
.St -ansiC-99
.It \-ieee754
.St -ieee754
.It \-iso8802-3
.St -iso8802-3
.It \-iso8601
.St -iso8601
.It \-ieee1275-94
.St -ieee1275-94
```



```
.It \-xpg3
.St -xpg3
.It \-xpg4
.St -xpg4
.It \-xpg4.2
.St -xpg4.2
.It \-xpg4.3
.St -xpg4.3
.It \-xbd5
.St -xbd5
.It \-xcu5
.St -xcu5
.It \-xsh5
.St -xsh5
.It \-xns5
.St -xns5
.It \-xns5.2
.St -xns5.2
.It \-xns5.2d2.0
.St -xns5.2d2.0
.It \-xcurses4.2
.St -xcurses4.2
.It \-susv2
.St -susv2
.It \-susv3
.St -susv3
.It \-svid4
.St -svid4
.El
.Ss \&Sx
Reference a section or subsection in the same manual page.
The referenced section or subsection name must be identical to the
enclosed argument, including whitespace.
.Pp
Examples:
.Dl \&.Sx MANUAL STRUCTURE
.Pp
See also
.Sx \&Sh
and
.Sx \&Ss .
.Ss \&Sy
Format enclosed arguments in symbolic
.Pq Dq boldface .
Note that this is a presentation term and should not be used for
stylistically decorating technical terms.
.Pp
See also
.Sx \&Bf ,
.Sx \&Em ,
.Sx \&Li ,
and
.Sx \&No .
.Ss \&Ta
Table cell separator in
.Sx \&Bl Fl column
lists; can only be used below
.Sx \&It .
.Ss \&Tn
Format a tradename.
.Pp
Since this macro is often implemented to use a small caps font,
it has historically been used for acronyms (like ASCII) as well.
Such usage is not recommended because it would use the same macro
sometimes for semantical annotation, sometimes for physical formatting.
.Pp
```

```
2502 Examples:
2503 .Dl \&.Tn IBM
2504 .Ss \&Ud
2505 Prints out
2506 .Dq currently under development.
2507 .Ss \&Ux
2508 Format the UNIX name.
2509 Accepts no argument.
2510 .Pp
2511 Examples:
2512 .Dl \&.Ux
2513 .Pp
2514 See also
2515 .Sx \&At ,
2516 .Sx \&Bsx ,
2517 .Sx \&Bx ,
2518 .Sx \&Dx ,
2519 .Sx \&Fx ,
2520 .Sx \&Nx ,
2521 and
2522 .Sx \&Ox .
2523 .Ss \&Va
2524 A variable name.
2525 .Pp
2526 Examples:
2527 .Dl \&.Va foo
2528 .Dl \&.Va const char *bar ;
2529 .Ss \&Vt
2530 A variable type.
2531 This is also used for indicating global variables in the
2532 .Em SYNOPSIS
2533 section, in which case a variable name is also specified.
2534 Note that it accepts
2535 .Sx Block partial-implicit
2536 syntax when invoked as the first macro on an input line in the
2537 .Em SYNOPSIS
2538 section, else it accepts ordinary
2539 .Sx In-line
2540 syntax.
2541 In the former case, this macro starts a new output line,
2542 and a blank line is inserted in front if there is a preceding
2543 function definition or include directive.
2544 .Pp
2545 Note that this should not be confused with
2546 .Sx \&Ft ,
2547 which is used for function return types.
2548 .Pp
2549 Examples:
2550 .Dl \&.Vt unsigned char
2551 .Dl \&.Vt extern const char * const sys_signame[] \&;
2552 .Pp
2553 See also
2554 .Sx MANUAL STRUCTURE
2555 and
2556 .Sx \&Va .
2557 .Ss \&Xc
2558 Close a scope opened by
2559 .Sx \&Xo .
2560 .Ss \&Xo
2561 Extend the header of an
2562 .Sx \&It
2563 macro or the body of a partial-implicit block macro
2564 beyond the end of the input line.
2565 This macro originally existed to work around the 9-argument limit
2566 of historic
2567 .Xr roff 5 .
```



```
2568 .Ss \&Xr
2569 Link to another manual
2570 .Pq Qq cross-reference .
2571 Its syntax is as follows:
2572 .Pp
2573 .D1 Pf \. Sx \&Xr Ar name section
2574 .Pp
2575 The
2576 .Ar name
2577 and
2578 .Ar section
2579 are the name and section of the linked manual.
2580 If
2581 .Ar section
2582 is followed by non-punctuation, an
2583 .Sx \&Ns
2584 is inserted into the token stream.
2585 This behaviour is for compatibility with
2586 GNU troff.
2587 .Pp
2588 Examples:
2589 .Dl \&.Xr mandoc 1
2590 .Dl \&.Xr mandoc 1 \&;
2591 .Dl \&.Xr mandoc 1 \&Ns s behaviour
2592 .Ss \&br
2593 Emits a line-break.
2594 This macro should not be used; it is implemented for compatibility with
2595 historical manuals.
2596 .Pp
2597 Consider using
2598 .Sx \&Pp
2599 in the event of natural paragraph breaks.
2600 .Ss \&sp
2601 Emits vertical space.
2602 This macro should not be used; it is implemented for compatibility with
2603 historical manuals.
2604 Its syntax is as follows:
2605 .Pp
2606 .D1 Pf \. Sx \&sp Op Ar height
2607 .Pp
2608 The
2609 .Ar height
2610 argument must be formatted as described in
2611 .Sx Scaling Widths .
2612 If unspecified,
2613 .Sx \&sp
2614 asserts a single vertical space.
2615 .Sh MACRO SYNTAX
2616 The syntax of a macro depends on its classification.
2617 In this section,
2618 .Sq \-arg
2619 refers to macro arguments, which may be followed by zero or more
2620 .Sq parm
2621 parameters;
2622 .Sq \&Yo
2623 opens the scope of a macro; and if specified,
2624 .Sq \&Yc
2625 closes it out.
2626 .Pp
2627 The
2628 .Em Callable
2629 column indicates that the macro may also be called by passing its name
2630 as an argument to another macro.
2631 For example,
2632 .Sq \&.Op \&Fl O \&Ar file
2633 produces
```

```
.Sq Op Fl O Ar file .
To prevent a macro call and render the macro name literally,
escape it by prepending a zero-width space,
.Sq \e& .
For example,
.Sq \&Op \e&Fl O
produces
.Sq Op \&Fl O .
If a macro is not callable but its name appears as an argument
to another macro, it is interpreted as opaque text.
For example,
.Sq \&.Fl \&Sh
produces
.Sq Fl \&Sh .
.Pp
The
.Em Parsed
column indicates whether the macro may call other macros by receiving
their names as arguments.
If a macro is not parsed but the name of another macro appears
as an argument, it is interpreted as opaque text.
.Pp
The
.Em Scope
column, if applicable, describes closure rules.
.Ss Block full-explicit
Multi-line scope closed by an explicit closing macro.
All macros contains bodies; only
.Sx \&Bf
and
.Pq optionally
.Sx \&Bl
contain a head.
.Bd -literal -offset indent
\&.Yo \(lB\-arg \(lBparm...\(rB\(rB \(lBhead...\(rB
\(lBbody...\(rB
\&.Yc
.Ed
.Bl -column "MacroX" "CallableX" "ParsedX" "closed by XXX" -offset indent
.It Em Macro Ta Em Callable Ta Em Parsed Ta Em Scope
.It Sx \&Bd  Ta    \&No     Ta    \&No     Ta    closed by Sx \&Ed
.It Sx \&Bf  Ta    \&No     Ta    \&No     Ta    closed by Sx \&Ef
.It Sx \&Bk  Ta    \&No     Ta    \&No     Ta    closed by Sx \&Ek
.It Sx \&Bl  Ta    \&No     Ta    \&No     Ta    closed by Sx \&El
.It Sx \&Ed  Ta    \&No     Ta    \&No     Ta    opened by Sx \&Bd
.It Sx \&Ef  Ta    \&No     Ta    \&No     Ta    opened by Sx \&Bf
.It Sx \&Ek  Ta    \&No     Ta    \&No     Ta    opened by Sx \&Bk
.It Sx \&El  Ta    \&No     Ta    \&No     Ta    opened by Sx \&Bl
.El
.Ss Block full-implicit
Multi-line scope closed by end-of-file or implicitly by another macro.
All macros have bodies; some
.Po
.Sx \&It Fl bullet ,
.Fl hyphen ,
.Fl dash ,
.Fl enum ,
.Fl item
.Pc
don’t have heads; only one
.Po
.Sx \&It
in
.Sx \&Bl Fl column
.Pc
has multiple heads.
```



```
.Bd -literal -offset indent
\&.Yo \(lB\-arg \(lBparm...\(rB\(rB \(lBhead... \(lBTa head...\(rB\(rB
\(lBbody...\(rB
.Ed
.Bl -column "MacroX" "CallableX" "ParsedX" "closed by XXXXXXXXXX" -offset inden
.It Em Macro Ta Em Callable Ta Em Parsed Ta Em Scope
.It Sx \&It Ta \&No Ta Yes  Ta closed by Sx \&It , Sx \&El
.It Sx \&Nd Ta \&No Ta \&No Ta closed by Sx \&Sh
.It Sx \&Nm Ta \&No Ta Yes  Ta closed by Sx \&Nm , Sx \&Sh , Sx \&Ss
.It Sx \&Sh Ta \&No Ta Yes  Ta closed by Sx \&Sh
.It Sx \&Ss Ta \&No Ta Yes  Ta closed by Sx \&Sh , Sx \&Ss
.El
.Pp
Note that the
.Sx \&Nm
macro is a
.Sx Block full-implicit
macro only when invoked as the first macro
in a
.Em SYNOPSIS
section line, else it is
.Sx In-line .
.Ss Block partial-explicit
Like block full-explicit, but also with single-line scope.
Each has at least a body and, in limited circumstances, a head
.Po
.Sx \&Fo ,
.Sx \&Eo
.Pc
and/or tail
.Pq Sx \&Ec .
.Bd -literal -offset indent
\&.Yo \(lB\-arg \(lBparm...\(rB\(rB \(lBhead...\(rB
\(lBbody...\(rB
\&.Yc \(lBtail...\(rB

\&.Yo \(lB\-arg \(lBparm...\(rB\(rB \(lBhead...\(rB \
\(lBbody...\(rB \&Yc \(lBtail...\(rB
.Ed
.Bl -column "MacroX" "CallableX" "ParsedX" "closed by XXXX" -offset indent
.It Em Macro Ta Em Callable Ta Em Parsed Ta Em Scope
.It Sx \&Ac  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Ao
.It Sx \&Ao  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Ac
.It Sx \&Bc  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Bo
.It Sx \&Bo  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Bc
.It Sx \&Brc Ta    Yes      Ta    Yes      Ta    opened by Sx \&Bro
.It Sx \&Bro Ta    Yes      Ta    Yes      Ta    closed by Sx \&Brc
.It Sx \&Dc  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Do
.It Sx \&Do  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Dc
.It Sx \&Ec  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Eo
.It Sx \&Eo  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Ec
.It Sx \&Fc  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Fo
.It Sx \&Fo  Ta    \&No     Ta    \&No     Ta    closed by Sx \&Fc
.It Sx \&Oc  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Oo
.It Sx \&Oo  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Oc
.It Sx \&Pc  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Po
.It Sx \&Po  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Pc
.It Sx \&Qc  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Oo
.It Sx \&Qo  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Oc
.It Sx \&Re  Ta    \&No     Ta    \&No     Ta    opened by Sx \&Rs
.It Sx \&Rs  Ta    \&No     Ta    \&No     Ta    closed by Sx \&Re
.It Sx \&Sc  Ta    Yes      Ta    Yes      Ta    opened by Sx \&So
.It Sx \&So  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Sc
.It Sx \&Xc  Ta    Yes      Ta    Yes      Ta    opened by Sx \&Xo
.It Sx \&Xo  Ta    Yes      Ta    Yes      Ta    closed by Sx \&Xc
.El
```

```
.Ss Block partial-implicit
Like block full-implicit, but with single-line scope closed by the
end of the line.
.Bd -literal -offset indent
\&.Yo \(lB\-arg \(lBval...\(rB\(rB \(lBbody...\(rB \(lBres...\(rB
.Ed
.Bl -column "MacroX" "CallableX" "ParsedX" -offset indent
.It Em Macro Ta Em Callable Ta Em Parsed
.It Sx \&Aq  Ta    Yes      Ta    Yes
.It Sx \&Bq  Ta    Yes      Ta    Yes
.It Sx \&Brq Ta    Yes      Ta    Yes
.It Sx \&D1  Ta    \&No     Ta    \&Yes
.It Sx \&Dl  Ta    \&No     Ta    Yes
.It Sx \&Dq  Ta    Yes      Ta    Yes
.It Sx \&Op  Ta    Yes      Ta    Yes
.It Sx \&Pq  Ta    Yes      Ta    Yes
.It Sx \&Ql  Ta    Yes      Ta    Yes
.It Sx \&Qq  Ta    Yes      Ta    Yes
.It Sx \&Sq  Ta    Yes      Ta    Yes
.It Sx \&Vt  Ta    Yes      Ta    Yes
.El
.Pp
Note that the
.Sx \&Vt
macro is a
.Sx Block partial-implicit
only when invoked as the first macro
in a
.Em SYNOPSIS
section line, else it is
.Sx In-line .
.Ss Special block macro
The
.Sx \&Ta
macro can only be used below
.Sx \&It
in
.Sx \&Bl Fl column
lists.
It delimits blocks representing table cells;
these blocks have bodies, but no heads.
.Bl -column "MacroX" "CallableX" "ParsedX" "closed by XXXX" -offset indent
.It Em Macro Ta Em Callable Ta Em Parsed Ta Em Scope
.It Sx \&Ta  Ta    Yes      Ta    Yes    Ta closed by Sx \&Ta , Sx \&It
.El
.Ss In-line
Closed by the end of the line, fixed argument lengths,
and/or subsequent macros.
In-line macros have only text children.
If a number (or inequality) of arguments is
.Pq n ,
then the macro accepts an arbitrary number of arguments.
.Bd -literal -offset indent
\&.Yo \(lB\-arg \(lBval...\(rB\(rB \(lBargs...\(rB \(lBres...\(rB

\&.Yo \(lB\-arg \(lBval...\(rB\(rB \(lBargs...\(rB Yc...

\&.Yo \(lB\-arg \(lBval...\(rB\(rB arg0 arg1 argN
.Ed
.Bl -column "MacroX" "CallableX" "ParsedX" "Arguments" -offset indent
.It Em Macro Ta Em Callable Ta Em Parsed Ta Em Arguments
.It Sx \&%A  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%B  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%C  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%D  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%I  Ta    \&No     Ta    \&No     Ta    >0
```



```
.It Sx \&%J  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%N  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%O  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%P  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%Q  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%R  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%T  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%U  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&%V  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&Ad  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&An  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Ap  Ta    Yes      Ta    Yes      Ta    0
.It Sx \&Ar  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&At  Ta    Yes      Ta    Yes      Ta    1
.It Sx \&Bsx Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Bt  Ta    \&No     Ta    \&No     Ta    0
.It Sx \&Bx  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Cd  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Cm  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Db  Ta    \&No     Ta    \&No     Ta    1
.It Sx \&Dd  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Dt  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Dv  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Dx  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Em  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&En  Ta    \&No     Ta    \&No     Ta    0
.It Sx \&Er  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Es  Ta    \&No     Ta    \&No     Ta    0
.It Sx \&Ev  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Ex  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Fa  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Fd  Ta    \&No     Ta    \&No     Ta    >0
.It Sx \&Fl  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Fn  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Fr  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Ft  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Fx  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Hf  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Ic  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&In  Ta    \&No     Ta    \&No     Ta    1
.It Sx \&Lb  Ta    \&No     Ta    \&No     Ta    1
.It Sx \&Li  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Lk  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Lp  Ta    \&No     Ta    \&No     Ta    0
.It Sx \&Ms  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Mt  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Nm  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&No  Ta    Yes      Ta    Yes      Ta    0
.It Sx \&Ns  Ta    Yes      Ta    Yes      Ta    0
.It Sx \&Nx  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Os  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Ot  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Ox  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Pa  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Pf  Ta    Yes      Ta    Yes      Ta    1
.It Sx \&Pp  Ta    \&No     Ta    \&No     Ta    0
.It Sx \&Rv  Ta    \&No     Ta    \&No     Ta    n
.It Sx \&Sm  Ta    \&No     Ta    \&No     Ta    1
.It Sx \&St  Ta    \&No     Ta    Yes      Ta    1
.It Sx \&Sx  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Sy  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Tn  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&Ud  Ta    \&No     Ta    \&No     Ta    0
.It Sx \&Ux  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Va  Ta    Yes      Ta    Yes      Ta    n
.It Sx \&Vt  Ta    Yes      Ta    Yes      Ta    >0
```

```
.It Sx \&Xr  Ta    Yes      Ta    Yes      Ta    >0
.It Sx \&br  Ta    \&No     Ta    \&No     Ta    0
.It Sx \&sp  Ta    \&No     Ta    \&No     Ta    1
.El
.Ss Delimiters
When a macro argument consists of one single input character
considered as a delimiter, the argument gets special handling.
This does not apply when delimiters appear in arguments containing
more than one character.
Consequently, to prevent special handling and just handle it
like any other argument, a delimiter can be escaped by prepending
a zero-width space
.Pq Sq \e& .
In text lines, delimiters never need escaping, but may be used
as normal punctuation.
.Pp
For many macros, when the leading arguments are opening delimiters,
these delimiters are put before the macro scope,
and when the trailing arguments are closing delimiters,
these delimiters are put after the macro scope.
For example,
.Pp
.D1 Pf \. \&Aq "( [ word ] ) ."
.Pp
renders as:
.Pp
.D1 Aq ( [ word ] ) .
.Pp
Opening delimiters are:
.Pp
.Bl -tag -width Ds -offset indent -compact
.It \&(
left parenthesis
.It \&[
left bracket
.El
.Pp
Closing delimiters are:
.Pp
.Bl -tag -width Ds -offset indent -compact
.It \&.
period
.It \&,
comma
.It \&:
colon
.It \&;
semicolon
.It \&)
right parenthesis
.It \&]
right bracket
.It \&?
question mark
.It \&!
exclamation mark
.El
.Pp
Note that even a period preceded by a backslash
.Pq Sq \e.\&
gets this special handling; use
.Sq \e&.
to prevent that.
.Pp
Many in-line macros interrupt their scope when they encounter
delimiters, and resume their scope when more arguments follow that
```



```
are not delimiters.
For example,
.Pp
.D1 Pf \. \&Fl "a ( b | c \e*(Ba d ) e"
.Pp
renders as:
.Pp
.D1 Fl a ( b | c \*(Ba d ) e
.Pp
This applies to both opening and closing delimiters,
and also to the middle delimiter:
.Pp
.Bl -tag -width Ds -offset indent -compact
.It \&|
vertical bar
.El
.Pp
As a special case, the predefined string \e*(Ba is handled and rendered
in the same way as a plain
.Sq \&|
character.
Using this predefined string is not recommended in new manuals.
.Ss Font handling
In
.Nm
documents, usage of semantic markup is recommended in order to have
proper fonts automatically selected; only when no fitting semantic markup
is available, consider falling back to
.Sx Physical markup
macros.
Whenever any
.Nm
macro switches the
.Xr roff 5
font mode, it will automatically restore the previous font when exiting
its scope.
Manually switching the font using the
.Xr roff 5
.Ql \ef
font escape sequences is never required.
.Sh COMPATIBILITY
This section documents compatibility between mandoc and other other
troff implementations, at this time limited to GNU troff
.Pq Qq groff .
The term
.Qq historic groff
refers to groff versions before 1.17,
which featured a significant update of the
.Pa doc.tmac
file.
.Pp
Heirloom troff, the other significant troff implementation accepting
\-mdoc, is similar to historic groff.
.Pp
The following problematic behaviour is found in groff:
.ds hist (Historic groff only.)
.Pp
.Bl -dash -compact
.It
Display macros
.Po
.Sx \&Bd ,
.Sx \&Dl ,
and
.Sx \&D1
.Pc
```

```
may not be nested.
\*[hist]
.It
.Sx \&At
with unknown arguments produces no output at all.
\*[hist]
Newer groff and mandoc print
.Qq AT&T UNIX
and the arguments.
.It
.Sx \&Bl Fl column
does not recognise trailing punctuation characters when they immediately
precede tabulator characters, but treats them as normal text and
outputs a space before them.
.It
.Sx \&Bd Fl ragged compact
does not start a new line.
\*[hist]
.It
.Sx \&Dd
with non-standard arguments behaves very strangely.
When there are three arguments, they are printed verbatim.
Any other number of arguments is replaced by the current date,
but without any arguments the string
.Dq Epoch
is printed.
.It
.Sx \&Fl
does not print a dash for an empty argument.
\*[hist]
.It
.Sx \&Fn
does not start a new line unless invoked as the line macro in the
.Em SYNOPSIS
section.
\*[hist]
.It
.Sx \&Fo
with
.Pf non- Sx \&Fa
children causes inconsistent spacing between arguments.
In mandoc, a single space is always inserted between arguments.
.It
.Sx \&Ft
in the
.Em SYNOPSIS
causes inconsistent vertical spacing, depending on whether a prior
.Sx \&Fn
has been invoked.
See
.Sx \&Ft
and
.Sx \&Fn
for the normalised behaviour in mandoc.
.It
.Sx \&In
ignores additional arguments and is not treated specially in the
.Em SYNOPSIS .
\*[hist]
.It
.Sx \&It
sometimes requires a
.Fl nested
flag.
\*[hist]
In new groff and mandoc, any list may be nested by default and
```



```
.Fl enum
lists will restart the sequence only for the sub-list.
.It
.Sx \&Li
followed by a delimiter is incorrectly used in some manuals
instead of properly quoting that character, which sometimes works with
historic groff.
.It
.Sx \&Lk
only accepts a single link-name argument; the remainder is misformatted.
.It
.Sx \&Pa
does not format its arguments when used in the FILES section under
certain list types.
.It
.Sx \&Ta
can only be called by other macros, but not at the beginning of a line.
.It
.Sx \&%C
is not implemented.
.It
Historic groff only allows up to eight or nine arguments per macro input
line, depending on the exact situation.
Providing more arguments causes garbled output.
The number of arguments on one input line is not limited with mandoc.
.It
Historic groff has many un-callable macros.
Most of these (excluding some block-level macros) are callable
in new groff and mandoc.
.It
.Sq \(ba
(vertical bar) is not fully supported as a delimiter.
\*[hist]
.It
.Sq \ef
.Pq font face
and
.Sq \ef
.Pq font family face
.Sx Text Decoration
escapes behave irregularly when specified within line-macro scopes.
.It
Negative scaling units return to prior lines.
Instead, mandoc truncates them to zero.
.El
.Pp
The following features are unimplemented in mandoc:
.Pp
.Bl -dash -compact
.It
.Sx \&Bd
.Fl file Ar file .
.It
.Sx \&Bd
.Fl offset Ar center
and
.Fl offset Ar right .
Groff does not implement centred and flush-right rendering either,
but produces large indentations.
.It
The
.Sq \eh
.Pq horizontal position ,
.Sq \ev
.Pq vertical position ,
.Sq \em
```

```
3162 .Pq text colour ,
3163 .Sq \eM
3164 .Pq text filling colour ,
3165 .Sq \ez
3166 .Pq zero-length character ,
3167 .Sq \ew
3168 .Pq string length ,
3169 .Sq \ek
3170 .Pq horizontal position marker ,
3171 .Sq \eo
3172 .Pq text overstrike ,
3173 and
3174 .Sq \es
3175 .Pq text size
3176 escape sequences are all discarded in mandoc.
3177 .It
3178 The
3179 .Sq \ef
3180 scaling unit is accepted by mandoc, but rendered as the default unit.
3181 .It
3182 In quoted literals, groff allows pairwise double-quotes to produce a
3183 standalone double-quote in formatted output.
3184 This is not supported by mandoc.
3185 .El
3186 .Sh SEE ALSO
3187 .Xr man 1 ,
3188 .Xr mandoc 1 ,
3189 .Xr eqn 5 ,
3190 .Xr man 5 ,
3191 .Xr mandoc_char 5 ,
3192 .Xr roff 5 ,
3193 .Xr tbl 5
3194 .Sh HISTORY
3195 The
3196 .Nm
3197 language first appeared as a troff macro package in
3198 .Bx 4.4 .
3199 It was later significantly updated by Werner Lemberg and Ruslan Ermilov
3200 in groff-1.17.
3201 The standalone implementation that is part of the
3202 .Xr mandoc 1
3203 utility written by Kristaps Dzonsons appeared in
3204 .Ox 4.6 .
3205 in July, 2014.
3206 .Sh AUTHORS
3207 The
3208 .Nm
3209 reference was written by
3210 .An Kristaps Dzonsons ,
3211 .Mt kristaps@bsd.lv .
```

```
**********************************************************
    4186 Thu Jul 17 00:50:42 2014
new/usr/src/man/man7d/cpqary3.7d
manpage lint.
**********************************************************
.\"
.\" This file and its contents are supplied under the terms of the
.\" Common Development and Distribution License ("CDDL"), version 1.0.
.\" You may only use this file in accordance with the terms of version
.\" 1.0 of the CDDL.
.\"
.\" A full copy of the text of the CDDL should have accompanied this
.\" source.  A copy of the CDDL is also available via the Internet at
.\" http://www.illumos.org/license/CDDL.
.\"
.\"
.\" Copyright (C) 2013 Hewlett-Packard Development Company, L.P.
.\"
.TH CPQARY3 7D Aug 26, 2013"
.SH NAME
.LP
cpqary3 - provides disk and SCSI tape support for HP Smart Array controllers
.LP
.SH DESCRIPTION
.LP
The cpqary3 module provides low-level interface routines between the common
disk I/O subsystem and the HP SMART Array controllers. The cpqary3 driver
provides disk and SCSI tape support for the HP Smart Array controllers.
.LP
Please refer to the cpqary3 release notes, for the supported HP Smart Array
Controllers and Storage boxes.
.LP
Each controller should be the sole initiator on a SCSI bus. Auto
configuration code determines if the adapter is present at the Configured
address and what types of devices are attached to it.
.SH CONFIGURATION
Use the Array Configuration Utility to configure the controllers. Each
controller can support up to 32 logical volumes. In addition, each controller
supports up to a maximum of 28 connected SCSI tape drives.
With 1.90 and later versions of cpqary3 driver, HP Smart Array SAS controllers,
having Firmware Revision 5.10 or later, will support 64 logical drives. This
firmware also supports Dual Domian Multipath configurations.
.LP
The driver attempts to initialize itself in accordance with the information
found in the configuration file, /kernel/drv/cpqary3.conf.
.LP
New component - hmaeventd which logs the storage events onto console and to the
Integrated Management Log is made a part of HPQhma 5.0.0 package, which is not
part of the operating system.  Therefore, by default, notify on event
functionality is disabled in the driver from 2.1.0 onwards.  Storage event
logging may be enabled in the driver by modifying cpqary3.conf to set the
cpqary3_noe property to "on".  Modification of driver properties requires
reloading the driver, which in most cases will occur only following a reboot of
the system.
.LP
The target driver's configuration file shall need entries if support is needed
for targets numbering greater than the default number of targets supported by
the corresponding target driver.
.LP
By default, entries for SCSI target numbers 0 to 15 are present in sd.conf.
Entries for target numbers 16 and above must be added to the \&'scsi' class in
sd.conf to support additional corresponding logical volumes.
.LP
If SCSI tape drives are connected to the supported controllers, entries for
target IDs from 33 to 33+N must be added in the /kernel/drv/st.conf file under
\&'scsi' class, where N is the total number of SCSI tape drives connected to the
```



```
controller with largest number of tape drives connected to it, in the existing
configuration. For example, two supported controller, c1 and c2 are present in
the system. If controller c1 has two (2) tape drives and controller c2 has five
(5) tape drives connected, then entries for target IDs 33 thru 38 are required
under 'scsi' class in /kernel/drv/st.conf file. The maximum number of tape
drives that can be connected to a controller is 28. With 1.90 and later versions
of cpqary3 driver, if tape drives are connected to Smart Array SAS controllers,
then target ID entries for tape drives from 65 to 65+N must be added in
/kernel/drv/st.conf file under the \&'scsi' class.
.SH FILES
.PD 0
.TP 25
.B /kernel/drv/cpqary3.conf
- configuration file for CPQary3
.PD
.SH "SEE ALSO"
.BR driver.conf (4),
.BR sd (7D),
.BR st (7D)
.LP
.SH NOTES
.LP
The Smart Array controllers supported by the current version of the
cpqary3 driver do not support 'format unit' SCSI command. Hence, selecting
\&'format' option under 'format' utility main menu is not supported. In addition
the 'repair' option under 'format' utility main menu is not supported as this
operation is not applicable to Logical volumes connected to the supported Smart
Array controllers.
```

```
***********************************************************
    2008 Thu Jul 17 00:50:42 2014
new/usr/src/man/man7d/nv_sata.7d
manpage lint.
***********************************************************
'\" te
.\" Copyright (c) 2008, Sun Microsystems, Inc.  All Rights Reserved
.\" Copyright 2011 Nexenta Systems, 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 NV_SATA 7D "Sep 25, 2011"
.TH nv_sata 7D "25 Sep 2011"
.SH NAME
nv_sata \- NVIDIA CK804/MCP04/MCP51/MCP55/MCP61 SATA controller driver
.SH SYNOPSIS
.LP
.nf
\fBsata@unit-address\fR
.fi

.SH DESCRIPTION
.sp
.LP
The \fBnv_sata\fR driver is a SATA HBA driver that supports NVIDIA CK804/MCP04
and MCP51/MCP55/MCP61 SATA HBA controllers. Note that while these controllers
support standard SATA features including SATA-II drives, NCQ, hotplug and ATAPI
drives, the driver currently does not support NCQ features.
.SH CONFIGURATION
.sp
.LP
The \fBnv_sata\fR module contains no user configurable parameters.
.SH FILES
.sp
.ne 2
.na
\fB\fB/kernel/drv/nv_sata\fR\fR
.ad
.sp .6
.RS 4n
32-bit \fBELF\fR kernel module (x86).
.RE

.sp
.ne 2
.na
\fB\fB/kernel/drv/amd64/nv_sata\fR\fR
.ad
.sp .6
.RS 4n
64-bit \fBELF\fR kernel module (x86).
.RE

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

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
```



```
Architecture    x86
.TE

.SH SEE ALSO
.sp
.LP
\fBcfgadm\fR(1M), \fBcfgadm_sata\fR(1M), \fBprtconf\fR(1M), \fBsata\fR(7D),
\fBsd\fR(7D)
.sp
.LP
\fIWriting Device Drivers\fR
```

```
**********************************************************
    4981 Thu Jul 17 00:50:42 2014
new/usr/src/man/man7d/pcn.7d
manpage lint.
**********************************************************
'\" te
.\" Copyright 2011 Jason King.  All rights reserved.
.\" Copyright (c) 2001-2007 by Garrett D'Amore.
.\" Redistribution and use in source and binary forms, with or without
.\" modification, are permitted provided that the following conditions are met:
.\" 1. Redistributions of source code must retain the above copyright notice,
.\"    this list of conditions and the following disclaimer.
.\" 2. Redistributions in binary form must reproduce the above copyright notice,
.\"    this list of conditions and the following disclaimer in the documentation
.\"    and/or other materials provided with the distribution.
.\" 3. Neither the name of the author nor the names of any co-contributors may
.\"    be used to endorse or promote products derived from this software
.\"    without specific prior written permission.
.\" THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER AND CONTRIBUTORS
.\" ''AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED
.\" TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
.\" PURPOSE ARE DISCLAIMED.  IN NO EVENT SHALL THE COPYRIGHT HOLDER OR
.\" CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
.\" EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
.\" PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS;
.\" OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,
.\" WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR
.\" OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF
.\" ADVISED OF THE POSSIBILITY OF SUCH DAMAGE
.\" Portions Copyright (c) 2007 by Sun Microsystems, Inc.  All Rights Reserved.

.TH "PCN" "7D" "Sep 16, 2011"
.
.SH "NAME"
\fBpcn\fR \- PCnet Ethernet device driver
.SH "SYNOPSIS"
.LP
.nf
\fB/dev/pcn\fR
.fi

.SH "DESCRIPTION"
.sp
.LP
The \fBpcn\fR driver is a multi\-threaded, loadable, clonable GLDv3\-based
STREAMS driver supporting the Data Link Provider Interface \fBdlpi\fR(7P) for
the AMD PCnet family of Ethernet controllers\.
.SH "APPLICATION PROGRAMMING INTERFACE"
The \fBpcn\fR driver can be used as either a "style 1" or a "style 2" Data Link
Service Provider\. Physical points of attachment (PPAs) are interpreted as the
instance number of the \fBpcn\fR controller as assigned by the
instance number of the \fBpcn\fR controller as assigned by the Illumos
operating environment\.
.sp
.LP
The values returned by the driver in the \fBDL_INFO_ACK\fR response are:
.RS +4
.TP
.ie t \(bu
.el o
Maximum SDU is 1500\.
.RE
.RS +4
.TP
.ie t \(bu
.el o
```



```
Minimum SDU is 0\.
.RE
.RS +4
.TP
.ie t \(bu
.el o
The dlsap address length is 8\.
.RE
.RS +4
.TP
.ie t \(bu
.el o
MAC type is \fBDL_ETHER\fR\.
.RE
.RS +4
.TP
.ie t \(bu
.el o
SAP length is \-2\. The 6\-byte physical address is immediately followed by a
2\-byte SAP\.
.RE
.RS +4
.TP
.ie t \(bu
.el o
Service mode is \fBDL_CLDLS\fR\.
.RE
.RS +4
.TP
.ie t \(bu
.el o
The broadcast address is the 6\-byte Ethernet broadcast address
(\fBff:ff:ff:ff:ff:ff\fR)\.
.SH "CONFIGURATION"
.sp
.LP
The \fBpcn\fR driver performs auto-negotiation to select the link speed and
mode\.  Link sped may be 100Mbps full\-duplex, 100Mbps half\-duplex,
10Mbps full\-duplex, or 10Mbps half\-duplex, depending on the hardware
adapter type\.  See the \fIIEEE802.3\fR standard for more information\.
.sp
.LP
The capabilities advertised by the \fBpcn\fR device can be set using
\fBdladm\fR(1m)\.  The driver supports a number of parameters whose names
being with \fBen_\fR (see below)\.  Each of these parameters contains a
boolean value that determines if the devices advertises that mode of
operations\.  The \fBadv_autoneg_cap\fR parameter controls whether
auto-negotioation is performed\.  If \fBadv_autoneg_cap\fR is set to 0, the
driver forces the mode of operation selected by the first non-zero
parameter in priority order as shown below:
.sp
.in +2
.nf
                              (highest priority/greatest throughput)
        en_100fdx_cap           100Mbps full duplex
        en_10fdx_cap            10Mbps full duplex
                             (lowest priority/least throughput)
.fi
.in -2

.sp
.LP
All capabilities default to enabled\.  Note that changing any capability
parameter causes te link to go down while the link partners renegotiate
the link speed/duplex using the newly changed capabilities\.
.SH "ATTRIBUTES"
```

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

.sp
.TS
box;
c | c
l | l .
ATTRIBUTE TYPE  ATTRIBUTE VALUE
_
Architecture    x86
_
Interface Stability     Committed
.TE

.SH "FILES"
.sp
.ne 2
.na
\fB\fB/dev/pcn\fR\fR
.ad
.sp .6
.RS 4n
Special character device\.
.RE

.sp
.ne 2
.na
\fB\fB/kernel/drv/pcn\fR\fR
.ad
.sp 6
.RS 4n
32\-bit driver binary\.
.RE

.sp
.ne 2
.na
\fB\fB/kernel/drv/amd64/pcn\fR\fR
.ad
.sp .6
.RS 4n
64\-bit driver binary (x86)\.
.RE

.SH "SEE ALSO"
.sp
.LP
\fBdlpi\fR(1m), \fBattributes\fR(5), \fBstreamio\fR(7I), \fBdlpi\fR(7p)
.sp
.LP
\fIIEEE 802.3\fR \(em Institute of Electrical and Electronics Engineers, 2002
```

```
**********************************************************
   11098 Thu Jul 17 00:50:42 2014
new/usr/src/man/man9f/ddi_dmae.9f
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
    7939 Thu Jul 17 00:50:42 2014
new/usr/src/man/man9f/rwlock.9f
manpage lint.
**********************************************************
   1 '\" te
   2 .\"  Copyright (c) 2006 Sun Microsystems, Inc. All Rights Reserved
   3 .\" Copyright (c) 2013, Joyent, 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 RWLOCK 9F "Sep 19, 2013"
   8 .SH NAME
   9 rwlock, rw_init, rw_destroy, rw_enter, rw_exit, rw_tryenter, rw_downgrade,
  10 rw_tryupgrade, rw_read_locked \- readers/writer lock functions
  11 .SH SYNOPSIS
  12 .LP
  13 .nf
  14 #include <sys/ksynch.h>



  18 \fBvoid\fR \fBrw_init\fR(\fBkrwlock_t *\fR\fIrwlp\fR, \fBchar *\fR\fIname\fR, \f
  19 .fi

  21 .LP
  22 .nf
  23 \fBvoid\fR \fBrw_destroy\fR(\fBkrwlock_t *\fR\fIrwlp\fR);
  24 .fi

  26 .LP
  27 .nf
  28 \fBvoid\fR \fBrw_enter\fR(\fBkrwlock_t *\fR\fIrwlp\fR, \fBkrw_t\fR \fIenter_type
  29 .fi

  31 .LP
  32 .nf
  33 \fBvoid\fR \fBrw_exit\fR(\fBkrwlock_t *\fR\fIrwlp\fR);
  34 .fi

  36 .LP
  37 .nf
  38 \fBint\fR \fBrw_tryenter\fR(\fBkrwlock_t *\fR\fIrwlp\fR, \fBkrw_t\fR \fIenter_ty
  39 .fi

  41 .LP
  42 .nf
  43 \fBvoid\fR \fBrw_downgrade\fR(\fBkrwlock_t *\fR\fIrwlp\fR);
  44 .fi

  46 .LP
  47 .nf
  48 \fBint\fR \fBrw_tryupgrade\fR(\fBkrwlock_t *\fR\fIrwlp\fR);
  49 .fi

  51 .LP
  52 .nf
  53 \fBint\fR \fBrw_read_locked\fR(\fBkrwlock_t *\fR\fIrwlp\fR);
  54 .fi

  56 .SH INTERFACE LEVEL
  57 .sp
  58 .LP
  59 Solaris DDI specific (Solaris DDI).
  60 .SH PARAMETERS
  61 .sp
```



```
  62 .ne 2
  63 .na
  64 \fB\fIrwlp\fR\fR
  65 .ad
  66 .RS 14n
  67 Pointer to a \fBkrwlock_t\fR readers/writer lock.
  68 .RE

  70 .sp
  71 .ne 2
  72 .na
  73 \fB\fIname\fR\fR
  74 .ad
  75 .RS 14n
  76 Descriptive string. This is obsolete and should be \fBNULL\fR. (Non-null
  77 strings are legal, but they're a waste of kernel memory.)
  78 .RE

  80 .sp
  81 .ne 2
  82 .na
  83 \fB\fItype\fR\fR
  84 .ad
  85 .RS 14n
  86 Type of readers/writer lock.
  87 .RE

  89 .sp
  90 .ne 2
  91 .na
  92 \fB\fIarg\fR\fR
  93 .ad
  94 .RS 14n
  95 Type-specific argument for initialization function.
  96 .RE

  98 .sp
  99 .ne 2
 100 .na
 101 \fB\fIenter_type\fR\fR
 102 .ad
 103 .RS 14n
 104 One of the values \fBRW_WRITER\fR, \fBRW_READER\fR or
 105 \fBRW_READER_STARVEWRITER\fR, indicating whether the
 106 lock is to be acquired exclusively (\fBRW_WRITER\fR), non-exclusively
 107 (\fBRW_READER\fR) or non-exclusively without regard to any threads
 108 that may be blocked on exclusive access (\fBRW_READER_STARVEWRITER\fR).
 109 .RE

 111 .SH DESCRIPTION
 112 .sp
 113 .LP
 114 A multiple-readers, single-writer lock is represented by the \fBkrwlock_t\fR
 115 data type. This type of lock will allow many threads to have simultaneous
 116 read-only access to an object. Only one thread may have write access at any one
 117 time. An object that is searched more frequently than it is changed is a good
 118 candidate for a readers/writer lock.
 119 .sp
 120 .LP
 121 Readers/writer locks are slightly more expensive than mutex locks, and the
 122 advantage of multiple read access may not occur if the lock will only be held
 123 for a short time.
 124 .sp
 125 .LP
 126 The \fBrw_init()\fR function initializes a readers/writer lock. It is an error
 127 to initialize a lock more than once. The \fItype\fR argument should be set to
```

```
\fBRW_DRIVER\fR. If the lock is used by the interrupt handler, the
type-specific argument, \fIarg\fR, should be the interrupt priority returned
from \fBddi_intr_get_pri\fR(9F) or \fBddi_intr_get_softint_pri\fR(9F). Note
that arg should be the value of the interrupt priority cast by calling the
\fBDDI_INTR_PRI\fR macro. If the lock is not used by any interrupt handler, the
argument should be \fINULL.\fR
.sp
.LP
The \fBrw_destroy()\fR function releases any resources that might have been
allocated by \fBrw_init()\fR. It should be called before freeing the memory
containing the lock. The lock must not be held by any thread when it is
destroyed.
.sp
.LP
The \fBrw_enter()\fR function acquires the lock, and blocks if necessary. If
\fIenter_type\fR is \fBRW_WRITER\fR, the caller blocks if any thread holds
the lock. If \fIenter_type\fR is \fBRW_READER\fR, the caller blocks if there
is a writer or a thread attempting to enter for writing. If \fIenter_type\fR
is \fBRW_READER_STARVEWRITER\fR, the caller blocks only if there is a writer;
if the lock is held for reading and a thread is blocked attempting to enter
for writing, the caller will acquire the lock as a reader instead of
blocking on the pending writer.

.sp
.LP
NOTE: It is a programming error for any thread to acquire an rwlock as
\fBRW_READER\fR that it already holds. Doing so can deadlock the system: if
thread R acquires the lock as \fBRW_READER\fR, then thread W tries to acquire
the lock as a writer, W will set write-wanted and block. When R tries to get
its second read hold on the lock, it will honor the write-wanted bit and block
waiting for W; but W cannot run until R drops the lock. Thus threads R and W
deadlock.  To opt out of this behavior -- that is, to safely allow a lock to
be grabbed recursively as a reader -- the lock should be acquired as
\fBRW_READER_STARVEWRITER\fR, which will allow R to get its second read hold
without regard for the write-wanted bit set by W.  Note that the
\fBRW_READER_STARVEWRITER\fR behavior will starve writers in the presence of
infinite readers; it should be used with care, and only where the default
\fBRW_READER\fR behavior is unacceptable.
.sp
.LP
The \fBrw_exit()\fR function releases the lock and may wake up one or more
threads waiting on the lock.
.sp
.LP
The \fBrw_tryenter()\fR function attempts to enter the lock, like
\fBrw_enter()\fR, but never blocks. It returns a non-zero value if the lock was
successfully entered, and zero otherwise.
.sp
.LP
A thread that holds the lock exclusively (entered with \fBRW_WRITER\fR), may
call \fBrw_downgrade()\fR to convert to holding the lock non-exclusively (as if
entered with \fBRW_READER\fR). One or more waiting readers may be unblocked.
.sp
.LP
The \fBrw_tryupgrade()\fR function can be called by a thread that holds the
lock for reading to attempt to convert to holding it for writing. This upgrade
can only succeed if no other thread is holding the lock and no other thread is
blocked waiting to acquire the lock for writing.
.sp
.LP
The \fBrw_read_locked()\fR function returns non-zero if the calling thread
holds the lock for read, and zero if the caller holds the lock for write. The
caller must hold the lock. The system may panic if \fBrw_read_locked()\fR is
called for a lock that isn't held by the caller.
.SH RETURN VALUES
.sp
```



```
.ne 2
.na
\fB\fB0\fR\fR
.ad
.RS 12n
\fBrw_tryenter()\fR could not obtain the lock without blocking.
.RE

.sp
.ne 2
.na
\fB\fB0\fR\fR
.ad
.RS 12n
\fBrw_tryupgrade()\fR was unable to perform the upgrade because of other
threads holding or waiting to hold the lock.
.RE

.sp
.ne 2
.na
\fB\fB0\fR\fR
.ad
.RS 12n
\fBrw_read_locked()\fR returns \fB0\fR if the lock is held by the caller for
write.
.RE

.sp
.ne 2
.na
\fB\fBnon-zero\fR\fR
.ad
.RS 12n
from \fBrw_read_locked()\fR if the lock is held by the caller for read.
.RE

.sp
.ne 2
.na
\fB\fBnon-zero\fR\fR
.ad
.RS 12n
successful return from \fBrw_tryenter()\fR or \fBrw_tryupgrade()\fR.
.RE

.SH CONTEXT
.sp
.LP
These functions can be called from user, interrupt, or kernel context, except
for \fBrw_init()\fR and \fBrw_destroy()\fR, which can be called from user
context only.
.SH SEE ALSO
.sp
.LP
\fBcondvar\fR(9F), \fBddi_intr_alloc\fR(9F), \fBddi_intr_add_handler\fR(9F),
\fBddi_intr_get_pri\fR(9F), \fBddi_intr_get_softint_pri\fR(9F),
\fBmutex\fR(9F), \fBsemaphore\fR(9F)
.sp
.LP
\fIWriting Device Drivers\fR
.SH NOTES
.sp
.LP
Compiling with \fB_LOCKTEST\fR or \fB_MPSTATS\fR defined no longer has any
effect. To gather lock statistics, see \fBlockstat\fR(1M).
```

```
**********************************************************
    7145 Thu Jul 17 00:50:43 2014
new/usr/src/man/man9f/scsi_hba_attach_setup.9f
manpage lint.
**********************************************************
______unchanged_portion_omitted_
```

```
**********************************************************
    1425 Thu Jul 17 00:50:43 2014
new/usr/src/tools/onbld/Checks/Makefile
manpage lint.
**********************************************************
```

```
   1 #
   2 # CDDL HEADER START
   3 #
   4 # The contents of this file are subject to the terms of the
   5 # Common Development and Distribution License (the "License").
   6 # You may not use this file except in compliance with the License.
   7 #
   8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
   9 # or http://www.opensolaris.org/os/licensing.
  10 # See the License for the specific language governing permissions
  11 # and limitations under the License.
  12 #
  13 # When distributing Covered Code, include this CDDL HEADER in each
  14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
  15 # If applicable, add the following below this CDDL HEADER, with the
  16 # fields enclosed by brackets "[]" replaced with your own identifying
  17 # information: Portions Copyright [yyyy] [name of copyright owner]
  18 #
  19 # CDDL HEADER END
  20 #

  22 #
  23 # Copyright (c) 2008, 2010, Oracle and/or its affiliates. All rights reserved.
  24 #

  26 # Copyright 2010, Richard Lowe
  27 # Copyright 2014 Garrett D'Amore <garrett@damore.org>

  29 include $(SRC)/Makefile.master
  30 include ../../Makefile.tools

  32 PYSRCS = \
  33         CStyle.py       \
  34         Cddl.py         \
  35         CmtBlk.py       \
  36         Comments.py     \
  37         Copyright.py    \
  38         DbLookups.py    \
  39         HdrChk.py       \
  40         JStyle.py       \
  41         Keywords.py     \
  42         ManLint.py      \
  43         Mapfile.py      \
  44         ProcessCheck.py \
  45         __init__.py

  47 PYOBJS =        $(PYSRCS:%.py=%.pyc)
  48 PYTOPDIR =      $(ROOTONBLDLIB)
  49 PYMODDIR =      onbld/Checks

  51 include ../../Makefile.python

  53 all: $(PYVERSOJBS)

  55 install: all $(ROOTPYFILES)

  57 clean:

  59 clobber: clean pyclobber
```

****
1631 Thu Jul 17 00:50:43 2014
new/usr/src/tools/onbld/Checks/ManLint.py
manpage lint.
****

```
  1 #
  2 # CDDL HEADER START
  3 #
  4 # The contents of this file are subject to the terms of the
  5 # Common Development and Distribution License (the "License").
  6 # You may not use this file except in compliance with the License.
  7 #
  8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
  9 # or http://www.opensolaris.org/os/licensing.
 10 # See the License for the specific language governing permissions
 11 # and limitations under the License.
 12 #
 13 # When distributing Covered Code, include this CDDL HEADER in each
 14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
 15 # If applicable, add the following below this CDDL HEADER, with the
 16 # fields enclosed by brackets "[]" replaced with your own identifying
 17 # information: Portions Copyright [yyyy] [name of copyright owner]
 18 #
 19 # CDDL HEADER END
 20 #

 22 #
 23 # Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
 24 # Use is subject to license terms.
 25 #
 26 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
 27 #

 29 #
 30 # ManLint, wrap the mandoc lint tool in a pythonic API
 31 #

 33 import sys
 34 from onbld.Checks.ProcessCheck import processcheck

 36 def manlint(fh, filename=None, output=sys.stderr, **opts):
 37         opttrans = { 'picky': None }

 39         for x in filter(lambda x: x not in opttrans, opts):
 40                 raise TypeError('mandoc() got an unexpected keyword '
 41                                 'argument %s' % x)

 43         options = [opttrans[x] for x in opts if opts[x] and opttrans[x]]
 44         options.append('-Tlint')

 46         if not filename:
 47                 filename = fh.name

 49         ret, tmpfile = processcheck('mandoc', options, fh, output)

 51         if tmpfile:
 52                 for line in tmpfile:
 53                         line = line.replace('<stdin>', filename)
 54                         output.write(line)

 56                 tmpfile.close()
 57         return ret
```

```
**********************************************************
    1193 Thu Jul 17 00:50:43 2014
new/usr/src/tools/onbld/Checks/__init__.py
manpage lint.
**********************************************************
#! /usr/bin/python
#
# 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 2009 Sun Microsystems, Inc.  All rights reserved.
# Use is subject to license terms.
#

# Copyright 2010, Richard Lowe
# Copyright 2014 Garrett D'Amore <garrett@damore.org>

#
# The 'checks' package contains various checks that may be run
#

__all__ = [
        'Cddl',
        'Comments',
        'Copyright',
        'CStyle',
        'HdrChk',
        'JStyle',
        'Keywords',
        'ManLint',
        'Mapfile']
```

```
**********************************************************
   51625 Thu Jul 17 00:50:43 2014
new/usr/src/tools/onbld/hgext/cdm.py
manpage lint.
**********************************************************
```

```
#
#  This program is free software; you can redistribute it and/or modify
#  it under the terms of the GNU General Public License version 2
#  as published by the Free Software Foundation.
#
#  This program is distributed in the hope that it will be useful,
#  but WITHOUT ANY WARRANTY; without even the implied warranty of
#  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#  GNU General Public License for more details.
#
#  You should have received a copy of the GNU General Public License
#  along with this program; if not, write to the Free Software
#  Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
#

#
# Copyright (c) 2008, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2008, 2011 Richard Lowe
# Copyright 2014 Garrett D'Amore <garrett@damore.org>
#

'''OpenSolaris extensions to Mercurial

    This extension contains a number of commands to help you work with
the OpenSolaris consolidations.  It provides commands to check your
changes against the various style rules used for OpenSolaris, to
backup and restore your changes, to generate code reviews, and to
prepare your changes for integration.


The Parent

    To provide a uniform notion of parent workspace regardless of
filesystem-based access, Cadmium uses the highest numbered changeset
on the current branch that is also in the parent workspace to
represent the parent workspace.


The Active List

    Many Cadmium commands operate on the active list, the set of
files ('active files') you have changed in this workspace in relation
to its parent workspace, and the metadata (commentary, primarily)
associated with those changes.


NOT Files

    Many of Cadmium's commands to check that your work obeys the
various stylistic rules of the OpenSolaris consolidations (such as
those run by 'hg nits') allow files to be excluded from this checking
by means of NOT files kept in the .hg/cdm/ directory of the Mercurial
repository for one-time exceptions, and in the exception_lists
directory at the repository root for permanent exceptions.  (For ON,
these would mean one in $CODEMGR_WS and one in
$CODEMGR_WS/usr/closed).

    These files are in the same format as the Mercurial hgignore
file, a description of which is available in the hgignore(5) manual
page.
```



```
Common Tasks

  - Show diffs relative to parent workspace              - pdiffs
  - Check source style rules                             - nits
  - Run pre-integration checks                           - pbchk
  - Collapse all your changes into a single changeset    - recommit
'''

import atexit, os, re, sys, stat, termios


#
# Adjust the load path based on the location of cdm.py and the version
# of python into which it is being loaded.  This assumes the normal
# onbld directory structure, where cdm.py is in
# lib/python(version)?/onbld/hgext/.  If that changes so too must
# this.
#
# This and the case below are not equivalent.  In this case we may be
# loading a cdm.py in python2.X/ via the lib/python/ symlink but need
# python2.Y in sys.path.
#
sys.path.insert(1, os.path.join(os.path.dirname(__file__), "..", "..", "..",
                                "python%d.%d" % sys.version_info[:2]))

#
# Add the relative path from cdm.py to usr/src/tools to the load path,
# such that a cdm.py loaded from the source tree uses the modules also
# within the source tree.
#
sys.path.insert(2, os.path.join(os.path.dirname(__file__), "..", ".."))

from onbld.Scm import Version
from mercurial import util

try:
    Version.check_version()
except Version.VersionMismatch, badversion:
    raise util.Abort("Version Mismatch:\n %s\n" % badversion)

from mercurial import cmdutil, ignore, node, patch

from onbld.Scm.WorkSpace import WorkSpace, WorkList
from onbld.Scm.Backup import CdmBackup
from onbld.Checks import Cddl, Comments, Copyright, CStyle, HdrChk
from onbld.Checks import JStyle, Keywords, ManLint, Mapfile
from onbld.Checks import JStyle, Keywords, Mapfile


def yes_no(ui, msg, default):
    if default:
        prompt = ' [Y/n]:'
        defanswer = 'y'
    else:
        prompt = ' [y/N]:'
        defanswer = 'n'

    if Version.at_least("1.4"):
        index = ui.promptchoice(msg + prompt, ['&yes', '&no'],
                                default=['y', 'n'].index(defanswer))
        resp = ('y', 'n')[index]
    else:
        resp = ui.prompt(msg + prompt, ['&yes', '&no'], default=defanswer)

    return resp[0] in ('Y', 'y')
```

```
129 def buildfilelist(ws, parent, files):
130     '''Build a list of files in which we're interested.
131
132     If no files are specified take files from the active list relative
133     to 'parent'.
134
135     Return a list of 2-tuples the first element being a path relative
136     to the current directory and the second an entry from the active
137     list, or None if an explicit file list was given.'''
138
139     if files:
140         return [(path, None) for path in sorted(files)]
141     else:
142         active = ws.active(parent=parent)
143         return [(ws.filepath(e.name), e) for e in sorted(active)]
144 buildfilelist = util.cachefunc(buildfilelist)
145
146
147 def not_check(repo, cmd):
148     '''return a function which returns boolean indicating whether a file
149     should be skipped for CMD.'''
150
151     #
152     # The ignore routines need a canonical path to the file (relative to the
153     # repo root), whereas the check commands get paths relative to the cwd.
154     #
155     # Wrap our argument such that the path is canonified before it is checked.
156     #
157     def canonified_check(ignfunc):
158         def f(path):
159             cpath = util.canonpath(repo.root, repo.getcwd(), path)
160             return ignfunc(cpath)
161         return f
162
163     ignorefiles = []
164
165     for f in [repo.join('cdm/%s.NOT' % cmd),
166               repo.wjoin('exception_lists/%s' % cmd)]:
167         if os.path.exists(f):
168             ignorefiles.append(f)
169
170     if ignorefiles:
171         ign = ignore.ignore(repo.root, ignorefiles, repo.ui.warn)
172         return canonified_check(ign)
173     else:
174         return util.never
175
176
177 def abort_if_dirty(ws):
178     '''Abort if the workspace has uncommitted changes, merges,
179     branches, or has Mq patches applied'''
180
181     if ws.modified():
182         raise util.Abort('workspace has uncommitted changes')
183     if ws.merged():
184         raise util.Abort('workspace contains uncommitted merge')
185     if ws.branched():
186         raise util.Abort('workspace contains uncommitted branch')
187     if ws.mq_applied():
188         raise util.Abort('workspace has Mq patches applied')
189
190
191 #
192 # Adding a reference to WorkSpace from a repo causes a circular reference
```



```
193 # repo <-> WorkSpace.
194 #
195 # This prevents repo, WorkSpace and members thereof from being garbage
196 # collected.  Since transactions are aborted when the transaction object
197 # is collected, and localrepo holds a reference to the most recently created
198 # transaction, this prevents transactions from cleanly aborting.
199 #
200 # Instead, we hold the repo->WorkSpace association in a dictionary, breaking
201 # that dependence.
202 #
203 wslist = {}
204
205
206 def reposetup(ui, repo):
207     if repo.local() and repo not in wslist:
208         wslist[repo] = WorkSpace(repo)
209
210         if ui.interactive() and sys.stdin.isatty():
211             ui.setconfig('hooks', 'preoutgoing.cdm_pbconfirm',
212                          'python:hgext_cdm.pbconfirm')
213
214
215 def pbconfirm(ui, repo, hooktype, source):
216     def wrapper(settings=None):
217         termios.tcsetattr(sys.stdin.fileno(), termios.TCSANOW, settings)
218
219     if source == 'push':
220         if not yes_no(ui, "Are you sure you wish to push?", False):
221             return 1
222         else:
223             settings = termios.tcgetattr(sys.stdin.fileno())
224             orig = list(settings)
225             atexit.register(wrapper, orig)
226             settings[3] = settings[3] & (~termios.ISIG) # c_lflag
227             termios.tcsetattr(sys.stdin.fileno(), termios.TCSANOW, settings)
228
229
230 def cdm_pdiffs(ui, repo, *pats, **opts):
231     '''diff workspace against its parent
232
233     Show differences between this workspace and its parent workspace
234     in the same manner as 'hg diff'.
235
236     For a description of the changeset used to represent the parent
237     workspace, see The Parent in the extension documentation ('hg help
238     cdm').
239     '''
240
241     act = wslist[repo].active(opts.get('parent'))
242     if not act.revs:
243         return
244
245     #
246     # If no patterns were specified, either explicitly or via -I or -X
247     # use the active list files to avoid a workspace walk.
248     #
249     if pats or opts.get('include') or opts.get('exclude'):
250         matchfunc = wslist[repo].matcher(pats=pats, opts=opts)
251     else:
252         matchfunc = wslist[repo].matcher(files=act.files())
253
254     opts = patch.diffopts(ui, opts)
255     diffs = wslist[repo].diff(act.parenttip.node(), act.localtip.node(),
256                               match=matchfunc, opts=opts)
257     if diffs:
258         ui.write(diffs)
```

```
261 def cdm_list(ui, repo, **opts):
262     '''list active files (those changed in this workspace)
263
264     Display a list of files changed in this workspace as compared to
265     its parent workspace.
266
267     File names are displayed one-per line, grouped by manner in which
268     they changed (added, modified, removed).  Information about
269     renames or copies is output in parentheses following the file
270     name.
271
272     For a description of the changeset used to represent the parent
273     workspace, see The Parent in the extension documentation ('hg help
274     cdm').
275
276     Output can be filtered by change type with --added, --modified,
277     and --removed.  By default, all files are shown.
278     '''
279
280     act = wslist[repo].active(opts['parent'])
281     wanted = set(x for x in ('added', 'modified', 'removed') if opts[x])
282     changes = {}
283
284     for entry in act:
285         if wanted and (entry.change not in wanted):
286             continue
287
288         if entry.change not in changes:
289             changes[entry.change] = []
290         changes[entry.change].append(entry)
291
292     for change in sorted(changes.keys()):
293         ui.write(change + ':\n')
294
295         for entry in sorted(changes[change]):
296             if entry.is_renamed():
297                 ui.write('\t%s (renamed from %s)\n' % (entry.name,
298                                                       entry.parentname))
299             elif entry.is_copied():
300                 ui.write('\t%s (copied from %s)\n' % (entry.name,
301                                                       entry.parentname))
302             else:
303                 ui.write('\t%s\n' % entry.name)
304
305
306 def cdm_bugs(ui, repo, parent=None):
307     '''show all bug IDs referenced in changeset comments'''
308
309     act = wslist[repo].active(parent)
310
311     for elt in set(filter(Comments.isBug, act.comments())):
312         ui.write(elt + '\n')
313
314
315 def cdm_comments(ui, repo, parent=None):
316     '''show changeset commentary for all active changesets'''
317     act = wslist[repo].active(parent)
318
319     for elt in act.comments():
320         ui.write(elt + '\n')
321
322
323 def cdm_renamed(ui, repo, parent=None):
324     '''show renamed active files
```



```
325
326     Renamed files are shown in the format::
327
328        new-name old-name
329
330     One pair per-line.
331     '''
332
333     act = wslist[repo].active(parent)
334
335     for entry in sorted(filter(lambda x: x.is_renamed(), act)):
336         ui.write('%s %s\n' % (entry.name, entry.parentname))
337
338
339 def cdm_comchk(ui, repo, **opts):
340     '''check active changeset comment formatting
341
342     Check that active changeset comments conform to O/N rules.
343
344     Each comment line must contain either one bug or ARC case ID
345     followed by its synopsis, or credit an external contributor.
346     '''
347
348     active = wslist[repo].active(opts.get('parent'))
349
350     ui.write('Comments check:\n')
351
352     check_db = not opts.get('nocheck')
353     return Comments.comchk(active.comments(), check_db=check_db, output=ui)
354
355
356 def cdm_cddlchk(ui, repo, *args, **opts):
357     '''check for a valid CDDL header comment in all active files.
358
359     Check active files for a valid Common Development and Distribution
360     License (CDDL) block comment.
361
362     Newly added files are checked for a copy of the CDDL header
363     comment.  Modified files are only checked if they contain what
364     appears to be an existing CDDL header comment.
365
366     Files can be excluded from this check using the cddlchk.NOT file.
367     See NOT Files in the extension documentation ('hg help cdm').
368     '''
369
370     filelist = buildfilelist(wslist[repo], opts.get('parent'), args)
371     exclude = not_check(repo, 'cddlchk')
372     lenient = True
373     ret = 0
374
375     ui.write('CDDL block check:\n')
376
377     for f, e in filelist:
378         if e and e.is_removed():
379             continue
380         elif (e or opts.get('honour_nots')) and exclude(f):
381             ui.status('Skipping %s...\n' % f)
382             continue
383         elif e and e.is_added():
384             lenient = False
385         else:
386             lenient = True
387
388         fh = open(f, 'r')
389         ret |= Cddl.cddlchk(fh, lenient=lenient, output=ui)
390         fh.close()
```

```
391     return ret


394 def cdm_manlintchk(ui, repo, *args, **opts):
395     '''check for mandoc lint

397     Check for man page formatting errors.

399     Files can be excluded from this check using the manlint.NOT
400     file.  See NOT Files in the extension documentation ('hg help
401     cdm').
402     '''

404     filelist = buildfilelist(wslist[repo], opts.get('parent'), args)
405     exclude = not_check(repo, 'manlint')
406     ret = 0

408     # Man pages are identified as having a suffix starting with a digit.
409     ManfileRE = re.compile(r'.*\.[0-9][a-z]*$', re.IGNORECASE)

411     ui.write('Man format check:\n')

413     for f, e in filelist:
414         if e and e.is_removed():
415             continue
416         elif (not ManfileRE.match(f)):
417             continue
418         elif (e or opts.get('honour_nots')) and exclude(f):
419             ui.status('Skipping %s...\n' % f)
420             continue

422         fh = open(f, 'r')
423         ret |= ManLint.manlint(fh, output=ui, picky=True)
424         fh.close()
425     return ret


428 def cdm_mapfilechk(ui, repo, *args, **opts):
429     '''check for a valid mapfile header block in active files

431     Check that all link-editor mapfiles contain the standard mapfile
432     header comment directing the reader to the document containing
433     Solaris object versioning rules (README.mapfile).

435     Files can be excluded from this check using the mapfilechk.NOT
436     file.  See NOT Files in the extension documentation ('hg help
437     cdm').
438     '''

440     filelist = buildfilelist(wslist[repo], opts.get('parent'), args)
441     exclude = not_check(repo, 'mapfilechk')
442     ret = 0

444     # We are interested in examining any file that has the following
445     # in its final path segment:
446     #     - Contains the word 'mapfile'
447     #     - Begins with 'map.'
448     #     - Ends with '.map'
449     # We don't want to match unless these things occur in final path segment
450     # because directory names with these strings don't indicate a mapfile.
451     # We also ignore files with suffixes that tell us that the files
452     # are not mapfiles.
453     MapfileRE = re.compile(r'.*((mapfile[^/]*)|(/map\.+[^/]*)|(\.map))$',
454         re.IGNORECASE)
455     NotMapSuffixRE = re.compile(r'.*\.[ch]$', re.IGNORECASE)
```



```
457     ui.write('Mapfile comment check:\n')

459     for f, e in filelist:
460         if e and e.is_removed():
461             continue
462         elif (not MapfileRE.match(f)) or NotMapSuffixRE.match(f):
463             continue
464         elif (e or opts.get('honour_nots')) and exclude(f):
465             ui.status('Skipping %s...\n' % f)
466             continue

468         fh = open(f, 'r')
469         ret |= Mapfile.mapfilechk(fh, output=ui)
470         fh.close()
471     return ret


474 def cdm_copyright(ui, repo, *args, **opts):
475     '''check each active file for a current and correct copyright notice

477     Check that all active files have a correctly formed copyright
478     notice containing the current year.

480     See the Non-Formatting Considerations section of the OpenSolaris
481     Developer's Reference for more info on the correct form of
482     copyright notice.
483     (http://hub.opensolaris.org/bin/view/Community+Group+on/devref_7#H723NonForm

485     Files can be excluded from this check using the copyright.NOT file.
486     See NOT Files in the extension documentation ('hg help cdm').
487     '''

489     filelist = buildfilelist(wslist[repo], opts.get('parent'), args)
490     exclude = not_check(repo, 'copyright')
491     ret = 0

493     ui.write('Copyright check:\n')

495     for f, e in filelist:
496         if e and e.is_removed():
497             continue
498         elif (e or opts.get('honour_nots')) and exclude(f):
499             ui.status('Skipping %s...\n' % f)
500             continue

502         fh = open(f, 'r')
503         ret |= Copyright.copyright(fh, output=ui)
504         fh.close()
505     return ret


508 def cdm_hdrchk(ui, repo, *args, **opts):
509     '''check active C header files conform to the O/N header rules

511     Check that any added or modified C header files conform to the O/N
512     header rules.

514     See the section 'HEADER STANDARDS' in the hdrchk(1) manual page
515     for more information on the rules for O/N header file formatting.

517     Files can be excluded from this check using the hdrchk.NOT file.
518     See NOT Files in the extension documentation ('hg help cdm').
519     '''

521     filelist = buildfilelist(wslist[repo], opts.get('parent'), args)
522     exclude = not_check(repo, 'hdrchk')
```

```
523     ret = 0
525     ui.write('Header format check:\n')
527     for f, e in filelist:
528         if e and e.is_removed():
529             continue
530         elif not f.endswith('.h'):
531             continue
532         elif (e or opts.get('honour_nots')) and exclude(f):
533             ui.status('Skipping %s...\n' % f)
534             continue
536         fh = open(f, 'r')
537         ret |= HdrChk.hdrchk(fh, lenient=True, output=ui)
538         fh.close()
539     return ret


542 def cdm_cstyle(ui, repo, *args, **opts):
543     '''check active C source files conform to the C Style Guide
545     Check that any added or modified C source file conform to the C
546     Style Guide.
548     See the C Style Guide for more information about correct C source
549     formatting.
550     (http://hub.opensolaris.org/bin/download/Community+Group+on/WebHome/cstyle.m
552     Files can be excluded from this check using the cstyle.NOT file.
553     See NOT Files in the extension documentation ('hg help cdm').
554     '''
556     filelist = buildfilelist(wslist[repo], opts.get('parent'), args)
557     exclude = not_check(repo, 'cstyle')
558     ret = 0
560     ui.write('C style check:\n')
562     for f, e in filelist:
563         if e and e.is_removed():
564             continue
565         elif not (f.endswith('.c') or f.endswith('.h')):
566             continue
567         elif (e or opts.get('honour_nots')) and exclude(f):
568             ui.status('Skipping %s...\n' % f)
569             continue
571         fh = open(f, 'r')
572         ret |= CStyle.cstyle(fh, output=ui,
573                              picky=True, check_posix_types=True,
574                              check_continuation=True)
575         fh.close()
576     return ret


579 def cdm_jstyle(ui, repo, *args, **opts):
580     '''check active Java source files for common stylistic errors
582     Files can be excluded from this check using the jstyle.NOT file.
583     See NOT Files in the extension documentation ('hg help cdm').
584     '''
586     filelist = buildfilelist(wslist[repo], opts.get('parent'), args)
587     exclude = not_check(repo, 'jstyle')
588     ret = 0
```



```
590     ui.write('Java style check:\n')
592     for f, e in filelist:
593         if e and e.is_removed():
594             continue
595         elif not f.endswith('.java'):
596             continue
597         elif (e or opts.get('honour_nots')) and exclude(f):
598             ui.status('Skipping %s...\n' % f)
599             continue
601         fh = open(f, 'r')
602         ret |= JStyle.jstyle(fh, output=ui, picky=True)
603         fh.close()
604     return ret


607 def cdm_permchk(ui, repo, *args, **opts):
608     '''check the permissions of each active file
610     Check that the file permissions of each added or modified file do not
611     contain the executable bit.
613     Files can be excluded from this check using the permchk.NOT file.
614     See NOT Files in the extension documentation ('hg help cdm').
615     '''
617     filelist = buildfilelist(wslist[repo], opts.get('parent'), args)
618     exclude = not_check(repo, 'permchk')
619     exeFiles = []
621     ui.write('File permission check:\n')
623     for f, e in filelist:
624         if e and e.is_removed():
625             continue
626         elif (e or opts.get('honour_nots')) and exclude(f):
627             ui.status('Skipping %s...\n' % f)
628             continue
630         mode = stat.S_IMODE(os.stat(f)[stat.ST_MODE])
631         if mode & stat.S_IEXEC:
632             exeFiles.append(f)
634     if len(exeFiles) > 0:
635         ui.write('Warning: the following active file(s) have executable mode '
636             '(+x) permission set,\nremove unless intentional:\n')
637         for fname in exeFiles:
638             ui.write("  %s\n" % fname)
640     return len(exeFiles) > 0


643 def cdm_tagchk(ui, repo, **opts):
644     '''check modification of workspace tags
646     Check for any modification of the repository's .hgtags file.
648     With the exception of the gatekeepers, nobody should introduce or
649     modify a repository's tags.
650     '''
652     active = wslist[repo].active(opts.get('parent'))
654     ui.write('Checking for new tags:\n')
```

```
656     if ".hgtags" in active:
657         tfile = wslist[repo].filepath('.hgtags')
658         ptip = active.parenttip.rev()
659
660         ui.write('Warning: Workspace contains new non-local tags.\n'
661                  'Only gatekeepers should add or modify such tags.\n'
662                  'Use the following commands to revert these changes:\n'
663                  '  hg revert -r%d %s\n'
664                  '  hg commit %s\n'
665                  'You should also recommit before integration\n' %
666                  (ptip, tfile, tfile))
667
668         return 1
669
670     return 0
671
672
673 def cdm_branchchk(ui, repo, **opts):
674     '''check for changes in number or name of branches
675
676     Check that the workspace contains only a single head, that it is
677     on the branch 'default', and that no new branches have been
678     introduced.
679     '''
680
681     ui.write('Checking for multiple heads (or branches):\n')
682
683     heads = set(repo.heads())
684     parents = set([x.node() for x in wslist[repo].workingctx().parents()])
685
686     #
687     # We care if there's more than one head, and those heads aren't
688     # identical to the dirstate parents (if they are identical, it's
689     # an uncommitted merge which mergechk will catch, no need to
690     # complain twice).
691     #
692     if len(heads) > 1 and heads != parents:
693         ui.write('Workspace has multiple heads (or branches):\n')
694         for head in [repo.changectx(head) for head in heads]:
695             ui.write("  %d:%s\t%s\n" %
696                 (head.rev(), str(head), head.description().splitlines()[0]))
697         ui.write('You must merge and recommit.\n')
698         return 1
699
700     ui.write('\nChecking for branch changes:\n')
701
702     if repo.dirstate.branch() != 'default':
703         ui.write("Warning: Workspace tip has named branch: '%s'\n"
704                  "Only gatekeepers should push new branches.\n"
705                  "Use the following commands to restore the branch name:\n"
706                  "  hg branch [-f] default\n"
707                  "  hg commit\n"
708                  "You should also recommit before integration\n" %
709                  (repo.dirstate.branch()))
710         return 1
711
712     branches = repo.branchtags().keys()
713     if len(branches) > 1:
714         ui.write('Warning: Workspace has named branches:\n')
715         for t in branches:
716             if t == 'default':
717                 continue
718             ui.write("\t%s\n" % t)
719
720         ui.write("Only gatekeepers should push new branches.\n"
```



```
721                  "Use the following commands to remove extraneous branches.\n"
722                  "  hg branch [-f] default\n"
723                  "  hg commit"
724                  "You should also recommit before integration\n")
725         return 1
726
727     return 0
728
729
730 def cdm_keywords(ui, repo, *args, **opts):
731     '''check active files for SCCS keywords
732
733     Check that any added or modified files do not contain SCCS keywords
734     (#ident lines, etc.).
735
736     Files can be excluded from this check using the keywords.NOT file.
737     See NOT Files in the extension documentation ('hg help cdm').
738     '''
739
740     filelist = buildfilelist(wslist[repo], opts.get('parent'), args)
741     exclude = not_check(repo, 'keywords')
742     ret = 0
743
744     ui.write('Keywords check:\n')
745
746     for f, e in filelist:
747         if e and e.is_removed():
748             continue
749         elif (e or opts.get('honour_nots')) and exclude(f):
750             ui.status('Skipping %s...\n' % f)
751             continue
752
753         fh = open(f, 'r')
754         ret |= Keywords.keywords(fh, output=ui)
755         fh.close()
756     return ret
757
758
759 #
760 # NB:
761 #    There's no reason to hook this up as an invokable command, since
762 #    we have 'hg status', but it must accept the same arguments.
763 #
764 def cdm_outchk(ui, repo, **opts):
765     '''Warn the user if they have uncommitted changes'''
766
767     ui.write('Checking for uncommitted changes:\n')
768
769     st = wslist[repo].modified()
770     if st:
771         ui.write('Warning: the following files have uncommitted changes:\n')
772         for elt in st:
773             ui.write('   %s\n' % elt)
774         return 1
775     return 0
776
777
778 def cdm_mergechk(ui, repo, **opts):
779     '''Warn the user if their workspace contains merges'''
780
781     active = wslist[repo].active(opts.get('parent'))
782
783     ui.write('Checking for merges:\n')
784
785     merges = filter(lambda x: len(x.parents()) == 2 and x.parents()[1],
786                     active.revs)
```

```
788     if merges:
789         ui.write('Workspace contains the following merges:\n')
790         for rev in merges:
791             desc = rev.description().splitlines()
792             ui.write('  %s:%s\t%s\n' %
793                      (rev.rev() or "working", str(rev),
794                       desc and desc[0] or "*** uncommitted change ***"))
795         return 1
796     return 0


799 def run_checks(ws, cmds, *args, **opts):
800     '''Run CMDS (with OPTS) over active files in WS'''

802     ret = 0

804     for cmd in cmds:
805         name = cmd.func_name.split('_')[1]
806         if not ws.ui.configbool('cdm', name, True):
807             ws.ui.status('Skipping %s check...\n' % name)
808         else:
809             ws.ui.pushbuffer()
810             result = cmd(ws.ui, ws.repo, honour_nots=True, *args, **opts)
811             output = ws.ui.popbuffer()

813             ret |= result

815             if not ws.ui.quiet or result != 0:
816                 ws.ui.write(output, '\n')
817     return ret


820 def cdm_nits(ui, repo, *args, **opts):
821     '''check for stylistic nits in active files

823     Check each active file for basic stylistic errors.

825     The following checks are run over each active file (see 'hg help
826     <check>' for more information about each):

828       - copyright  (copyright statements)
829       - cstyle     (C source style)
830       - hdrchk     (C header style)
831       - jstyle     (java source style)
832       - manlint    (man page formatting)
833       - mapfilechk (link-editor mapfiles)
834       - permchk    (file permissions)
835       - keywords   (SCCS keywords)

837     With the global -q/--quiet option, only provide output for those
838     checks which fail.
839     '''

841     cmds = [cdm_copyright,
842         cdm_cstyle,
843         cdm_hdrchk,
844         cdm_jstyle,
845         cmd_manlintchk,
846         cdm_mapfilechk,
847         cdm_permchk,
848         cdm_keywords]

850     return run_checks(wslist[repo], cmds, *args, **opts)
```



```
853 def cdm_pbchk(ui, repo, **opts):
854     '''run pre-integration checks on this workspace

856     Check this workspace for common errors prior to integration.

858     The following checks are run over the active list (see 'hg help
859     <check>' for more information about each):

861       - branchchk  (addition/modification of branches)
862       - comchk     (changeset descriptions)
863       - copyright  (copyright statements)
864       - cstyle     (C source style)
865       - hdrchk     (C header style)
866       - jstyle     (java source style)
867       - keywords   (SCCS keywords)
868       - manlint    (man page formatting)
869       - mapfilechk (link-editor mapfiles)
870       - permchk    (file permissions)
871       - tagchk     (addition/modification of tags)

873     Additionally, the workspace is checked for outgoing merges (which
874     should be removed with 'hg recommit'), and uncommitted changes.

876     With the global -q/--quiet option, only provide output for those
877     checks which fail.
878     '''

880     #
881     # The current ordering of these is that the commands from cdm_nits
882     # run first in the same order as they would in cdm_nits, then the
883     # pbchk specifics are run.
884     #
885     cmds = [cdm_copyright,
886         cdm_cstyle,
887         cdm_hdrchk,
888         cdm_jstyle,
889         cdm_manlintchk,
890         cdm_mapfilechk,
891         cdm_permchk,
892         cdm_keywords,
893         cdm_comchk,
894         cdm_tagchk,
895         cdm_branchchk,
896         cdm_outchk,
897         cdm_mergechk]

899     return run_checks(wslist[repo], cmds, **opts)


902 def cdm_recommit(ui, repo, **opts):
903     '''replace outgoing changesets with a single equivalent changeset

905     Replace all outgoing changesets with a single changeset containing
906     equivalent changes.  This removes uninteresting changesets created
907     during development that would only serve as noise in the gate.

909     Any changed file that is now identical in content to that in the
910     parent workspace (whether identical in history or otherwise) will
911     not be included in the new changeset.  Any merges information will
912     also be removed.

914     If no files are changed in comparison to the parent workspace, the
915     outgoing changesets will be removed, but no new changeset created.

917     recommit will refuse to run if the workspace contains more than
918     one outgoing head, even if those heads are on the same branch.  To
```

```
 919     recommit with only one branch containing outgoing changesets, your
 920     workspace must be on that branch and at that branch head.

 922     recommit will prompt you to take a backup if your workspace has
 923     been changed since the last backup was taken.  In almost all
 924     cases, you should allow it to take one (the default).

 926     recommit cannot be run if the workspace contains any uncommitted
 927     changes, applied Mq patches, or has multiple outgoing heads (or
 928     branches).
 929     '''

 931     ws = wslist[repo]

 933     if not os.getcwd().startswith(repo.root):
 934         raise util.Abort('recommit is not safe to run with -R')

 936     abort_if_dirty(ws)

 938     wlock = repo.wlock()
 939     lock = repo.lock()

 941     try:
 942         parent = ws.parent(opts['parent'])
 943         between = repo.changelog.nodesbetween(ws.findoutgoing(parent))[2]
 944         heads = set(between) & set(repo.heads())

 946         if len(heads) > 1:
 947             ui.warn('Workspace has multiple outgoing heads (or branches):\n')
 948             for head in sorted(map(repo.changelog.rev, heads), reverse=True):
 949                 ui.warn('\t%d\n' % head)
 950             raise util.Abort('you must merge before recommitting')

 952         #
 953         # We can safely use the worklist here, as we know (from the
 954         # abort_if_dirty() check above) that the working copy has not been
 955         # modified.
 956         #
 957         active = ws.active(parent)

 959         if filter(lambda b: len(b.parents()) > 1, active.bases()):
 960             raise util.Abort('Cannot recommit a merge of two non-outgoing '
 961                              'changesets')

 963         if len(active.revs) <= 0:
 964             raise util.Abort("no changes to recommit")

 966         if len(active.files()) <= 0:
 967             ui.warn("Recommitting %d active changesets, but no active files\n" %
 968                     len(active.revs))

 970         #
 971         # During the course of a recommit, any file bearing a name
 972         # matching the source name of any renamed file will be
 973         # clobbered by the operation.
 974         #
 975         # As such, we ask the user before proceeding.
 976         #
 977         bogosity = [f.parentname for f in active if f.is_renamed() and
 978                     os.path.exists(repo.wjoin(f.parentname))]
 979         if bogosity:
 980             ui.warn("The following file names are the original name of a "
 981                     "rename and also present\n"
 982                     "in the working directory:\n")

 984             for fname in bogosity:
```



```
 985                 ui.warn("  %s\n" % fname)

 987             if not yes_no(ui, "These files will be removed by recommit."
 988                           "  Continue?",
 989                           False):
 990                 raise util.Abort("recommit would clobber files")

 992         user = opts['user'] or ui.username()
 993         comments = '\n'.join(active.comments())

 995         message = cmdutil.logmessage(opts) or ui.edit(comments, user)
 996         if not message:
 997             raise util.Abort('empty commit message')

 999         bk = CdmBackup(ui, ws, backup_name(repo.root))
1000         if bk.need_backup():
1001             if yes_no(ui, 'Do you want to backup files first?', True):
1002                 bk.backup()

1004         oldtags = repo.tags()
1005         clearedtags = [(name, nd, repo.changelog.rev(nd), local)
1006                 for name, nd, local in active.tags()]

1008         ws.squishdeltas(active, message, user=user)
1009     finally:
1010         lock.release()
1011         wlock.release()

1013     if clearedtags:
1014         ui.write("Removed tags:\n")
1015         for name, nd, rev, local in sorted(clearedtags,
1016                                            key=lambda x: x[0].lower()):
1017             ui.write("  %5s:%s:\t%s%s\n" % (rev, node.short(nd),
1018                                             name, (local and ' (local)' or '')))

1020         for ntag, nnode in sorted(repo.tags().items(),
1021                                   key=lambda x: x[0].lower()):
1022             if ntag in oldtags and ntag != "tip":
1023                 if oldtags[ntag] != nnode:
1024                     ui.write("tag '%s' now refers to revision %d:%s\n" %
1025                              (ntag, repo.changelog.rev(nnode),
1026                               node.short(nnode)))


1029 def do_eval(cmd, files, root, changedir=True):
1030     if not changedir:
1031         os.chdir(root)

1033     for path in sorted(files):
1034         dirn, base = os.path.split(path)

1036         if changedir:
1037             os.chdir(os.path.join(root, dirn))

1039         os.putenv('workspace', root)
1040         os.putenv('filepath', path)
1041         os.putenv('dir', dirn)
1042         os.putenv('file', base)
1043         os.system(cmd)


1046 def cdm_eval(ui, repo, *command, **opts):
1047     '''run specified command for each active file

1049     Run the command specified on the command line for each active
1050     file, with the following variables present in the environment:
```

```
1052       :$file:      -  active file basename.
1053       :$dir:       -  active file dirname.
1054       :$filepath:  -  path from workspace root to active file.
1055       :$workspace: -  full path to workspace root.

1057     For example:

1059       hg eval 'echo $dir; hg log -l3 $file'

1061     will show the last the 3 log entries for each active file,
1062     preceded by its directory.
1063     '''

1065     act = wslist[repo].active(opts['parent'])
1066     cmd = ' '.join(command)
1067     files = [x.name for x in act if not x.is_removed()]

1069     do_eval(cmd, files, repo.root, not opts['remain'])


1072 def cdm_apply(ui, repo, *command, **opts):
1073     '''apply specified command to all active files

1075     Run the command specified on the command line over each active
1076     file.

1078     For example 'hg apply "wc -l"' will output a count of the lines in
1079     each active file.
1080     '''

1082     act = wslist[repo].active(opts['parent'])

1084     if opts['remain']:
1085         appnd = ' $filepath'
1086     else:
1087         appnd = ' $file'

1089     cmd = ' '.join(command) + appnd
1090     files = [x.name for x in act if not x.is_removed()]

1092     do_eval(cmd, files, repo.root, not opts['remain'])


1095 def cdm_reparent(ui, repo, parent):
1096     '''reparent your workspace

1098     Update the 'default' path alias that is used as the default source
1099     for 'hg pull' and the default destination for 'hg push' (unless
1100     there is a 'default-push' alias).  This is also the path all
1101     Cadmium commands treat as your parent workspace.
1102     '''

1104     def append_new_parent(parent):
1105         fp = None
1106         try:
1107             fp = repo.opener('hgrc', 'a', atomictemp=True)
1108             if fp.tell() != 0:
1109                 fp.write('\n')
1110             fp.write('[paths]\n'
1111                      'default = %s\n\n' % parent)
1112             fp.rename()
1113         finally:
1114             if fp and not fp.closed:
1115                 fp.close()
```



```
1117     def update_parent(path, line, parent):
1118         line = line - 1 # The line number we're passed will be 1-based
1119         fp = None

1121         try:
1122             fp = open(path)
1123             data = fp.readlines()
1124         finally:
1125             if fp and not fp.closed:
1126                 fp.close()

1128         #
1129         # line will be the last line of any continued block, go back
1130         # to the first removing the continuation as we go.
1131         #
1132         while data[line][0].isspace():
1133             data.pop(line)
1134             line -= 1

1136         assert data[line].startswith('default')

1138         data[line] = "default = %s\n" % parent
1139         if data[-1] != '\n':
1140             data.append('\n')

1142         try:
1143             fp = util.atomictempfile(path, 'w', 0644)
1144             fp.writelines(data)
1145             fp.rename()
1146         finally:
1147             if fp and not fp.closed:
1148                 fp.close()

1150     from mercurial import config
1151     parent = ui.expandpath(parent)

1153     if not os.path.exists(repo.join('hgrc')):
1154         append_new_parent(parent)
1155         return

1157     cfg = config.config()
1158     cfg.read(repo.join('hgrc'))
1159     source = cfg.source('paths', 'default')

1161     if not source:
1162         append_new_parent(parent)
1163         return
1164     else:
1165         path, target = source.rsplit(':', 1)

1167         if path != repo.join('hgrc'):
1168             raise util.Abort("Cannot edit path specification not in repo hgrc\n"
1169                              "default path is from: %s" % source)

1171         update_parent(path, int(target), parent)


1174 def backup_name(fullpath):
1175     '''Create a backup directory name based on the specified path.

1177     In most cases this is the basename of the path specified, but
1178     certain cases are handled specially to create meaningful names'''

1180     special = ['usr/closed']

1182     fullpath = fullpath.rstrip(os.path.sep).split(os.path.sep)
```

```
1184     #
1185     # If a path is 'special', we append the basename of the path to
1186     # the path element preceding the constant, special, part.
1187     #
1188     # Such that for instance:
1189     #     /foo/bar/onnv-fixes/usr/closed
1190     #  has a backup name of:
1191     #     onnv-fixes-closed
1192     #
1193     for elt in special:
1194         elt = elt.split(os.path.sep)
1195         pathpos = len(elt)

1197         if fullpath[-pathpos:] == elt:
1198             return "%s-%s" % (fullpath[-pathpos - 1], elt[-1])
1199     else:
1200         return fullpath[-1]


1203 def cdm_backup(ui, repo, if_newer=False):
1204     '''backup workspace changes and metadata

1206     Create a backup copy of changes made in this workspace as compared
1207     to its parent workspace, as well as important metadata of this
1208     workspace.

1210     NOTE: Only changes as compared to the parent workspace are backed
1211     up.  If you lose this workspace and its parent, you will not be
1212     able to restore a backup into a clone of the grandparent
1213     workspace.

1215     By default, backups are stored in the cdm.backup/ directory in
1216     your home directory.  This is configurable using the cdm.backupdir
1217     configuration variable, for example:

1219       hg backup --config cdm.backupdir=/net/foo/backups

1221     or place the following in an appropriate hgrc file::

1223       [cdm]
1224       backupdir = /net/foo/backups

1226     Backups have the same name as the workspace in which they were
1227     taken, with '-closed' appended in the case of O/N's usr/closed.
1228     '''

1230     name = backup_name(repo.root)
1231     bk = CdmBackup(ui, wslist[repo], name)

1233     wlock = repo.wlock()
1234     lock = repo.lock()

1236     try:
1237         if if_newer and not bk.need_backup():
1238             ui.status('backup is up-to-date\n')
1239         else:
1240             bk.backup()
1241     finally:
1242         lock.release()
1243         wlock.release()


1246 def cdm_restore(ui, repo, backup, **opts):
1247     '''restore workspace from backup
```



```
1249     Restore this workspace from a backup (taken by 'hg backup').

1251     If the specified backup directory does not exist, it is assumed to
1252     be relative to the cadmium backup directory (~/cdm.backup/ by
1253     default).

1255     For example::

1257       % hg restore on-rfe - Restore the latest backup of ~/cdm.backup/on-rfe
1258       % hg restore -g3 on-rfe - Restore the 3rd backup of ~/cdm.backup/on-rfe
1259       % hg restore /net/foo/backup/on-rfe - Restore from an explicit path
1260     '''

1262     if not os.getcwd().startswith(repo.root):
1263         raise util.Abort('restore is not safe to run with -R')

1265     abort_if_dirty(wslist[repo])

1267     if opts['generation']:
1268         gen = int(opts['generation'])
1269     else:
1270         gen = None

1272     if os.path.exists(backup):
1273         backup = os.path.abspath(backup)

1275     wlock = repo.wlock()
1276     lock = repo.lock()

1278     try:
1279         bk = CdmBackup(ui, wslist[repo], backup)
1280         bk.restore(gen)
1281     finally:
1282         lock.release()
1283         wlock.release()


1286 def cdm_webrev(ui, repo, **opts):
1287     '''generate web-based code review and optionally upload it

1289     Generate a web-based code review using webrev(1) and optionally
1290     upload it.  All known arguments are passed through to webrev(1).
1291     '''

1293     webrev_args = ""
1294     for key in opts.keys():
1295         if opts[key]:
1296             if type(opts[key]) == type(True):
1297                 webrev_args += '-' + key + ' '
1298             else:
1299                 webrev_args += '-' + key + ' ' + opts[key] + ' '

1301     retval = os.system('webrev ' + webrev_args)
1302     if retval != 0:
1303         return retval - 255

1305     return 0


1308 def cdm_debugcdmal(ui, repo, *pats, **opts):
1309     '''dump the active list for the sake of debugging/testing'''

1311     ui.write(wslist[repo].active(opts['parent']).as_text(pats))


1314 def cdm_changed(ui, repo, *pats, **opts):
```

```
1315     '''mark a file as changed in the working copy

1317     Maintain a list of files checked for modification in the working
1318     copy.  If the list exists, most cadmium commands will only check
1319     the working copy for changes to those files, rather than checking
1320     the whole workspace (this does not apply to committed changes,
1321     which are always seen).

1323     Since this list functions only as a hint as to where in the
1324     working copy to look for changes, entries that have not actually
1325     been modified (in the working copy, or in general) are not
1326     problematic.


1329     Note: If such a list exists, it must be kept up-to-date.


1332     Renamed files can be added with reference only to their new name:
1333       $ hg mv foo bar
1334       $ hg changed bar

1336     Without arguments, 'hg changed' will list all files recorded as
1337     altered, such that, for instance:
1338       $ hg status $(hg changed)
1339       $ hg diff $(hg changed)
1340     Become useful (generally faster than their unadorned counterparts)

1342     To create an initially empty list:
1343       $ hg changed -i
1344     Until files are added to the list it is equivalent to saying
1345     "Nothing has been changed"

1347     Update the list based on the current active list:
1348       $ hg changed -u
1349     The old list is emptied, and replaced with paths from the
1350     current active list.

1352     Remove the list entirely:
1353       $ hg changed -d
1354     '''

1356     def modded_files(repo, parent):
1357         out = wslist[repo].findoutgoing(wslist[repo].parent(parent))
1358         outnodes = repo.changelog.nodesbetween(out)[0]

1360         files = set()
1361         for n in outnodes:
1362             files.update(repo.changectx(n).files())

1364         files.update(wslist[repo].status().keys())
1365         return files

1367     #
1368     # specced_pats is convenient to treat as a boolean indicating
1369     # whether any file patterns or paths were specified.
1370     #
1371     specced_pats = pats or opts['include'] or opts['exclude']
1372     if len(filter(None, [opts['delete'], opts['update'], opts['init'],
1373                          specced_pats])) > 1:
1374         raise util.Abort("-d, -u, -i and patterns are mutually exclusive")

1376     wl = WorkList(wslist[repo])

1378     if (not wl and specced_pats) or opts['init']:
1379         wl.delete()
1380         if yes_no(ui, "Create a list based on your changes thus far?", True):
```



```
1381             map(wl.add, modded_files(repo, opts.get('parent')))

1383     if opts['delete']:
1384         wl.delete()
1385     elif opts['update']:
1386         wl.delete()
1387         map(wl.add, modded_files(repo, opts.get('parent')))
1388         wl.write()
1389     elif opts['init']:       # Any possible old list was deleted above
1390         wl.write()
1391     elif specced_pats:
1392         sources = []

1394         match = wslist[repo].matcher(pats=pats, opts=opts)
1395         for abso in repo.walk(match):
1396             if abso in repo.dirstate:
1397                 wl.add(abso)
1398                 #
1399                 # Store the source name of any copy.  We use this so
1400                 # both the add and delete of a rename can be entered
1401                 # into the WorkList with only the destination name
1402                 # explicitly being mentioned.
1403                 #
1404                 fctx = wslist[repo].workingctx().filectx(abso)
1405                 rn = fctx.renamed()
1406                 if rn:
1407                     sources.append(rn[0])
1408             else:
1409                 ui.warn("%s is not version controlled -- skipping\n" %
1410                         match.rel(abso))

1412         if sources:
1413             for fname, chng in wslist[repo].status(files=sources).iteritems():
1414                 if chng == 'removed':
1415                     wl.add(fname)
1416         wl.write()
1417     else:
1418         for elt in sorted(wl.list()):
1419             ui.write("%s\n" % wslist[repo].filepath(elt))


1422 cmdtable = {
1423     'apply': (cdm_apply, [('p', 'parent', '', 'parent workspace'),
1424                           ('r', 'remain', None, 'do not change directory')],
1425               'hg apply [-p PARENT] [-r] command...'),
1426     '^backup|bu': (cdm_backup, [('t', 'if-newer', None,
1427                                  'only backup if workspace files are newer')],
1428                    'hg backup [-t]'),
1429     'branchchk': (cdm_branchchk, [('p', 'parent', '', 'parent workspace')],
1430                   'hg branchchk [-p PARENT]'),
1431     'bugs': (cdm_bugs, [('p', 'parent', '', 'parent workspace')],
1432              'hg bugs [-p PARENT]'),
1433     'cddlchk': (cdm_cddlchk, [('p', 'parent', '', 'parent workspace')],
1434                 'hg cddlchk [-p PARENT]'),
1435     'changed': (cdm_changed, [('d', 'delete', None, 'delete the file list'),
1436                               ('u', 'update', None, 'mark all changed files'),
1437                               ('i', 'init', None, 'create an empty file list'),
1438                               ('p', 'parent', '', 'parent workspace'),
1439                               ('I', 'include', [],
1440                                'include names matching the given patterns'),
1441                               ('X', 'exclude', [],
1442                                'exclude names matching the given patterns')],
1443                 'hg changed -d\n'
1444                 'hg changed -u\n'
1445                 'hg changed -i\n'
1446                 'hg changed [-I PATTERN...] [-X PATTERN...] [FILE...]'),
```

```
1447     'comchk': (cdm_comchk, [('p', 'parent', '', 'parent workspace'),
1448                             ('N', 'nocheck', None,
1449                              'do not compare comments with databases')],
1450                'hg comchk [-p PARENT]'),
1451     'comments': (cdm_comments, [('p', 'parent', '', 'parent workspace')],
1452                  'hg comments [-p PARENT]'),
1453     'copyright': (cdm_copyright, [('p', 'parent', '', 'parent workspace')],
1454                   'hg copyright [-p PARENT]'),
1455     'cstyle': (cdm_cstyle, [('p', 'parent', '', 'parent workspace')],
1456                'hg cstyle [-p PARENT]'),
1457     'debugcdmal': (cdm_debugcdmal, [('p', 'parent', '', 'parent workspace')],
1458                    'hg debugcdmal [-p PARENT] [FILE...]'),
1459     'eval': (cdm_eval, [('p', 'parent', '', 'parent workspace'),
1460                         ('r', 'remain', None, 'do not change directory')],
1461              'hg eval [-p PARENT] [-r] command...'),
1462     'hdrchk': (cdm_hdrchk, [('p', 'parent', '', 'parent workspace')],
1463                'hg hdrchk [-p PARENT]'),
1464     'jstyle': (cdm_jstyle, [('p', 'parent', '', 'parent workspace')],
1465                'hg jstyle [-p PARENT]'),
1466     'keywords': (cdm_keywords, [('p', 'parent', '', 'parent workspace')],
1467                  'hg keywords [-p PARENT]'),
1468     '^list|active': (cdm_list, [('p', 'parent', '', 'parent workspace'),
1469                                 ('a', 'added', None, 'show added files'),
1470                                 ('m', 'modified', None, 'show modified files'),
1471                                 ('r', 'removed', None, 'show removed files')],
1472                      'hg list [-amrRu] [-p PARENT]'),
1473     'manlint': (cdm_manlintchk, [('p', 'parent', '', 'parent workspace')],
1474                 'hg manlint [-p PARENT]'),
1475     'mapfilechk': (cdm_mapfilechk, [('p', 'parent', '', 'parent workspace')],
1476                    'hg mapfilechk [-p PARENT]'),
1477     '^nits': (cdm_nits, [('p', 'parent', '', 'parent workspace')],
1478               'hg nits [-p PARENT]'),
1479     '^pbchk': (cdm_pbchk, [('p', 'parent', '', 'parent workspace'),
1480                            ('N', 'nocheck', None, 'skip database checks')],
1481                'hg pbchk [-N] [-p PARENT]'),
1482     'permchk': (cdm_permchk, [('p', 'parent', '', 'parent workspace')],
1483                 'hg permchk [-p PARENT]'),
1484     '^pdiffs': (cdm_pdiffs, [('p', 'parent', '', 'parent workspace'),
1485                              ('a', 'text', None, 'treat all files as text'),
1486                              ('g', 'git', None, 'use extended git diff format'),
1487                              ('w', 'ignore-all-space', None,
1488                               'ignore white space when comparing lines'),
1489                              ('b', 'ignore-space-change', None,
1490                               'ignore changes in the amount of white space'),
1491                              ('B', 'ignore-blank-lines', None,
1492                               'ignore changes whose lines are all blank'),
1493                              ('U', 'unified', 3,
1494                               'number of lines of context to show'),
1495                              ('I', 'include', [],
1496                               'include names matching the given patterns'),
1497                              ('X', 'exclude', [],
1498                               'exclude names matching the given patterns')],
1499                'hg pdiffs [OPTION...] [-p PARENT] [FILE...]'),
1500     '^recommit|reci': (cdm_recommit, [('p', 'parent', '', 'parent workspace'),
1501                                       ('m', 'message', '',
1502                                        'use <text> as commit message'),
1503                                       ('l', 'logfile', '',
1504                                        'read commit message from file'),
1505                                       ('u', 'user', '',
1506                                        'record user as committer')],
1507                        'hg recommit [-m TEXT] [-l FILE] [-u USER] [-p PARENT]'),
1508     'renamed': (cdm_renamed, [('p', 'parent', '', 'parent workspace')],
1509                 'hg renamed [-p PARENT]'),
1510     'reparent': (cdm_reparent, [], 'hg reparent PARENT'),
1511     '^restore': (cdm_restore, [('g', 'generation', '', 'generation number')],
1512                  'hg restore [-g GENERATION] BACKUP'),
```



```
1513     'tagchk': (cdm_tagchk, [('p', 'parent', '', 'parent workspace')],
1514                'hg tagchk [-p PARENT]'),
1515     'webrev': (cdm_webrev, [('C', 'C', '', 'ITS priority file'),
1516                             ('D', 'D', '', 'delete remote webrev'),
1517                             ('I', 'I', '', 'ITS configuration file'),
1518                             ('i', 'i', '', 'include file'),
1519                             ('N', 'N', None, 'suppress comments'),
1520                             ('n', 'n', None, 'do not generate webrev'),
1521                             ('O', 'O', None, 'OpenSolaris mode'),
1522                             ('o', 'o', '', 'output directory'),
1523                             ('p', 'p', '', 'use specified parent'),
1524                             ('t', 't', '', 'upload target'),
1525                             ('U', 'U', None, 'upload the webrev'),
1526                             ('w', 'w', '', 'use wx active file')],
1527                'hg webrev [WEBREV_OPTIONS]'),
1528 }
______unchanged_portion_omitted_
```

```
**********
   10980 Thu Jul 17 00:50:43 2014
new/usr/src/tools/scripts/git-pbchk.py
manpage lint.
**********
```

```
#!/usr/bin/python2.6
#
#  This program is free software; you can redistribute it and/or modify
#  it under the terms of the GNU General Public License version 2
#  as published by the Free Software Foundation.
#
#  This program is distributed in the hope that it will be useful,
#  but WITHOUT ANY WARRANTY; without even the implied warranty of
#  MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#  GNU General Public License for more details.
#
#  You should have received a copy of the GNU General Public License
#  along with this program; if not, write to the Free Software
#  Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.
#

#
# Copyright (c) 2008, 2010, Oracle and/or its affiliates. All rights reserved.
# Copyright 2008, 2012 Richard Lowe
# Copyright 2014 Garrett D'Amore <garrett@damore.org>
#

import getopt
import os
import re
import subprocess
import sys
import tempfile

from cStringIO import StringIO

# This is necessary because, in a fit of pique, we used hg-format ignore lists
# for NOT files.
from mercurial import ignore

#
# Adjust the load path based on our location and the version of python into
# which it is being loaded.  This assumes the normal onbld directory
# structure, where we are in bin/ and the modules are in
# lib/python(version)?/onbld/Scm/.  If that changes so too must this.
#
sys.path.insert(1, os.path.join(os.path.dirname(__file__), "..", "lib",
                                "python%d.%d" % sys.version_info[:2]))

#
# Add the relative path to usr/src/tools to the load path, such that when run
# from the source tree we use the modules also within the source tree.
#
sys.path.insert(2, os.path.join(os.path.dirname(__file__), ".."))

from onbld.Checks import Comments, Copyright, CStyle, HdrChk
from onbld.Checks import JStyle, Keywords, ManLint, Mapfile
from onbld.Checks import JStyle, Keywords, Mapfile


class GitError(Exception):
    pass

def git(command):
    """Run a command and return a stream containing its stdout (and write its
    stderr to its stdout)"""
```



```
    if type(command) != list:
        command = command.split()

    command = ["git"] + command

    try:
        tmpfile = tempfile.TemporaryFile(prefix="git-nits")
    except EnvironmentError, e:
        raise GitError("Could not create temporary file: %s\n" % e)

    try:
        p = subprocess.Popen(command,
                             stdout=tmpfile,
                             stderr=subprocess.STDOUT)
    except OSError, e:
        raise GitError("could not execute %s: %s\n" (command, e))

    err = p.wait()
    if err != 0:
        raise GitError(p.stdout.read())

    tmpfile.seek(0)
    return tmpfile


def git_root():
    """Return the root of the current git workspace"""

    p = git('rev-parse --git-dir')

    if not p:
        sys.stderr.write("Failed finding git workspace\n")
        sys.exit(err)

    return os.path.abspath(os.path.join(p.readlines()[0],
                                        os.path.pardir))


def git_branch():
    """Return the current git branch"""

    p = git('branch')

    if not p:
        sys.stderr.write("Failed finding git branch\n")
        sys.exit(err)

    for elt in p:
        if elt[0] == '*':
            if elt.endswith('(no branch)'):
                return None
            return elt.split()[1]


def git_parent_branch(branch):
    """Return the parent of the current git branch.

    If this branch tracks a remote branch, return the remote branch which is
    tracked.  If not, default to origin/master."""

    if not branch:
        return None

    p = git("for-each-ref --format=%(refname:short) %(upstream:short) " +
            "refs/heads/")
```

```
128     if not p:
129         sys.stderr.write("Failed finding git parent branch\n")
130         sys.exit(err)
131
132     for line in p:
133         # Git 1.7 will leave a ' ' trailing any non-tracking branch
134         if ' ' in line and not line.endswith(' \n'):
135             local, remote = line.split()
136             if local == branch:
137                 return remote
138     return 'origin/master'
139
140
141 def git_comments(parent):
142     """Return a list of any checkin comments on this git branch"""
143
144     p = git('log --pretty=tformat:%%B:SEP: %s..' % parent)
145
146     if not p:
147         sys.stderr.write("Failed getting git comments\n")
148         sys.exit(err)
149
150     return [x.strip() for x in p.readlines() if x != ':SEP:\n']
151
152
153 def git_file_list(parent, paths=None):
154     """Return the set of files which have ever changed on this branch.
155
156     NB: This includes files which no longer exist, or no longer actually
157     differ."""
158
159     p = git("log --name-only --pretty=format: %s.. %s" %
160             (parent, ' '.join(paths)))
161
162     if not p:
163         sys.stderr.write("Failed building file-list from git\n")
164         sys.exit(err)
165
166     ret = set()
167     for fname in p:
168         if fname and not fname.isspace() and fname not in ret:
169             ret.add(fname.strip())
170
171     return ret
172
173
174 def not_check(root, cmd):
175     """Return a function which returns True if a file given as an argument
176     should be excluded from the check named by 'cmd'"""
177
178     ignorefiles = filter(os.path.exists,
179                          [os.path.join(root, ".git", "%s.NOT" % cmd),
180                           os.path.join(root, "exception_lists", cmd)])
181     if len(ignorefiles) > 0:
182         return ignore.ignore(root, ignorefiles, sys.stderr.write)
183     else:
184         return lambda x: False
185
186
187 def gen_files(root, parent, paths, exclude):
188     """Return a function producing file names, relative to the current
189     directory, of any file changed on this branch (limited to 'paths' if
190     requested), and excluding files for which exclude returns a true value """
191
192     # Taken entirely from Python 2.6's os.path.relpath which we would use if we
```



```
193     # could.
194     def relpath(path, here):
195         c = os.path.abspath(os.path.join(root, path)).split(os.path.sep)
196         s = os.path.abspath(here).split(os.path.sep)
197         l = len(os.path.commonprefix((s, c)))
198         return os.path.join(*[os.path.pardir] * (len(s)-l) + c[l:])
199
200     def ret(select=None):
201         if not select:
202             select = lambda x: True
203
204         for f in git_file_list(parent, paths):
205             f = relpath(f, '.')
206             if (os.path.exists(f) and select(f) and not exclude(f)):
207                 yield f
208     return ret
209
210
211 def comchk(root, parent, flist, output):
212     output.write("Comments:\n")
213
214     return Comments.comchk(git_comments(parent), check_db=True,
215                            output=output)
216
217
218 def mapfilechk(root, parent, flist, output):
219     ret = 0
220
221     # We are interested in examining any file that has the following
222     # in its final path segment:
223     #    - Contains the word 'mapfile'
224     #    - Begins with 'map.'
225     #    - Ends with '.map'
226     # We don't want to match unless these things occur in final path segment
227     # because directory names with these strings don't indicate a mapfile.
228     # We also ignore files with suffixes that tell us that the files
229     # are not mapfiles.
230     MapfileRE = re.compile(r'.*((mapfile[^/]*)|(/map\.+[^/]*)|(\.map))$',
231         re.IGNORECASE)
232     NotMapSuffixRE = re.compile(r'.*\.[ch]$', re.IGNORECASE)
233
234     output.write("Mapfile comments:\n")
235
236     for f in flist(lambda x: MapfileRE.match(x) and not
237                    NotMapSuffixRE.match(x)):
238         fh = open(f, 'r')
239         ret |= Mapfile.mapfilechk(fh, output=output)
240         fh.close()
241     return ret
242
243
244 def copyright(root, parent, flist, output):
245     ret = 0
246     output.write("Copyrights:\n")
247     for f in flist():
248         fh = open(f, 'r')
249         ret |= Copyright.copyright(fh, output=output)
250         fh.close()
251     return ret
252
253
254 def hdrchk(root, parent, flist, output):
255     ret = 0
256     output.write("Header format:\n")
257     for f in flist(lambda x: x.endswith('.h')):
258         fh = open(f, 'r')
```

```
259         ret |= HdrChk.hdrchk(fh, lenient=True, output=output)
260         fh.close()
261     return ret


264 def cstyle(root, parent, flist, output):
265     ret = 0
266     output.write("C style:\n")
267     for f in flist(lambda x: x.endswith('.c') or x.endswith('.h')):
268         fh = open(f, 'r')
269         ret |= CStyle.cstyle(fh, output=output, picky=True,
270                              check_posix_types=True,
271                              check_continuation=True)
272         fh.close()
273     return ret


276 def jstyle(root, parent, flist, output):
277     ret = 0
278     output.write("Java style:\n")
279     for f in flist(lambda x: x.endswith('.java')):
280         fh = open(f, 'r')
281         ret |= JStyle.jstyle(fh, output=output, picky=True)
282         fh.close()
283     return ret


286 def manlint(root, parent, flist, output):
287     ret = 0
288     output.write("Man page format:\n")
289     ManfileRE = re.compile(r'.*\.[0-9][a-z]*$', re.IGNORECASE)
290     for f in flist(lambda x: ManfileRE.match(x)):
291         fh = open(f, 'r')
292         ret |= ManLint.manlint(fh, output=output, picky=True)
293         fh.close()
294     return ret

296 def keywords(root, parent, flist, output):
297     ret = 0
298     output.write("SCCS Keywords:\n")
299     for f in flist():
300         fh = open(f, 'r')
301         ret |= Keywords.keywords(fh, output=output)
302         fh.close()
303     return ret


306 def run_checks(root, parent, cmds, paths='', opts={}):
307     """Run the checks given in 'cmds', expected to have well-known signatures,
308     and report results for any which fail.

310     Return failure if any of them did.

312     NB: the function name of the commands passed in is used to name the NOT
313     file which excepts files from them."""

315     ret = 0

317     for cmd in cmds:
318         s = StringIO()

320         exclude = not_check(root, cmd.func_name)
321         result = cmd(root, parent, gen_files(root, parent, paths, exclude),
322                      output=s)
323         ret |= result
```



```
325         if result != 0:
326             print s.getvalue()

328     return ret


331 def nits(root, parent, paths):
332     cmds = [copyright,
333             cstyle,
334             hdrchk,
335             jstyle,
336             keywords,
337             manlint,
338             mapfilechk]
339     run_checks(root, parent, cmds, paths)


342 def pbchk(root, parent, paths):
343     cmds = [comchk,
344             copyright,
345             cstyle,
346             hdrchk,
347             jstyle,
348             keywords,
349             manlint,
350             mapfilechk]
351     run_checks(root, parent, cmds)


354 def main(cmd, args):
355     parent_branch = None

357     try:
358         opts, args = getopt.getopt(args, 'b:')
359     except getopt.GetoptError, e:
360         sys.stderr.write(str(e) + '\n')
361         sys.stderr.write("Usage: %s [-b branch] [path...]\n" % cmd)
362         sys.exit(1)

364     for opt, arg in opts:
365         if opt == '-b':
366             parent_branch = arg

368     if not parent_branch:
369         parent_branch = git_parent_branch(git_branch())

371     func = nits
372     if cmd == 'git-pbchk':
373         func = pbchk
374         if args:
375             sys.stderr.write("only complete workspaces may be pbchk'd\n");
376             sys.exit(1)

378     func(git_root(), parent_branch, args)

380 if __name__ == '__main__':
381     try:
382         main(os.path.basename(sys.argv[0]), sys.argv[1:])
383     except GitError, e:
384         sys.stderr.write("failed to run git:\n %s\n" % str(e))
385         sys.exit(1)
```