

NAME

dd - convert and copy a file

SYNOPSIS

dd [*operands ...*]

DESCRIPTION

The **dd** utility copies the standard input to the standard output. Input data is read and written in 512-byte blocks. If input reads are short, input from multiple reads are aggregated to form the output block.

When finished, **dd** displays the number of complete and partial input and output blocks and truncated input records to the standard error output.

The following operands are available:

bs=*n* Set both input and output block size to *n* bytes, superseding the **ibs** and **obs** operands. If no conversion values other than **noerror**, **notrunc** or **sync** are specified, then each input block is copied to the output as a single block without any aggregation of short blocks.

cbs=*n* Set the conversion record size to *n* bytes. The conversion record size is required by the record oriented conversion values.

count=*n*
Copy only *n* input blocks.

files=*n*
Copy *n* input files before terminating. This operand is only applicable when the input device is a tape.

fillchar=*c*
When padding a block in conversion mode or due to use of **noerror** and **sync** modes, fill with the specified ASCII character, rather than using a space or NUL.

ibs=*n* Set the input block size to *n* bytes instead of the default 512.

if=*file* Read input from *file* instead of the standard input.

iflag=*value*[,*value ...*]
Where **value** is one of the symbols from the following list.

fullblock Reading from the input file may not obtain a full block. When a read returns short,

continue reading to fill the block. Without this flag, **count** limits the number of times read(2) is called on the input rather than the number of blocks copied in full. May not be combined with **conv=sync**.

direct Set the O_DIRECT flag on the input file to make reads bypass any local caching.

iseek=*n*

Seek on the input file *n* blocks. This is synonymous with **skip=*n***.

obs=*n* Set the output block size to *n* bytes instead of the default 512.

of=*file* Write output to *file* instead of the standard output. Any regular output file is truncated unless the **notrunc** conversion value is specified. If an initial portion of the output file is seeked past (see the **oseek** operand), the output file is truncated at that point.

oflag=*value*[,*value* ...]

Where **value** is one of the symbols from the following list.

fsync Set the O_FSYNC flag on the output file to make writes synchronous.

sync Set the O_SYNC flag on the output file to make writes synchronous. This is synonymous with the **fsync** value.

direct Set the O_DIRECT flag on the output file to make writes bypass any local caching.

oseek=*n*

Seek on the output file *n* blocks. This is synonymous with **seek=*n***.

seek=*n*

Seek *n* blocks from the beginning of the output before copying. On non-tape devices, an lseek(2) operation is used. Otherwise, existing blocks are read and the data discarded. If the user does not have read permission for the tape, it is positioned using the tape ioctl(2) function calls. If the seek operation is past the end of file, space from the current end of file to the specified offset is filled with blocks of NUL bytes.

skip=*n*

Skip *n* blocks from the beginning of the input before copying. On input which supports seeks, an lseek(2) operation is used. Otherwise, input data is read and discarded. For pipes, the correct number of bytes is read. For all other devices, the correct number of blocks is read without distinguishing between a partial or complete block being read.

speed=*n*

Limit the copying speed to *n* bytes per second.

status=*value*

Where **value** is one of the symbols from the following list.

noxfers Do not print the transfer statistics as the last line of status output.

none Do not print the status output. Error messages are shown; informational messages are not.

progress Print basic transfer statistics once per second.

conv=*value*[,*value* ...]

Where **value** is one of the symbols from the following list.

ascii, oldascii

The same as the **unblock** value except that characters are translated from EBCDIC to ASCII before the records are converted. (These values imply **unblock** if the operand **cbs** is also specified.) There are two conversion maps for ASCII. The value **ascii** specifies the recommended one which is compatible with AT&T System V UNIX. The value **oldascii** specifies the one used in historic AT&T UNIX and pre-4.3BSD-Reno systems.

block Treats the input as a sequence of newline or end-of-file terminated variable length records independent of input and output block boundaries. Any trailing newline character is discarded. Each input record is converted to a fixed length output record where the length is specified by the **cbs** operand. Input records shorter than the conversion record size are padded with spaces. Input records longer than the conversion record size are truncated. The number of truncated input records, if any, are reported to the standard error output at the completion of the copy.

ebcdic, ibm, oldebcdic, oldibm

The same as the **block** value except that characters are translated from ASCII to EBCDIC after the records are converted. (These values imply **block** if the operand **cbs** is also specified.) There are four conversion maps for EBCDIC. The value **ebcdic** specifies the recommended one which is compatible with AT&T System V UNIX. The value **ibm** is a slightly different mapping, which is compatible with the AT&T System V UNIX **ibm** value. The values **oldebcdic** and **oldibm** are maps used in historic AT&T UNIX and pre-4.3BSD-Reno systems.

fdatasync

Perform an `fdatasync(2)` on the output file before closing it.

fsync Perform an `fsync(2)` on the output file before closing it.

lcase Transform uppercase characters into lowercase characters.

pareven, parnone, parodd, parset

Output data with the specified parity. The parity bit on input is stripped unless EBCDIC to ASCII conversions is also specified.

noerror Do not stop processing on an input error. When an input error occurs, a diagnostic message followed by the current input and output block counts will be written to the standard error output in the same format as the standard completion message. If the **sync** conversion is also specified, any missing input data will be replaced with NUL bytes (or with spaces if a block oriented conversion value was specified) and processed as a normal input buffer. If the **fillchar** option is specified, the fill character provided on the command line will override the automatic selection of the fill character. If the **sync** conversion is not specified, the input block is omitted from the output. On input files which are not tapes or pipes, the file offset will be positioned past the block in which the error occurred using `lseek(2)`.

notrunc Do not truncate the output file. This will preserve any blocks in the output file not explicitly written by **dd**. The **notrunc** value is not supported for tapes.

osync Pad the final output block to the full output block size. If the input file is not a multiple of the output block size after conversion, this conversion forces the final output block to be the same size as preceding blocks for use on devices that require regularly sized blocks to be written. This option is incompatible with use of the **bs=*n*** block size specification.

sparse If one or more output blocks would consist solely of NUL bytes, try to seek the output file by the required space instead of filling them with NULs, resulting in a sparse file.

swab Swap every pair of input bytes. If an input buffer has an odd number of bytes, the last byte will be ignored during swapping.

sync Pad every input block to the input buffer size. Spaces are used for pad bytes if a block oriented conversion value is specified, otherwise NUL bytes are used.

ucase Transform lowercase characters into uppercase characters.

unblock Treats the input as a sequence of fixed length records independent of input and output block boundaries. The length of the input records is specified by the **cbs** operand. Any trailing space characters are discarded and a newline character is appended.

Where sizes or speed are specified, a decimal, octal, or hexadecimal number of bytes is expected. If the number ends with a "b", "k", "m", "g", "t", "p", or "w", the number is multiplied by 512, 1024 (1K), 1048576 (1M), 1073741824 (1G), 1099511627776 (1T), 1125899906842624 (1P) or the number of bytes in an integer, respectively. Two or more numbers may be separated by an "x" to indicate a product.

When finished, **dd** displays the number of complete and partial input and output blocks, truncated input records and odd-length byte-swapping blocks to the standard error output. A partial input block is one where less than the input block size was read. A partial output block is one where less than the output block size was written. Partial output blocks to tape devices are considered fatal errors. Otherwise, the rest of the block will be written. Partial output blocks to character devices will produce a warning message. A truncated input block is one where a variable length record oriented conversion value was specified and the input line was too long to fit in the conversion record or was not newline terminated.

Normally, data resulting from input or conversion or both are aggregated into output blocks of the specified size. After the end of input is reached, any remaining output is written as a block. This means that the final output block may be shorter than the output block size.

If **dd** receives a SIGINFO (see the **status** argument for stty(1)) signal, the current input and output block counts will be written to the standard error output in the same format as the standard completion message. If **dd** receives a SIGINT signal, the current input and output block counts will be written to the standard error output in the same format as the standard completion message and **dd** will exit.

EXIT STATUS

The **dd** utility exits 0 on success, and >0 if an error occurs.

EXAMPLES

Check that a disk drive contains no bad blocks:

```
dd if=/dev/ada0 of=/dev/null bs=1m
```

Do a refresh of a disk drive, in order to prevent presently recoverable read errors from progressing into unrecoverable read errors:

```
dd if=/dev/ada0 of=/dev/ada0 bs=1m
```

Remove parity bit from a file:

```
dd if=file conv=parnone of=file.txt
```

Check for (even) parity errors on a file:

```
dd if=file conv=pareven | cmp -x - file
```

To create an image of a Mode-1 CD-ROM, which is a commonly used format for data CD-ROM disks, use a block size of 2048 bytes:

```
dd if=/dev/cd0 of=filename.iso bs=2048
```

Write a filesystem image to a memory stick, padding the end with zeros, if necessary, to a 1MiB boundary:

```
dd if=memstick.img of=/dev/da0 bs=1m conv=noerror,sync
```

SEE ALSO

cp(1), mt(1), recoverdisk(1), tr(1), geom(4), trim(8)

STANDARDS

The **dd** utility is expected to be a superset of the IEEE Std 1003.2 ("POSIX.2") standard. The **files** and **status** operands and the **ascii**, **ebsdic**, **ibm**, **oldascii**, **oldebcdic** and **oldibm** values are extensions to the POSIX standard.

HISTORY

A **dd** command appeared in Version 5 AT&T UNIX.

BUGS

Protection mechanisms in the geom(4) subsystem might prevent the super-user from writing blocks to a disk. Instructions for temporarily disabling these protection mechanisms can be found in the geom(4) man page.