

NAME

dd – convert and copy a file

SYNOPSIS

dd [*OPERAND*]...
dd *OPTION*

DESCRIPTION

Copy a file, converting and formatting according to the operands.

bs=BYTES

read and write up to BYTES bytes at a time (default: 512); overrides **ibs** and **obs**

cbs=BYTES

convert BYTES bytes at a time

conv=CONVS

convert the file as per the comma separated symbol list

count=N

copy only N input blocks

ibs=BYTES

read up to BYTES bytes at a time (default: 512)

if=FILE

read from FILE instead of stdin

iflag=FLAGS

read as per the comma separated symbol list

obs=BYTES

write BYTES bytes at a time (default: 512)

of=FILE

write to FILE instead of stdout

oflag=FLAGS

write as per the comma separated symbol list

seek=N (or **oseek=N**) skip N **obs**-sized output blocks

skip=N (or **iseek=N**) skip N **ibs**-sized input blocks

status=LEVEL

The **LEVEL** of information to print to stderr; 'none' suppresses everything but error messages, 'noxfer' suppresses the final transfer statistics, 'progress' shows periodic transfer statistics

N and **BYTES** may be followed by the following multiplicative suffixes: **c=1**, **w=2**, **b=512**, **kB=1000**, **K=1024**, **MB=1000*1000**, **M=1024*1024**, **xM=M**, **GB=1000*1000*1000**, **G=1024*1024*1024**, and so on for **T**, **P**, **E**, **Z**, **Y**, **R**, **Q**. Binary prefixes can be used, too: **KiB=K**, **MiB=M**, and so on. If **N** ends in 'B', it counts bytes not blocks.

Each **CONV** symbol may be:

ascii from EBCDIC to ASCII

ebcdic from ASCII to EBCDIC

ibm from ASCII to alternate EBCDIC

block pad newline-terminated records with spaces to **cbs**-size

unblock

replace trailing spaces in **cbs**-size records with newline

lcase change upper case to lower case
ucase change lower case to upper case
sparse try to seek rather than write all-NUL output blocks
swab swap every pair of input bytes
sync pad every input block with NULs to `ibs-size`; when used with `block` or `unblock`, pad with spaces rather than NULs
excl fail if the output file already exists
nocreat do not create the output file
notrunc do not truncate the output file
noerror continue after read errors
fdatsync
 physically write output file data before finishing
fsync likewise, but also write metadata

Each FLAG symbol may be:

append append mode (makes sense only for output; `conv=notrunc` suggested)
direct use direct I/O for data
directory
 fail unless a directory
dsync use synchronized I/O for data
sync likewise, but also for metadata
fullblock
 accumulate full blocks of input (`iflag` only)
nonblock
 use non-blocking I/O
noatime
 do not update access time
nocache
 Request to drop cache. See also `oflag=sync`
noctty do not assign controlling terminal from file
nofollow
 do not follow symlinks

Sending a USR1 signal to a running 'dd' process makes it print I/O statistics to standard error and then resume copying.

Options are:

--help display this help and exit
--version
 output version information and exit

AUTHOR

Written by Paul Rubin, David MacKenzie, and Stuart Kemp.

REPORTING BUGS

GNU coreutils online help: <<https://www.gnu.org/software/coreutils/>>
Report any translation bugs to <<https://translationproject.org/team/>>

COPYRIGHT

Copyright © 2023 Free Software Foundation, Inc. License GPLv3+: GNU GPL version 3 or later <<https://gnu.org/licenses/gpl.html>>.

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

SEE ALSO

Full documentation <<https://www.gnu.org/software/coreutils/dd>>
or available locally via: info '(coreutils) dd invocation'