

**DD(I)**

**DD(I)**

**NAME**

`dd` — convert and copy a file

**SYNOPSIS**

`dd` [option=value] ...

**DESCRIPTION**

*Dd* copies the specified input file to the specified output with possible conversions. The standard input and output are used by default. The input and output block size may be specified to take advantage of raw physical I/O.

<i>option</i>	<i>values</i>
<code>if=</code>	input file name; standard input is default
<code>of=</code>	output file name; standard output is default
<code>ibs=</code>	input block size (default 512)
<code>obs=</code>	output block size (default 512)
<code>bs=</code>	set both input and output block size, superseding <i>ibs</i> and <i>obs</i> ; also, if no conversion is specified, it is particularly efficient since no copy need be done
<code>cbs=<i>n</i></code>	conversion buffer size
<code>skip=<i>n</i></code>	skip <i>n</i> input records before starting copy
<code>count=<i>n</i></code>	copy only <i>n</i> input records
<code>conv=ascii</code>	convert EBCDIC to ASCII
<code>ibm</code>	convert ASCII to IBM character set
<code>ebcdic</code>	convert ASCII to EBCDIC
<code>lcase</code>	map alphabetic to lower case
<code>ucase</code>	map alphabetic to upper case
<code>swab</code>	swap every pair of bytes
<code>noerror</code>	do not stop processing on an error
... , ...	several comma-separated conversions

Where sizes are specified, a number of bytes is expected. A number may end with **k**, **b** or **w** to specify multiplication by 1024, 512, or 2 respectively. Also a pair of numbers may be separated by **x** to indicate a product.

*Cbs* is used only if *ascii* or *ebcdic* conversion is specified. In the former case *cbs* characters are placed into the conversion buffer, converted to ASCII, and trailing blanks trimmed and new-line added before sending the line to the output. In the latter case ASCII characters are read into the conversion buffer, converted to EBCDIC, and blanks added to make up an output record of size *cbs*.

After completion, *dd* reports the number of whole and partial input and output blocks.

For example, to read an EBCDIC tape blocked ten 80-byte EBCDIC card images per record into the ASCII file *x*:

```
dd if=/dev/rmt0 of=x ibs=800 cbs=80 conv=ascii,lcase
```

Note the use of raw magtape. *Dd* is especially suited to I/O on the raw physical devices because it allows reading and writing in arbitrary record sizes.

**SEE ALSO**

`cp` (I)

**DD(I)**

**DD(I)**

**BUGS**

The ASCII/EBCDIC conversion tables are taken from the 256 character standard in the CACM Nov, 1968. It is not clear how this relates to real life.

Newlines are inserted only on conversion to ASCII; padding is done only on conversion to EBCDIC. There should be separate options.