dd Raw block copier

CONCEPTS

Danger
dd overwrites whatever of= points at, with no prompt. A wrong device letter destroys that disk. Check lsblk twice before pressing Enter.

if= input file or device
↓ read ibs
dd conv= converts
↓ write obs
of= output file or device

Operands are NAME=VALUE with no dashes and no spaces around the equals sign. Without if= it reads stdin, without of= it writes stdout.

dd copies raw bytes and knows nothing about filesystems. It is just a block copier, which is why it fits images and devices.

BASICS

`dd if=<in> of=<out>`
Copy a file or device

bs=*<size>*
Block size for read and write

count=*<n>*
Copy only n input blocks

skip=*<n>*
Skip n blocks of input first

seek=*<n>*
Skip n blocks of output first

status=progress
Show running transfer stats

status=none
Print only errors

bs=4M
Good default for disks and USB

count=*<n>***B**
Count bytes instead of blocks

skip=*<n>***B**
Skip bytes instead of blocks

kill -USR1 $(pidof dd)
Print stats once, keep copying

Size suffixes: K=1024, M=1024*1024, G=1024^3. kB, MB, GB are powers of 1000. Plain bs=1M means MiB.

WRITING AND FLUSHING

conv=fsync
Flush data and metadata at the end

conv=fdatasync
Flush data only at the end

oflag=direct
Bypass the page cache on write

conv=notrunc
Do not truncate the output file

conv=excl
Fail if the output exists

iflag=fullblock
Fill whole blocks when reading pipes

Without conv=fsync dd can report done while data still sits in the cache. Wait for the prompt, then run sync before unplugging a USB stick.

CONVERSIONS AND ERRORS

conv=noerror,sync
Continue on read errors, pad with zeros

conv=sparse
Skip all-zero blocks, make a sparse file

noerror alone shifts data after a bad block. Always pair it with sync so offsets stay aligned. For failing disks prefer ddrescue.

DISKS AND IMAGES

lsblk -o NAME,SIZE,MODEL, TRAN,MOUNTPOINT
Identify the target device

`dd if=<iso> of=/dev/<usb>` **bs=4M conv=fsync status=progress**
Write an ISO to a USB stick

`dd if=/dev/<disk>` **of=***<disk.img>* **bs=4M status=progress**
Back up a whole disk

`dd if=<disk.img> of=/dev/<disk>` **bs=4M conv=fsync**
Restore a disk image

DISKS AND IMAGES (cont.)

`dd if=/dev/<disk>` **bs=4M | gzip >** *<disk.img.gz>*
Compressed disk image

gunzip -c *<disk.img.gz>* **| dd of=/dev/***<disk>* **bs=4M**
Restore compressed image

`dd if=/dev/<disk>` **of=***<mbr.bin>* **bs=512 count=1**
Save the first sector (MBR)

`dd if=/dev/<part>` **of=***<part.img>* **bs=4M**
Image a single partition

`dd if=<disk.img>` **bs=4M | sha256sum**
Hash the image to verify

Unmount the target first (umount /dev/<usb>*). Use the whole device (/dev/sdb), not a partition (/dev/sdb1), for bootable ISOs.

WIPING AND TEST FILES

`dd if=/dev/zero of=<file>` **bs=1M count=***<n>*
File of n MiB zeros

`dd if=/dev/zero of=<file>` **bs=1 count=0 seek=***<size>*
Sparse file, no data written

`dd if=/dev/zero of=/dev/<disk>` **bs=4M status=progress**
Zero a whole disk

`dd if=/dev/zero of=/dev/<disk>` **bs=1M count=10**
Wipe start of disk (old signatures)

SSDs remap blocks, so overwriting is not a secure erase. Use blkdiscard or the drive's secure erase instead.

BENCHMARKING AND EXTRACTION

`dd if=/dev/zero of=<file>` **bs=1G count=1 oflag=direct**
Rough write speed

`dd if=<file>` **of=/dev/null bs=1M**
Rough read speed