dd Raw block copier

START HERE

lsblk
Find the right device

kill -USR1 $(pidof dd)
Show progress now

dd if=_<in>_ **of=**_<out>_ **bs=4M status=progress**
Copy with progress

sync
Flush writes to disk

dd if=_<iso>_ **of=/dev/**_<usb>_ **bs=4M conv=fsync**
Write an image to USB

dd if=/dev/_<disk>_ **of=**_<img>_ **bs=1M**
Disk to image file

dd if=/dev/zero of=_<file>_ **bs=1M count=100**
Make a 100 MiB file

dd if=_<in>_ **of=**_<out>_ **count=1**
Copy one block only

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

CONCEPTS

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

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

BASICS (cont.)

status=noxfer
Hide the final summary

bs=4M
Good default for disks and USB

ibs=_<size>_ **obs=**_<size>_
Separate read and write sizes

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

watch -n1 kill -USR1 $(pidof dd)
Poll stats on a dd already running

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=sync
Synchronous writes, slow and safe

WRITING AND FLUSHING (cont.)

oflag=direct
Bypass the page cache on write

iflag=direct
Bypass the page cache on read

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

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=_<mbr.bin>_ **of=/dev/**_<disk>_ **bs=446 count=1**
Restore boot code, keep partitions

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

DISKS AND IMAGES (cont.)

cmp _<iso>_ **/dev/**_<usb>_ **-n $(stat -c %s** _<iso>_**)**
Verify what was written

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/urandom of=_<file>_ **bs=1M count=**_<n>_
File of random bytes

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

dd if=/dev/urandom of=/dev/_<disk>_ **bs=4M status=progress**
Overwrite with random data

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

WIPING AND TEST FILES (cont.)

truncate -s _<size>_ _<file>_
Resize or create a sparse file

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

dd if=_<in>_ **of=**_<out>_ **bs=1 skip=**_<off>_ **count=**_<len>_
Cut a byte range from a file

RECIPES

Make a bootable USB stick

```
lsblk -o NAME,SIZE,MODEL,TRAN
sudo umount /dev/sdX*
sudo dd if=distro.iso of=/dev/
sdX bs=4M conv=fsync
status=progress
sync
```

Back up and verify a disk

```
sudo dd if=/dev/sdX bs=4M
status=progress | gzip >
disk.img.gz
# restore later
gunzip -c disk.img.gz | sudo dd
of=/dev/sdY bs=4M conv=fsync
```