smartctl Disk health check

START HERE

`sudo smartctl --scan`
List drives to check

`sudo smartctl -t short <dev>`
Start a quick self-test

`sudo smartctl -i <dev>`
Model, serial, SMART on?

`sudo smartctl -l selftest <dev>`
Read the test result

`sudo smartctl -H <dev>`
Overall health verdict

`sudo smartctl -x <dev>`
Everything the drive reports

`sudo smartctl -A <dev>`
Attribute table

Find drive --scan, lsblk → `-i / -H` **Check health** verdict + attributes → `-t short|long` **Self-test** runs inside the drive → `-l selftest` **Read the log** pass or first bad LBA

CONCEPTS

Run as root. Without it most commands fail with a permission error on the device node.

PASSED only means no attribute has crossed its failure threshold. A drive can still be dying: watch reallocated, pending and uncorrectable sectors.

Device is the whole disk, such as /dev/sda or /dev/nvme0, not a partition like /dev/sda1.

INFORMATION AND HEALTH

`smartctl --scan`
List devices smartctl can see

`smartctl -i <dev>`
Model, serial, firmware, capacity

`smartctl -H <dev>`
Overall health: PASSED or FAILED

`smartctl -A <dev>`
Vendor attributes (NVMe: health log)

`smartctl -a <dev>`
Info, health, attributes, error and test logs
For ATA this adds -c and the selective self-test log too.

`smartctl -x <dev>`
Extended: all of -a plus device stats, SCT data

`smartctl -c <dev>`
Drive SMART capabilities

`smartctl -H -q errorsonly <dev>`
Print only on problems (for scripts)

`smartctl -j -a <dev>`
JSON output for tools like jq

`smartctl -a -q noserial <dev>`
Hide serial number before sharing output

`smartctl -s on <dev>`
Enable SMART on the drive

SELF-TESTS

`smartctl -t short <dev>`
Quick test, usually a few minutes

SELF-TESTS (cont.)

`smartctl -t long <dev>`
Full surface test, minutes to hours

`smartctl -l selftest <dev>`
Show self-test results

`smartctl -t conveyance <dev>`
Check for shipping damage (ATA)

`smartctl -t offline <dev>`
Run offline data collection (ATA)

`smartctl -t select,<start>-<end> <dev>`
Test one LBA range (ATA)

`smartctl -X <dev>`
Abort a running self-test

Self-tests run on the drive itself and are non-captive by default, so the disk stays usable. Check back with -l selftest after the estimated time.

LOGS AND ERRORS

`smartctl -l error <dev>`
Drive error log

`smartctl -l xerror <dev>`
Extended error log, more entries (ATA)

`smartctl -l xselftest <dev>`
Extended self-test log (ATA)

`smartctl -l scterc <dev>`
Show error recovery timeouts (ATA)

`smartctl -l scterc,70,70 <dev>`
Set read/write timeout to 7.0 s
Units are 100 ms. Usually resets at power cycle; useful for RAID members.

DEVICES AND NVME

`smartctl -d <type> <dev>`
Force the device type

`smartctl -d sat <dev>`
USB or SATA bridge (SCSI-to-ATA)
If auto-detect fails on a USB enclosure, try sat first, then usbjmicron, usbcypress or usbprolific.

`smartctl -d scsi <dev>`
Treat as SCSI/SAS drive

`smartctl -d ata <dev>`
Treat as ATA/SATA drive

`smartctl -d nvme <dev>`
NVMe device explicitly

DEVICES AND NVME (cont.)

`smartctl -a /dev/nvme0`
NVMe health log: spare, wear, temperature

`smartctl -d megaraid,<n> /dev/sda`
Disk n behind a MegaRAID controller

NVMe paths: /dev/nvme0 is the controller, /dev/nvme0n1 a namespace. Both normally work with -a.

READING ATTRIBUTES

Attributes worth watching (ATA)

ID / name	Concern when
`5 Reallocated_Sector_Ct`	raw above 0 and rising
`187 Reported_Uncorrect`	raw above 0
`197 Current_Pending_Sector`	raw above 0 (unreadable sectors)
`198 Offline_Uncorrectable`	raw above 0
`199 UDMA_CRC_Error_Count`	rising: suspect cable or port
`194 Temperature_Celsius`	sustained above ~55 C

Health log fields (NVMe)

Field	Concern when
Critical Warning	not 0x00
Available Spare	near Threshold
Percentage Used	approaching 100%
Media and Data Integrity Errors	above 0
Temperature	sustained above ~70 C

In the -A table, a normalized VALUE falling to its THRESH means failing; the RAW column is vendor-specific and often counts something other than you expect.

A rising pending-sector count that later turns into reallocations is the classic sign of a failing disk. Back up first, test later.

SCRIPTING AND OPTIONS

`smartctl -n standby <dev>`
Skip the check if the drive is asleep
Prevents spinning up a sleeping drive; exits with status 2.

`smartctl -o on <dev>`
Enable automatic offline testing (ATA)

`smartctl -V`
Show version

Exit status bits (add up)

Bit	Value	Meaning
0	1	Command line did not parse
1	2	Device open failed or in low-power mode
2	4	SMART command failed or bad checksum
3	8	Disk failing
4	16	Prefail attribute at/below threshold
5	32	Attribute was below threshold in the past
6	64	Error log has entries
7	128	Self-test log has errors

Test one bit in a script with shell arithmetic, for example $(($? & 8)) for a failing disk.

RECIPES

Quick health check of all drives

```
for d in $(sudo smartctl --scan
| awk '{print $1}'); do
  echo "== $d"
  sudo smartctl -H -q errorsonly
"$d"
done
```

Run a long test and read the verdict

```
sudo smartctl -t long /dev/sda
# note the estimated finish
time, then wait
sudo smartctl -l selftest /dev/
sda
sudo smartctl -a /dev/sda | less
```

Check a USB disk

```
lsblk -o NAME,MODEL,SIZE,TRAN
sudo smartctl -d sat -a /dev/sdb
# if rejected, try -d
usbjmicron / usbcypress /
usbprolific
```

Extract fields as JSON

```
sudo smartctl -j -a /dev/nvme0 \
  | jq .
nvme_smart_health_information_log
```