Engineer

S.M.A.R.T.

a disk's built-in self-check that warns of coming failure

What it is

S.M.A.R.T. stands for Self-Monitoring, Analysis and Reporting Technology; smartmontools and its manuals write it without the dots, as SMART. It is a record a disk keeps about itself, in its own firmware, while it works: counts of events, readings of its condition and the results of tests it has run. The operating system can ask the drive for that record at any time. smartmontools, the toolkit this page draws on, says most current disks in the ATA/SATA, SCSI/SAS and NVMe families implement this technology. The package ships two programs: smartctl, which reads and tests one disk when you run it, and smartd, a daemon that keeps watch in the background.

What a disk can report depends on the command set it speaks. An ATA or SATA disk keeps a table of numbered attributes, from 1 to 253, each with a name; attribute 12, for example, counts how many times the disk has been powered up. Every attribute carries a raw value, the actual count or reading, and a normalized value between 1 and 254 that the vendor’s firmware works out from the raw one by its own formula. Next to those sit the worst normalized value the disk has recorded in its life and a threshold chosen by the manufacturer. An attribute whose normalized value drops to its threshold or below has failed.

Each attribute is one of two types. A pre-failure attribute at or below its threshold says the disk is about to fail. An old-age attribute, also called a usage attribute, at or below its threshold says the disk has reached the end of its life through ordinary wear. The smartctl manual stresses that the Pre-fail label is no cause for worry by itself: it only carries that meaning once the value has crossed the threshold.

Above the table sits one verdict, the health status. An ATA disk answers a command named SMART RETURN STATUS with pass or fail. A SCSI or SAS disk reports through an exceptions log page and its sense data, and an NVMe drive sets bits in the Critical Warning byte of its SMART/Health Information log. Disks can also test themselves on request: a short test, usually done in under ten minutes, a long one that takes anything from tens of minutes to several hours, and on ATA disks a conveyance test aimed at damage from transport. The outcome of each lands in a self-test log on the disk.

Use the concept in a decision

Put “What is S.M.A.R.T.?” to work in a real decision. We can help relate the concept to the system you are building or buying. The laptop, the printer, the login and the mailbox, sorted quickly and in plain language, without anyone feeling foolish.

Explore our services Get in touch →

Test the assumptions behind the term

Use the explanation and linked work to frame questions for your project. We can examine the assumptions with you before a tool is chosen. Established a Technical Support department — 10,000 clients served by a support department built from scratch

Established a Technical Support department Get in touch →

Why it matters

A disk sees its own trouble before anyone else does. It knows which of its sectors no longer read, how often it has been switched on and how hot it runs. S.M.A.R.T. is the channel through which that knowledge reaches the people responsible for the data, and smartmontools describes the payoff modestly: in many cases, it says, the tools give “advanced warning of disk degradation and failure”. Many cases is not all of them, and much of this page is about that difference.

The strongest signal is the verdict. When smartctl reports a failing health status, its manual says the disk has either failed already or expects to fail within the next 24 hours, and tells the reader to collect the full report and move the data somewhere safe without delay. Weaker signals come earlier. A pending sector is one the disk wants to mark as bad and swap for a spare, typically because a read from it failed; a non-zero count means some data on the disk is no longer readable. The disk counts those sectors in attribute 197, and in attribute 198 it counts sectors that neither an offline scan nor a self-test managed to read.

None of this helps if nobody looks. smartd switches S.M.A.R.T. monitoring on for ATA disks, polls the disks every 30 minutes by default, reports errors and attribute changes to syslog and can send a warning by email. It can also start self-tests on a schedule, so no cron job is needed for that part. The project is old and

still moving: it marked the 20th anniversary of its first release in October 2022, released version 7.5 on 30 April 2025, and moved its development from SourceForge to GitHub on 1 June 2025. It runs on Linux, the BSDs, macOS, Solaris and Windows, among others.

What it looks like in practice

Take a new server with a SATA disk at `/dev/sda`. Three commands give the picture:

```
smartctl -i /dev/sda
smartctl -H /dev/sda
smartctl -A /dev/sda
```

The first prints the model, serial number and firmware, whether the drive supports S.M.A.R.T. and whether it is switched on, and whether the drive is listed in the smartmontools drive database. The second prints the verdict. The third prints the attribute table: VALUE and WORST hold the normalized numbers, THRESH the threshold, TYPE says Pre-fail or Old_age, RAW_VALUE holds the raw count, and WHEN_FAILED says `FAILING_NOW`, `In_the_past` or a dash for an attribute that has never failed. When you want everything in one go, `smartctl -x` prints all the information the tool can read, S.M.A.R.T. and otherwise, with the table in a denser layout.

A long self-test is the more thorough version of the short one:

```
smartctl -t long /dev/sda
smartctl -l selftest /dev/sda
```

Unless `-C` asks for captive mode, the test runs inside the disk while the system carries on with its work. It can slow the drive down, heavy traffic from the operating system makes it take longer, and only one test runs at a time. The manual adds that a shutdown in the middle should do no harm, since the test is either aborted or resumed. The second command reads the self-test log once the estimated time has passed.

Scripts should read the exit status rather than the text. smartctl builds it from eight bits, and the manual's own example tests bit 3, which is set when the disk reports that it is failing:

```
smartctl -q silent -H /dev/sda
smartstat=$(($? & 8))
```

`-q silent` prints nothing, so the exit status is the only answer. A non-zero status does not always mean a dying disk: bit 6 alone, for instance, means only that the error log holds entries.

For the long run, smartd reads its rules from `smartd.conf`. One line can cover every disk the scan finds:

```
DEVICESCAN -a -s (S/../../../02|L/../../6/03) -W 2,40,45 \
  -m ops@example.com -M test
```

`-a` turns on the standard set: the health status, the error and self-test logs, failed usage attributes, changes in attribute values and, on ATA disks, the counts in attributes 197 and 198. The `-s` pattern starts a short test every day between 02:00 and 03:00 and a long one every Saturday between 03:00 and 04:00. `-W 2,40,45` reports a temperature change of 2 degrees, logs a note at 40 °C and raises a critical warning, by syslog and email, at 45 °C. `-m` names the recipient, and `-M test` sends one test message when smartd starts. After an edit, `killall -HUP smartd` makes the daemon read the file again.

Numbers like these matter most while the disk still answers. Our portfolio entry on recovering data from failed storage starts after that point: between 2003 and 2014, people brought in memory cards, hard drives, SSDs and RAID arrays that had already failed, and the method had to suit the particular kind of medium.

Where it goes wrong

A passing verdict is read as a clean bill. The health status only says the disk has not declared itself failed. Attributes can be at or below their thresholds now or in the past, and pending sectors can already hold unreadable data, while the verdict

still reads as passed. The fix is to read the attribute table and the self-test log as well, let smartd report attributes 197 and 198, and keep backups that do not rely on the disk's opinion of itself.

Raw values are taken at face value. The standard does not say how a raw value maps to a physical unit. The smartctl manual mentions a Hitachi disk that counts power-on time in minutes rather than hours, and IBM disks that keep three temperatures in their raw values. Since revision 4 of ATA/ATAPI-4 the meaning of the attributes has been left entirely to vendors, and solid-state drives give some numbers other meanings, so the printed name can be wrong unless the drive is in the database. The fix is to compare a disk with its own history, not with a different model, and to check what `-i` says about the database before trusting a name.

`-a` **is used on an ATA disk.** It was the classic way to see everything, but the manual no longer recommends it for ATA, because it leaves out the options that need 48-bit ATA commands, such as the extended error log. The fix is to use `-x` .

A RAID controller hides the disks. The operating system sees one logical drive, which has nothing useful to report, and smartd's automatic scan on Linux leaves out disks behind RAID controllers altogether. The fix is to address each physical disk through the controller, with the matching device type: `-d megaraid,N` reaches disk N behind a MegaRAID or Dell PERC controller on Linux and FreeBSD, and a line such as `/dev/sda -d megaraid,0 -a` in `smartd.conf` puts that disk under watch.

A USB enclosure sits in the way. USB storage speaks the SCSI command set to the computer but nearly always holds an ATA or SATA disk or flash memory. The manual names bridge firmware, along with RAID controllers, as a layer that can leave the health status unknown, and smartctl then warns and falls back to comparing pre-failure attributes with their thresholds. The fix is to try `-d sat` , which wraps the ATA commands in the SCSI-to-ATA translation standard; `-d sat,12` selects the shorter of its two command forms.

The warning email never arrives. smartd sends mail with the system's `mail` command, which has to be on the path of the environment smartd was started from. If it is missing, the warning exists only in syslog. The fix is `-M test` , then checking that the test message has actually landed in an inbox someone reads; `-M exec` can point smartd at another mailer or a script.

A test schedule never fires. In the `-s` pattern, month, day of the month and hour each take two digits. A pattern such as `S/../../2` never matches, and the manual warns about exactly that. The fix is to write `02`, and to read the log after the first night to see that a test ran.

A broken edit goes unnoticed. When smartd reloads its configuration on a hang-up signal and finds a syntax error, it prints the error and carries on with the old rules, as though the signal never came. The fix is to read the log after every reload, or restart the service, since smartd refuses to start at all with a faulty file.

Temperature goes unwatched. All temperature reports are off by default, and `-a` does not turn them on. The fix is to add `-W` with limits that suit the hardware.

A check wakes a sleeping disk. A disk spun down to save power will spin up when smartctl talks to it. The fix, for scripts that run often, is `-n standby`, which makes smartctl exit with status 2 instead of waking the disk, or `-n standby,0` to treat that case as success.

Further reading

- smartmontools — the project's home page: both programs, supported systems and release news.
- smartctl(8), manual page source — attributes, thresholds, health status, self-tests, device types, exit bits.
- smartd(8), manual page source — the daemon: polling, syslog, signals and how it scans for disks.
- smartd.conf(5), manual page source — directives for monitoring, test schedules, temperature limits and email.

Glossary

Choose for your actual constraints

Describe the device or service, the observed symptoms and the data that must be preserved. We can establish the safest useful diagnostic step.

See what it costs Get in touch →

Make the terminology practical

Ask how “What is S.M.A.R.T.?” applies to your organisation. A concrete example from your current system is enough to start the discussion.

Explore our services

welcome@engineer.company +45 50 10 61 84 Copenhagen, Denmark

Take this page with you

Contact card

Carefully Crafted With Love by D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – present · Engineer ApS · All rights reserved.

Printed from https://engineer.company/glossary/smart/