NAME

smartctl − Control and Monitor Utility for SMART Disks

SYNOPSIS

smartctl [options] device

DESCRIPTION

[This man page is generated for the Windows version of smartmontools. It does not contain info specific to other platforms.]

smartctl controls the Self-Monitoring, Analysis and Reporting Technology (SMART) system built into most ATA/SATA and SCSI/SAS hard drives and solid-state drives. The purpose of SMART is to monitor the reliability of the hard drive and predict drive failures, and to carry out different types of drive self-tests. **smartctl** also supports some features not related to SMART. This version of **smartctl** is compatible with ACS-3, ACS-2, ATA8-ACS, ATA/ATAPI-7 and earlier standards (see **REFERENCES** below).

smartctl also provides support for polling TapeAlert messages from SCSI tape drives and changers.

The user must specify the device to be controlled or interrogated as the final argument to **smartctl**. The command set used by the device is often derived from the device path but may need help with the '−d' option (for more information see the section on "ATA, SCSI command sets and SAT" below). Device paths are as follows:

WINDOWS:

Use the forms **"/dev/sd[a−z]"** for IDE/(S)ATA and SCSI disks "\\.\PhysicalDrive[0−25]" (where "a" maps to "0"). Use **"/dev/sd[a−z][a−z]"** for "\\.\PhysicalDrive[26−...]". These disks can also be referred to as **"/dev/pd[0−255]"** for "\\.\PhysicalDrive[0−255]". ATA disks can also be referred to as **"/dev/hd[a−z]"** for "\\.\PhysicalDrive[0−25]". Use one the forms **"/dev/tape[0−255]"**, **"/dev/st[0−255]"**, or **"/dev/nst[0−255]"** for SCSI tape drives "\\.\Tape[0−255]".

Alternatively, drive letters **"X:"** or **"X:\"** may be used to specify the ('basic') disk behind a mounted partition. This does not work with 'dynamic' disks.

For disks behind 3ware 9000 controllers use **"/dev/sd[a−z],N"** where N specifies the disk number (3ware 'port') behind the controller providing the logical drive ('unit') specified by **"/dev/sd[a−z]"**. Alternatively, use **"/dev/tw_cli/cx/py"** for controller x, port y to run the 'tw_cli' tool and parse the output. This provides limited monitoring ('−i', '−c', '−A' below) if SMART support is missing in the driver. Use**"/de v/tw_cli/stdin"** or **"/dev/tw_cli/clip"** to parse CLI or 3DM output from standard input or clipboard. The option '−d 3ware,N' is not necessary on Windows.

For disks behind an Intel ICHxR controller with RST driver use **"/dev/csmi[0−9],N"** where N specifies the port behind the logical scsi controller "\\.\Scsi[0−9]:".

For SATA or SAS disks behind an Areca controller use **"/dev/arcmsr[0−9]"**, see '−d areca,N[/E]' below.

Use the forms **"/dev/nvme[0−9]"** (broadcast namespace) or **"/dev/nvme[0−9]n[1−9]"** (specific namespace 1−9) for first, second, ..., NVMe device. Alternatively use the forms **"/dev/nvmes[0−9][n[1−9]]"** for NVMe devices behind the logical scsi controller "\\.\Scsi[0−9]:". Both forms require a NVMe driver which supports NVME_PASS_THROUGH_IOCTL.

[NEW EXPERIMENTAL SMARTCTL FEATURE] Use the forms **"/dev/sd[...]"** or **"/dev/pd[...]"** (see above) for NVMe devices behind Windows 10 NVMe driver (stornvme.sys).

The prefix **"/dev/"** is optional.

if '−' is specified as the device path, **smartctl** reads and interprets it's own debug output from standard input. See '−r ataioctl' below for details.

smartctl guesses the device type if possible. If necessary, the '−d' option can be used to override this guess.

Note that the printed output of **smartctl** displays most numerical values in base 10 (decimal), but some values are displayed in base 16 (hexadecimal). To distinguish them, the base 16 values are always displayed with a leading **"0x"**, for example: "0xff". This man page follows the same convention.

OPTIONS

The options are grouped below into several categories. **smartctl** will execute the corresponding commands in the order: INFORMATION, ENABLE/DISABLE, DISPLAY DATA, RUN/ABORT TESTS.

SHOW INFORMATION OPTIONS:

−h, −−help, −−usage

Prints a usage message to STDOUT and exits.

−V, −−version, −−copyright, −−license

Prints version, copyright, license, home page and SVN revision information for your copy of **smartctl** to STDOUT and then exits.

−i, −−info

Prints the device model number, serial number, firmware version, and ATA Standard version/revision information. Says if the device supports SMART, and if so, whether SMART support is currently enabled or disabled. If the device supports Logical Block Address mode (LBA mode) print current user drive capacity in bytes. (If drive is has a user protected area reserved, or is "clipped", this may be smaller than the potential maximum drive capacity.) Indicates if the drive is in the smartmontools database (see '−v' options below). If so, the drive model family may also be printed. If '−n' (see below) is specified, the power mode of the drive is printed.

[NVMe] For NVMe devices the information is obtained from the Identify Controller and the Identify Namespace data structure.

−−identify[=[w][nvb]]

[ATA only] Prints an annotated table of the IDENTIFY DEVICE data. By default, only valid words (words not equal to 0x0000 or 0xffff) and nonzero bits and bit fields are printed. This can be changed by the optional argument which consists of one or two characters from the set 'wnvb'. The character 'w' enables printing of all 256 words. The character 'n' suppresses printing of bits, 'v' enables printing of all bits from valid words, 'b' enables printing of all bits. For example '−−identify=n' (valid words, no bits) produces the shortest output and '−−identify=wb' (all words, all bits) produces the longest output.

−a, −−all

Prints all SMART information about the disk, or TapeAlert information about the tape drive or changer. For ATA devices this is equivalent to
'−H −i −c −A −l error −l selftest −l selective'
and for SCSI, this is equivalent to
'−H −i −A −l error −l selftest'.
For NVMe, this is equivalent to
'−H −i −c −A −l error'.
Note that for ATA disks this does **not** enable the non-SMART options and the SMART options which require support for 48-bit ATA commands.

−x, −−xall

Prints all SMART and non-SMART information about the device. For ATA devices this is equivalent to
'−H −i −g all −g wcreorder −c −A −f brief −l xerror,error −l xselftest,selftest −l selective −l directory −l scttemp −l scterc −l devstat −l defects −l sataphy'.
and for SCSI, this is equivalent to
'−H −i −g all −A −l error −l selftest −l background −l sasphy'.
For NVMe, this is equivalent to
'−H −i −c −A −l error'.

−−scan Scans for devices and prints each device name, device type and protocol ([ATA] or [SCSI]) info. May be used in conjunction with '−d TYPE' to restrict the scan to a specific TYPE. See also info about platform specific device scan and the **DEVICESCAN** directive on **smartd**(8) man page.

−−scan−open
Same as −−scan, but also tries to open each device before printing device info. The device open may change the device type due to autodetection (see also '−d test').

This option can be used to create a draft **smartd.conf** file. All options after '−−' are appended to each output line. For example:

```
smartctl −−scan−open −− −a −W 4,45,50 −m admin@work > smartd.conf
```

Multiple '−d TYPE' options may be specified with '−−scan[−open]' to combine the scan results of more than one TYPE.

−g NAME, −−get=NAME
Get non-SMART device settings. See '−s, −−set' below for further info.

RUN-TIME BEHAVIOR OPTIONS:

−j, −−json[=cgiosuv]
[NEW EXPERIMENTAL SMARTCTL FEATURE] Enables JSON output mode.

The output could be modified or enhanced by the optional argument which consists of one or more characters from the set 'cgiosuv':
'c': Outputs **c**ompact format without extra spaces and newlines. By default, output is pretty-printed.
'g': Outputs JSON structure as single assignments to allow the usage of **g**rep. Each assignment reflects the absolute path of a value. The syntax is compatible with**gr on**:
'json.KEY1[INDEX2].KEY3 = VALUE;'.
'o': Includes the full **o**riginal plaintext **o**utput of **smartctl** as a JSON array 'smartctl.output[]'.
's': Outputs JSON object elements **s**orted by key. By default, object elements are ordered as generated internally.
'v': Enables **v**erbose output of possible unsafe integers. If specified, values which may exceed JSON safe integer (53-bit) range are always output as a number (with some 'KEY') and a string ('KEY_s'), regardless of the actual value. Values which may exceed 64-bit range are also output as a little endian byte array ('KEY_le'). By default, the additional elements are only output if the value actually exceeds the range.

The following two arguments are primarily indented for development:
'i': Includes lines from the plaintext output which print info already **i**mplemented for JSON output. The lines appear as objects with key 'smartctl_NNNN_i'.
'u': Includes lines from the plaintext output which print info still **u**nimplemented for JSON output. The lines appear as objects with key 'smartctl_NNNN_u'.

−q TYPE, −−quietmode=TYPE
Specifies that **smartctl** should run in one of the quiet modes described here. The valid arguments to this option are:

errorsonly − only print: For the '−l error' option, if nonzero, the number of errors recorded in the SMART error log and the power-on time when they occurred; For the '−l selftest' option, errors recorded in the device self-test log; For the '−H' option, SMART "disk failing" status or device Attributes (pre-failure or usage) which failed either now or in the past; For the '−A' option, device Attributes (pre-failure or usage) which failed either now or in the past.

silent − print no output. The only way to learn about what was found is to use the exit status of **smartctl** (see EXIT STATUS below).

noserial − Do not print the serial number of the device.

−d TYPE, −−device=TYPE

Specifies the type of the device. The valid arguments to this option are:

auto − attempt to guess the device type from the device name or from controller type info provided by the operating system or from a matching USB ID entry in the drive database. This is the default.

test − prints the guessed TYPE, then opens the device and prints the (possibly changed) TYPE name and then exits without performing any further commands.

ata − the device type is ATA. This prevents **smartctl** from issuing SCSI commands to an ATA device.

scsi − the device type is SCSI. This prevents **smartctl** from issuing ATA commands to a SCSI device.

nvme[,NSID] − the device type is NVM Express (NVMe). The optional parameter NSID specifies the namespace id (in hex) passed to the driver. Use 0xffffffff for the broadcast namespace id. The default for NSID is the namespace id addressed by the device name.

sat[,auto][,N] − the device type is SCSI to ATA Translation (SAT). This is for ATA disks that have a SCSI to ATA Translation Layer (SATL) between the disk and the operating system. SAT defines two ATA PASS THROUGH SCSI commands, one 12 bytes long and the other 16 bytes long. The default is the 16 byte variant which can be overridden with either '−d sat,12' or '−d sat,16'.

If '−d sat,auto' is specified, device type SAT (for ATA/SATA disks) is only used if the SCSI INQUIRY data reports a SATL (VENDOR: "ATA "). Otherwise device type SCSI (for SCSI/SAS disks) is used.

usbcypress − this device type is for ATA disks that are behind a Cypress USB to PATA bridge. This will use the ATACB proprietary scsi pass through command. The default SCSI operation code is 0x24, but although it can be overridden with '−d usbcypress,0xN', where N is the scsi operation code, you’re running the risk of damage to the device or filesystems on it.

usbjmicron[,p][,x][,PORT] − this device type is for SATA disks that are behind a JMicron USB to PATA/SATA bridge. The 48-bit ATA commands (required e.g. for '−l xerror', see below) do not work with all of these bridges and are therefore disabled by default. These commands can be enabled by '−d usbjmicron,x'. If two disks are connected to a bridge with two ports, an error message is printed if no PORT is specified. The port can be specified by '−d usbjmicron[,x],PORT' where PORT is 0 (master) or 1 (slave). This is not necessary if the device uses a port multiplier to connect multiple disks to one port. The disks appear under separate /dev/ice names then. CAUTION: Specifying ',x' for a device which does not support it results in I/O errors and may disconnect the drive. The same applies if the specified PORT does not exist or is not connected to a disk.

The Prolific PL2507/3507 USB bridges with older firmware support a pass-through command similar to JMicron and work with '−d usbjmicron,0'. Newer Prolific firmware requires a modified command which can be selected by '−d usbjmicron,p'. Note that this does not yet support the SMART status command.

usbprolific − this device type is for SATA disks that are behind a Prolific PL2571/2771/2773/2775 USB to SATA bridge.

usbsunplus − this device type is for SATA disks that are behind a SunplusIT USB to SATA bridge.

sntjmicron[,NSID] − [NEW EXPERIMENTAL SMARTCTL FEATURE] this device type is for NVMe disks that are behind a JMicron USB to NVMe bridge. The optional parameter NSID specifies the namespace id (in hex) passed to the driver. The default namespace id is the broadcast namespace id (0xffffffff).

aacraid,H,L,ID − [Linux, Windows and Cygwin only] the device consists of one or more SCSI/SAS or SATA disks connected to an AacRaid controller. The non-negative integers H,L,ID (Host number, Lun, ID) denote which disk on the controller is monitored. Use syntax such as:

smartctl −a −d aacraid,0,0,2 /dev/sda
smartctl −a −d aacraid,1,0,4 /dev/sdb

Option '−d sat,auto+...' is implicitly enabled to detect SATA disks. Use '−d scsi+aacraid,H,L,ID' to disable it.

On Windows, the device name parameter /dev/sdX is ignored if '−d aacraid' is specified.

areca,N − [FreeBSD, Linux, Windows and Cygwin only] the device consists of one or more SATA disks connected to an Areca SATA RAID controller. The positive integer N (in the range from 1 to 24 inclusive) denotes which disk on the controller is monitored. On Windows and Cygwin use syntax such as:
smartctl −a −d areca,2 /dev/arcmsr0
smartctl −a −d areca,3 /dev/arcmsr1
The first line above addresses the second disk on the first Areca RAID controller. The second line addresses the third disk on the second Areca RAID controller.

Important: the Areca controller must have firmware version 1.46 or later. Lower-numbered firmware versions will give (harmless) SCSI error messages and no SMART information.

areca,N/E − [FreeBSD, Linux, Windows and Cygwin only] the device consists of one or more SATA or SAS disks connected to an Areca SAS RAID controller. The integer N (range 1 to 128) denotes the channel (slot) and E (range 1 to 8) denotes the enclosure. Important: This requires Areca SAS controller firmware version 1.51 or later.

intelliprop,N[+TYPE] − [NEW EXPERIMENTAL SMARTCTL FEATURE] the device consists of multiple ATA disks connected to an Intelliprop controller. The integer N is the port number from 0 to 3 of the ATA drive to be targeted. The TYPE can be ata(default), sat, or a USB controller listed above. Note: if a type of ATA does not work, try a type of sat. Use syntax such as:
smartctl −a −d intelliprop,1 /dev/sda (under Linux)
smartctl −a −d intelliprop,1+sat /dev/sda (under Linux)
WARNING: The disks are selected by write commands to the ATA Device Vendor Specific Log at address 0xc0. Using this option with other devices may have undesirable side effects.

−T TYPE, −−tolerance=TYPE

[ATA only] Specifies how tolerant **smartctl** should be of ATA and SMART command failures.

The behavior of **smartctl** depends upon whether the command is "**optional**" or "**mandatory**". Here "**mandatory**" means "required by the ATA Specification if the device implements the SMART command set" and "**optional**" means "not required by the ATA Specification even if the device implements the SMART command set." The "**mandatory**" ATA and SMART commands are: (1) ATA IDENTIFY DEVICE, (2) SMART ENABLE/DISABLE ATTRIBUTE AUTOSAVE, (3) SMART ENABLE/DISABLE, and (4) SMART RETURN STATUS.

The valid arguments to this option are:

normal − exit on failure of any **mandatory** SMART command, and ignore all failures of **optional** SMART commands. This is the default. Note that on some devices, issuing unimplemented optional SMART commands doesn't cause an error. This can result in misleading**smartctl** messages such as "Feature X not implemented", followed shortly by "Feature X: enabled". In most such cases, contrary to the final message, Feature X is **not** enabled.

conservative − exit on failure of any **optional** SMART command.

permissive − ignore failure(s) of **mandatory** SMART commands. This option may be given more than once. Each additional use of this option will cause one more additional failure to be ignored. Note that the use of this option can lead to messages like "Feature X not supported", followed shortly by "Feature X enable failed". In a few such cases, contrary to the final message, Feature X **is** enabled.

verypermissive − equivalent to giving a large number of '−T permissive' options: ignore failures of **any number** of **mandatory** SMART commands. Please see the note above.

−b TYPE, −−badsum=TYPE

[ATA only] Specifies the action **smartctl** should take if a checksum error is detected in the: (1) Device Identity Structure, (2) SMART Self-Test Log Structure, (3) SMART Attribute Value Structure, (4) SMART Attribute Threshold Structure, or (5) ATA Error Log Structure.

The valid arguments to this option are:

warn − report the incorrect checksum but carry on in spite of it. This is the default.

exit − exit **smartctl**.

ignore − continue silently without issuing a warning.

−r TYPE, −−report=TYPE

Intended primarily to help **smartmontools** developers understand the behavior of **smartmontools** on non-conforming or poorly conforming hardware. This option reports details of**smartctl** transactions with the device. The option can be used multiple times. When used just once, it shows a record of the ioctl() transactions with the device. When used more than once, the detail of these ioctl() transactions are reported in greater detail. The valid arguments to this option are:

ioctl − report all ioctl() transactions.

ataioctl − report only ioctl() transactions with ATA devices.

scsiioctl − report only ioctl() transactions with SCSI devices. Invoking this once shows the SCSI commands in hex and the corresponding status. Invoking it a second time adds a hex listing of the first 64 bytes of data send to, or received from the device.

nvmeioctl − report only ioctl() transactions with NVMe devices.

Any argument may include a positive integer to specify the level of detail that should be reported. The argument should be followed by a comma then the integer with no spaces. For example, *ataioctl,2* The default level is 1, so '−r ataioctl,1' and '−r ataioctl' are equivalent.

For testing purposes, the output of '−r ataioctl,2' can later be parsed by **smartctl** itself if '−' is used as device path argument. The ATA command input parameters, sector data and return values are reconstructed from the debug report read from stdin. Then **smartctl** internally simulates an ATA device with the same behaviour. This is does not work for SCSI devices yet.

−n POWERMODE[,STATUS], −−nocheck=POWERMODE[,STATUS]

[ATA only] Specifies if **smartctl** should exit before performing any checks when the device is in a low-power mode. It may be used to prevent a disk from being spun-up by **smartctl**. The power mode is ignored by default.

Note: If this option is used it may also be necessary to specify the device type with the '−d' option. Otherwise the device may spin up due to commands issued during device type autodetection.

By default, exit status 2 is returned if the device is in one of the specified low-power modes. This status is also returned if the device open or identification failed (see EXIT STATUS below).

[NEW EXPERIMENTAL SMARTCTL FEATURE] The optional STATUS parameter allows to override this default. STATUS is an integer in the range from 0 to 255 inclusive. For example use '−n standby,0' to return success if a device is in SLEEP or STANDBY mode. Use '−n standby,3' to return a unique exit status in this case.

The valid arguments to this option are:

never − check the device always, but print the power mode if '−i' is specified.

sleep[,STATUS] − check the device unless it is in SLEEP mode.

standby[,STATUS] − check the device unless it is in SLEEP or STANDBY mode. In these modes most disks are not spinning, so if you want to prevent a disk from spinning up, this is probably what you want.

idle[,STATUS] − check the device unless it is in SLEEP, STANDBY or IDLE mode. In the IDLE

state, most disks are still spinning, so this is probably not what you want.

SMART FEATURE ENABLE/DISABLE COMMANDS:

Note: if multiple options are used to both enable and disable a feature, then **both** the enable and disable commands will be issued. The enable command will always be issued **before** the corresponding disable command.

−s VALUE, −−smart=VALUE
Enables or disables SMART on device. The valid arguments to this option are *on* and *off*.

[ATA] Note that the ATA commands SMART ENABLE/DISABLE OPERATIONS were declared obsolete in ATA ACS-4 Revision 10 (Nov 2015).

[SCSI tape drive or changer] It is not necessary (or useful) to enable SMART to see the TapeAlert messages.

−o VALUE, −−offlineauto=VALUE
[ATA only] Enables or disables SMART automatic offline test, which scans the drive every four hours for disk defects. This command can be given during normal system operation. The valid arguments to this option are *on* and *off*.

Note that the SMART automatic offline test command is listed as "Obsolete" in every version of the ATA and ATA/ATAPI Specifications. It was originally part of the SFF-8035i Revision 2.0 specification, but was never part of any ATA specification. However it is implemented and used by many vendors. You can tell if automatic offline testing is supported by seeing if this command enables and disables it, as indicated by the 'Auto Offline Data Collection' part of the SMART capabilities report (displayed with '−c').

SMART provides **three** basic categories of testing. The **first** category, called "online" testing, has no effect on the performance of the device. It is turned on by the '−s on' option.

The **second** category of testing is called "offline" testing. This type of test can, in principle, degrade the device performance. The '−o on' option causes this offline testing to be carried out, automatically, on a regular scheduled basis. Normally, the disk will suspend offline testing while disk accesses are taking place, and then automatically resume it when the disk would otherwise be idle, so in practice it has little effect. Note that a one-time offline test can also be carried out immediately upon receipt of a user command. See the '−t offline' option below, which causes a one-time offline test to be carried out immediately.

The choice (made by the SFF-8035i and ATA specification authors) of the word *testing* for these first two categories is unfortunate, and often leads to confusion. In fact these first two categories of online and offline testing could have been more accurately described as online and offline **data collection**.

The results of this automatic or immediate offline testing (data collection) are reflected in the values of the SMART Attributes. Thus, if problems or errors are detected, the values of these Attributes will go below their failure thresholds; some types of errors may also appear in the SMART error log. These are visible with the '−A' and '−l error' options respectively.

Some SMART attribute values are updated only during off-line data collection activities; the rest are updated during normal operation of the device or during both normal operation and off-line testing. The Attribute value table produced by the '−A' option indicates this in the UPDATED column. Attributes of the first type are labeled "Offline" and Attributes of the second type are labeled "Always".

The **third** category of testing (and the *only* category for which the word 'testing' is really an appropriate choice) is "self" testing. This third type of test is only performed (immediately) when a command to run it is issued. The '−t' and '−X' options can be used to carry out and abort such self-tests; please see below for further details.

Any errors detected in the self testing will be shown in the SMART self-test log, which can be

examined using the '−l selftest' option.

Note: in this manual page, the word **"Test"** is used in connection with the second category just described, e.g. for the "offline" testing. The words **"Self-test"** are used in connection with the third category.

−S VALUE, −−saveauto=VALUE

[ATA] Enables or disables SMART autosave of device vendor-specific Attributes. The valid arguments to this option are *on* and *off*. Note that this feature is preserved across disk power cycles, so you should only need to issue it once.

The ATA standard does not specify a method to check whether SMART autosave is enabled. Unlike SCSI (below), smartctl is unable to print a warning if autosave is disabled.

Note that the ATA commands SMART ENABLE/DISABLE AUTOSAVE were declared obsolete in ATA ACS-4 Revision 10 (Nov 2015).

[SCSI] For SCSI devices this toggles the value of the Global Logging Target Save Disabled (GLTSD) bit in the Control Mode Page. Some disk manufacturers set this bit by default. This prevents error counters, power-up hours and other useful data from being placed in non-volatile storage, so these values may be reset to zero the next time the device is power-cycled. If the GLTSD bit is set then 'smartctl −a' will issue a warning. Use*on* to clear the GL TSD bit and thus enable saving counters to non-volatile storage. For extreme streaming-video type applications you might consider using *off* to set the GLTSD bit.

−g NAME, −−get=NAME, −s NAME[,VALUE], −−set=NAME[,VALUE]

Gets/sets non-SMART device settings. Note that the '−−set' option shares its short option '−s' with '−−smart'. Valid arguments are:

all − Gets all values. This is equivalent to
'−g aam −g apm −g lookahead −g security −g wcache −g rcache −g dsn'

aam[,N|off] − [ATA only] Gets/sets the Automatic Acoustic Management (AAM) feature (if supported). A value of 128 sets the most quiet (slowest) mode and 254 the fastest (loudest) mode, 'off' disables AAM. Devices may support intermediate levels. Values below 128 are defined as vendor specific (0) or retired (1 to 127). Note that the AAM feature was declared obsolete in ATA ACS-2 Revision 4a (Dec 2010).

apm[,N|off] − [ATA only] Gets/sets the Advanced Power Management (APM) feature on device (if supported). If a value between 1 and 254 is provided, it will attempt to enable APM and set the specified value, 'off' disables APM. Note the actual behavior depends on the drive, for example some drives disable APM if their value is set above 128. Values below 128 are supposed to allow drive spindown, values 128 and above adjust only head-parking frequency, although the actual behavior defined is also vendor-specific.

lookahead[,on|off] − [ATA only] Gets/sets the read look-ahead feature (if supported). Read look-ahead is usually enabled by default.

security − [ATA only] Gets the status of ATA Security feature (if supported). If ATA Security is enabled an ATA user password is set. The drive will be locked on next reset then.

security-freeze − [ATA only] Sets ATA Security feature to frozen mode. This prevents that the drive accepts any security commands until next reset. Note that the frozen mode may already be set by BIOS or OS.

standby,[N|off] − [ATA only] Sets the standby (spindown) timer and places the drive in the IDLE mode. A value of 0 or 'off' disables the standby timer. Values from 1 to 240 specify timeouts from 5 seconds to 20 minutes in 5 second increments. Values from 241 to 251 specify timeouts from 30 minutes to 330 minutes in 30 minute increments. Value 252 specifies 21 minutes. Value 253 specifies a vendor specific time between 8 and 12 hours. Value 255 specifies 21 minutes and 15 seconds. Some drives may use a vendor specific interpretation for the values. Note that there is no get option because ATA standards do not specify a method to read the standby timer.

[NEW EXPERIMENTAL SMARTCTL FEATURE] If '−s standby,now' is also specified, the drive is immediately placed in the STANDBY mode without temporarily placing it in the IDLE mode. Note that ATA standards do not specify a command to set the standby timer without affecting the power mode.

standby,now − [ATA only] Places the drive in the STANDBY mode. This usually spins down the drive. The setting of the standby timer is not affected unless '−s standby,[N|off]' is also specified.

wcache[,on|off] − [ATA] Gets/sets the volatile write cache feature (if supported). The write cache is usually enabled by default.

wcache[,on|off] − [SCSI] Gets/sets the 'Write Cache Enable' (WCE) bit (if supported). The write cache is usually enabled by default.

wcache-sct[,ata|on|off[,p]] − [ATA only] [NEW EXPERIMENTAL SMARTCTL FEATURE] Gets/sets the write cache feature through SCT Feature Control (if supported). The state of write cache in SCT Feature Control could be "Controlled by ATA", "Force Enabled", or "Force Disabled". SCT Feature control overwrites the setting by ATA Set Features command (wcache[,on|off] option). If SCT Feature Control sets write cache as "Force Enabled" or "Force Disabled", the setting of wcache[,on|off] is ignored by the drive. SCT Feature Control usually sets write cache as "Controlled by ATA" by default. If ',p' is specified, the setting is preserved across power cycles.

wcreorder[,on|off[,p]] − [ATA only] Gets/sets Write Cache Reordering. If it is disabled (off), disk write scheduling is executed on a first-in-first-out (FIFO) basis. If Write Cache Reordering is enabled (on), then disk write scheduling may be reordered by the drive. If write cache is disabled, the current Write Cache Reordering state is remembered but has no effect on non-cached writes, which are always written in the order received. The state of Write Cache Reordering has no effect on either NCQ or LCQ queued commands. [NEW EXPERIMENTAL SMARTCTL FEATURE] If ',p' is specified, the setting is preserved across power cycles.

rcache[,on|off] − [SCSI only] Gets/sets the 'Read Cache Disable' (RCE) bit. 'Off' value disables read cache (if supported). The read cache is usually enabled by default.

dsn[,on|off] − [ATA only] [NEW EXPERIMENTAL SMARTCTL FEATURE] Gets/sets the DSN feature (if supported). The dsn is usually disabled by default.

SMART READ AND DISPLAY DATA OPTIONS:

−H, −−health

Prints the health status of the device or pending TapeAlert messages.

If the device reports failing health status, this means **either** that the device has already failed, **or** that it is predicting its own failure within the next 24 hours. If this happens, use the '−a' option to get more information, and **get your data off the disk and to someplace safe as soon as you can.**

[ATA] Health status is obtained by checking the (boolean) result returned by the SMART RETURN STATUS command. The return value of this ATA command may be unknown due to limitations or bugs in some layer (e.g. RAID controller or USB bridge firmware) between disk and operating system. In this case, **smartctl** prints a warning and checks whether any Prefailure SMART Attribute value is less than or equal to its threshold (see '−A' below).

[SCSI] Health status is obtained by checking the Additional Sense Code (ASC) and Additional Sense Code Qualifier (ASCQ) from Informal Exceptions (IE) log page (if supported) and/or from SCSI sense data.

[SCSI tape drive or changer] TapeAlert status is obtained by reading the TapeAlert log page. Please note that the TapeAlert log page flags are cleared for the initiator when the page is read. This means that each alert condition is reported only once by **smartctl** for each initiator for each activation of the condition.

[NVMe] NVMe status is obtained by reading the "Critical Warning" byte from the SMART/Health

Information log.

−c, −−capabilities

[ATA] Prints only the generic SMART capabilities. These show what SMART features are implemented and how the device will respond to some of the different SMART commands. For example it shows if the device logs errors, if it supports offline surface scanning, and so on. If the device can carry out self-tests, this option also shows the estimated time required to run those tests.

[NVMe] Prints various NVMe device capabilities obtained from the Identify Controller and the Identify Namespace data structure.

−A, −−attributes

[ATA] Prints only the vendor specific SMART Attributes. The Attributes are numbered from 1 to 253 and have specific names and ID numbers. For example Attribute 12 is "power cycle count": how many times has the disk been powered up.

Each Attribute has a "Raw" value, printed under the heading "RAW_VALUE", and a "Normalized" value printed under the heading "VALUE". [Note:**smartctl** prints these v alues in base-10.] In the example just given, the "Raw Value" for Attribute 12 would be the actual number of times that the disk has been power-cycled, for example 365 if the disk has been turned on once per day for exactly one year. Each vendor uses their own algorithm to convert this "Raw" value to a "Normalized" value in the range from 1 to 254. Please keep in mind that **smartctl** only reports the different Attribute types, values, and thresholds as read from the device. It does**not** carry out the conversion between "Raw" and "Normalized" values: this is done by the disk's firmware.

The conversion from Raw value to a quantity with physical units is not specified by the SMART standard. In most cases, the values printed by **smartctl** are sensible. For example the temperature Attribute generally has its raw value equal to the temperature in Celsius. However in some cases vendors use unusual conventions. For example the Hitachi disk on my laptop reports its power-on hours in minutes, not hours. Some IBM disks track three temperatures rather than one, in their raw values. And so on.

Each Attribute also has a Threshold value (whose range is 0 to 255) which is printed under the heading "THRESH". If the Normalized value is **less than or equal to** the Threshold value, then the Attribute is said to have failed. If the Attribute is a pre-failure Attribute, then disk failure is imminent.

Each Attribute also has a "Worst" value shown under the heading "WORST". This is the smallest (closest to failure) value that the disk has recorded at any time during its lifetime when SMART was enabled. [Note however that some vendors firmware may actually **increase** the "Worst" value for some "rate-type" Attributes.]

The Attribute table printed out by **smartctl** also shows the "TYPE" of the Attribute. Attributes are one of two possible types: Pre-failure or Old age. Pre-failure Attributes are ones which, if less than or equal to their threshold values, indicate pending disk failure. Old age, or usage Attributes, are ones which indicate end-of-product life from old-age or normal aging and wearout, if the Attribute value is less than or equal to the threshold. **Please note**: the f act that an Attribute is of type 'Pre-fail' does **not** mean that your disk is about to fail! It only has this meaning if the Attribute's current Normalized value is less than or equal to the threshold value.

If the Attribute's current Normalized value is less than or equal to the threshold value, then the "WHEN_FAILED" column will display "FAILING_NOW". If not, but the worst recorded value is less than or equal to the threshold value, then this column will display "In_the_past". If the "WHEN_FAILED" column has no entry (indicated by a dash: '−') then this Attribute is OK now (not failing) and has also never failed in the past.

The table column labeled "UPDATED" shows if the SMART Attribute values are updated during both normal operation and off-line testing, or only during offline testing. The former are labeled "Always" and the latter are labeled "Offline".

So to summarize: the Raw Attribute values are the ones that might have a real physical interpretation, such as "Temperature Celsius", "Hours", or "Start-Stop Cycles". Each manufacturer converts these, using their detailed knowledge of the disk's operations and failure modes, to Normalized Attribute values in the range 1–254. The current and worst (lowest measured) of these Normalized Attribute values are stored on the disk, along with a Threshold value that the manufacturer has determined will indicate that the disk is going to fail, or that it has exceeded its design age or aging limit. **smartctl** does **not** calculate any of the Attribute values, thresholds, or types, it merely reports them from the SMART data on the device.

Note that starting with ATA/ATAPI-4, revision 4, the meaning of these Attribute fields has been made entirely vendor-specific. However most newer ATA/SATA disks seem to respect their meaning, so we have retained the option of printing the Attribute values.

Solid-state drives use different meanings for some of the attributes. In this case the attribute name printed by smartctl is incorrect unless the drive is already in the smartmontools drive database.

Note that the ATA command SMART READ DATA was declared obsolete in ATA ACS-4 Revision 10 (Nov 2015).

[SCSI] For SCSI devices the "attributes" are obtained from the temperature and start-stop cycle counter log pages. Certain vendor specific attributes are listed if recognised. The attributes are output in a relatively free format (compared with ATA disk attributes).

[NVMe] For NVMe devices the attributes are obtained from the SMART/Health Information log.

−f FORMAT, −−format=FORMAT

[ATA only] Selects the output format of the attributes:

old − Old smartctl format. This is the default unless the '−x' option is specified.

brief − New format which fits into 80 columns (except in some rare cases). This format also decodes four additional attribute flags. This is the default if the '−x' option is specified.

hex,id − Print all attribute IDs as hexadecimal numbers.

hex,val − Print all normalized values as hexadecimal numbers.

hex − Same as '−f hex,id −f hex,val'.

−l TYPE, −−log=TYPE

Prints various device logs. The valid arguments to this option are:

error − [ATA] prints the Summary SMART error log. SMART disks maintain a log of the most recent five non-trivial errors. For each of these errors, the disk power-on lifetime at which the error occurred is recorded, as is the device status (idle, standby, etc) at the time of the error. For some common types of errors, the Error Register (ER) and Status Register (SR) values are decoded and printed as text. The meanings of these are:

ABRT: Command**ABo**R **T**ed
AMNF: **A**ddress **M**ark **N**ot **F**ound
CCTO: **C**ommand **C**ompletion **T**imed **O**ut
EOM: **E**nd **O**f **M**edia
ICRC: **I**nterface **C**yclic **R**edundancy **C**ode (CRC) error
IDNF: **ID**entity **N**ot **F**ound
ILI: (packet command-set specific)
MC: **M**edia **C**hanged
MCR: **M**edia **C**hange **R**equest
NM: **N**o **M**edia
obs: **obs**olete
TK0NF: **T**rac**K 0** **N**ot **F**ound
UNC: **UNC**orrectable Error in Data
WP: Media is**W**rite **P**rotected

In addition, up to the last five commands that preceded the error are listed, along with a timestamp

measured from the start of the corresponding power cycle. This is displayed in the form Dd+HH:MM:SS.msec where D is the number of days, HH is hours, MM is minutes, SS is seconds and msec is milliseconds. [Note: this time stamp wraps after 2^32 milliseconds, or 49 days 17 hours 2 minutes and 47.296 seconds.] The key ATA disk registers are also recorded in the log. The final column of the error log is a text-string description of the ATA command defined by the Command Register (CR) and Feature Register (FR) values. Commands that are obsolete in the most current spec are listed like this: **READ LONG (w/ retry) [OBS-4]**, indicating that the command became obsolete with or in the ATA-4 specification. Similarly, the notation **[RET−*N*]** is used to indicate that a command was retired in the ATA-*N* specification. Some commands are not defined in any version of the ATA specification but are in common use nonetheless; these are marked **[NS]**, meaning non-standard.

The ATA Specification (ATA ACS-2 Revision 7, Section A.7.1) says: **"Error log data structures shall include, but are not limited to, Uncorrectable errors, ID Not Found errors for which the LBA requested was valid, servo errors, and write fault errors. Error log data structures shall not include errors attributed to the receipt of faulty commands."** The definitions of these terms are:
UNC (**UNC**orrectable): data is uncorrectable. This refers to data which has been read from the disk, but for which the Error Checking and Correction (ECC) codes are inconsistent. In effect, this means that the data can not be read.
IDNF (**ID N**ot **F**ound): user-accessible address could not be found. For READ LOG type commands, **IDNF** can also indicate that a device data log structure checksum was incorrect.

If the command that caused the error was a READ or WRITE command, then the Logical Block Address (LBA) at which the error occurred will be printed in base 10 and base 16. The LBA is a linear address, which counts 512-byte sectors on the disk, starting from zero. (Because of the limitations of the SMART error log, if the LBA is greater than 0xfffffff, then either no error log entry will be made, or the error log entry will have an incorrect LBA. This may happen for drives with a capacity greater than 128 GiB or 137 GB.) On Linux systems the smartmontools web page has instructions about how to convert the LBA address to the name of the disk file containing the erroneous disk sector.

Please note that some manufacturers **ignore** the ATA specifications, and make entries in the error log if the device receives a command which is not implemented or is not valid.

error − [SCSI] prints the error counter log pages for reads, write and verifies. The verify row is only output if it has an element other than zero.

error[,NUM] − [NVMe] prints the NVMe Error Information log. Only the 16 most recent log entries are printed by default. This number can be changed by the optional parameter NUM. The maximum number of log entries is vendor specific (in the range from 1 to 256 inclusive).

xerror[,NUM][,error] − [ATA only] prints the Extended Comprehensive SMART error log (General Purpose Log address 0x03). Unlike the Summary SMART error log (see '−l error' above), it provides sufficient space to log the contents of the 48-bit LBA register set introduced with ATA-6. It also supports logs with more than one sector. Each sector holds up to 4 log entries. The actual number of log sectors is vendor specific.

Only the 8 most recent error log entries are printed by default. This number can be changed by the optional parameter NUM.

If ',error' is appended and the Extended Comprehensive SMART error log is not supported, the Summary SMART self-test log is printed.

Please note that recent drives may report errors only in the Extended Comprehensive SMART error log. The Summary SMART error log may be reported as supported but is always empty then.

selftest − [ATA] prints the SMART self-test log. The disk maintains a self-test log showing the results of the self tests, which can be run using the '−t' option described below. For each of the

most recent twenty-one self-tests, the log shows the type of test (short or extended, off-line or captive) and the final status of the test. If the test did not complete successfully, then the percentage of the test remaining is shown. The time at which the test took place, measured in hours of disk lifetime, is also printed. [Note: this time stamp wraps after 2^16 hours, or 2730 days and 16 hours, or about 7.5 years.] If any errors were detected, the Logical Block Address (LBA) of the first error is printed in decimal notation.

selftest − [SCSI] the self-test log for a SCSI device has a slightly different format than for an ATA device. For each of the most recent twenty self-tests, it shows the type of test and the status (final or in progress) of the test. SCSI standards use the terms "foreground" and "background" (rather than ATA's corresponding "captive" and "off-line") and "short" and "long" (rather than ATA's corresponding "short" and "extended") to describe the type of the test. The printed segment number is only relevant when a test fails in the third or later test segment. It identifies the test that failed and consists of either the number of the segment that failed during the test, or the number of the test that failed and the number of the segment in which the test was run, using a vendor-specific method of putting both numbers into a single byte. The Logical Block Address (LBA) of the first error is printed in hexadecimal notation. If provided, the SCSI Sense Key (SK), Additional Sense Code (ASC) and Additional Sense Code Qualifier (ASCQ) are also printed. The self tests can be run using the '−t' option described below (using the ATA test terminology).

xselftest[,NUM][,selftest] − [ATA only] prints the Extended SMART self-test log (General Purpose Log address 0x07). Unlike the SMART self-test log (see '−l selftest' above), it supports 48-bit LBA and logs with more than one sector. Each sector holds up to 19 log entries. The actual number of log sectors is vendor specific.

Only the 25 most recent log entries are printed by default. This number can be changed by the optional parameter NUM.

If ',selftest' is appended and the Extended SMART self-test log is not supported, the old SMART self-test log is printed.

selective − [ATA only] Please see the '−t select' option below for a description of selective self-tests. The selective self-test log shows the start/end Logical Block Addresses (LBA) of each of the five test spans, and their current test status. If the span is being tested or the remainder of the disk is being read-scanned, the current 65536-sector block of LBAs being tested is also displayed. The selective self-test log also shows if a read-scan of the remainder of the disk will be carried out after the selective self-test has completed (see '−t afterselect' option) and the time delay before restarting this read-scan if it is interrupted (see '−t pending' option).

directory[,gs] − [ATA only] if the device supports the General Purpose Logging feature set (ATA-6 and above) then this prints the Log Directory (the log at address 0). The Log Directory shows what logs are available and their length in sectors (512 bytes). The contents of the logs at address 1 [Summary SMART error log] and at address 6 [SMART self-test log] may be printed using the previously-described *error* and *selftest* arguments to this option. If your version of smartctl supports 48-bit ATA commands, both the General Purpose Log (GPL) and SMART Log (SL) directories are printed in one combined table. The output can be restricted to the GPL directory or SL directory by '−l directory,q' or '−l directory,s' respectively.

background − [SCSI only] the background scan results log outputs information derived from Background Media Scans (BMS) done after power up and/or periodically (e.g. every 24 hours) on recent SCSI disks. If supported, the BMS status is output first, indicating whether a background scan is currently underway (and if so a progress percentage), the amount of time the disk has been powered up and the number of scans already completed. Then there is a header and a line for each background scan "event". These will typically be either recovered or unrecoverable errors. That latter group may need some attention. There is a description of the background scan mechanism in section 4.18 of SBC-3 revision 6 (see www.t10.org).

scttemp, scttempsts, scttemphist − [ATA only] prints the disk temperature information provided by the SMART Command Transport (SCT) commands. The option 'scttempsts' prints current

temperature and temperature ranges returned by the SCT Status command, 'scttemphist' prints temperature limits and the temperature history table returned by the SCT Data Table command, and 'scttemp' prints both. The temperature values are preserved across power cycles. The logging interval can be configured with the '−l scttempint,N[,p]' option, see below. The SCT commands were introduced in ATA8-ACS and were also supported by many ATA-7 disks.

scttempint,N[,p] − [ATA only] clears the SCT temperature history table and sets the time interval for temperature logging to N minutes. If ',p' is specified, the setting is preserved across power cycles. Otherwise, the setting is volatile and will be reverted to the last non-volatile setting by the next hard reset. The default interval is vendor specific, typical values are 1, 2, or 5 minutes.

scterc[,READTIME,WRITETIME] − [ATA only] prints values and descriptions of the SCT Error Recovery Control settings. These are equivalent to TLER (as used by Western Digital), CCTL (as used by Samsung and Hitachi/HGST) and ERC (as used by Seagate). READTIME and WRITE-TIME arguments (deciseconds) set the specified values. Values of 0 disable the feature, other values less than 65 are probably not supported. For RAID configurations, this is typically set to 70,70 deciseconds.

devstat[,PAGE] − [ATA only] prints values and descriptions of the ATA Device Statistics log pages (General Purpose Log address 0x04). If no PAGE number is specified, entries from all supported pages are printed. If PAGE 0 is specified, the list of supported pages is printed. Device Statistics was introduced in ACS-2 and is only supported by some recent devices.

defects[,NUM] − [ATA only] [NEW EXPERIMENTAL SMARTCTL FEATURE] prints LBA and hours values from the ATA Pending Defects log (General Purpose Log address 0x0c). Only the 31 entries from first log page are printed by default. This number can be changed by the optional parameter NUM. The size of the log and the order of the entries are vendor specific. The Pending Defects log was introduced in ACS-4 Revision 01 (Mar 2014).

sataphy[,reset] − [SATA only] prints values and descriptions of the SATA Phy Event Counters (General Purpose Log address 0x11). If '−l sataphy,reset' is specified, all counters are reset after reading the values. This also works for SATA devices with Packet interface like CD/DVD drives.

sasphy[,reset] − [SAS (SCSI) only] prints values and descriptions of the SAS (SSP) Protocol Specific log page (log page 0x18). If '−l sasphy,reset' is specified, all counters are reset after reading the values.

gplog,ADDR[,FIRST[−LAST|+SIZE]] − [ATA only] prints a hex dump of any log accessible via General Purpose Logging (GPL) feature. The log address ADDR is the hex address listed in the log directory (see '−l directory' above). The range of log sectors (pages) can be specified by decimal values FIRST−LAST or FIRST+SIZE. FIRST defaults to 0, SIZE defaults to 1. LAST can be set to 'max' to specify the last page of the log.

smartlog,ADDR[,FIRST[−LAST|+SIZE]] − [ATA only] prints a hex dump of any log accessible via SMART Read Log command. See '−l gplog,...' above for parameter syntax.

For example, all these commands:

```
smartctl −l gplog,0x80,10−15 /dev/sda
smartctl −l gplog,0x80,10+6 /dev/sda
smartctl −l smartlog,0x80,10−15 /dev/sda
```

print pages 10–15 of log 0x80 (first host vendor specific log).

The hex dump format is compatible with the 'xxd −r' command. This command:

```
smartctl −l gplog,0x11 /dev/sda | grep ^0 | xxd −r >log.bin
```

writes a binary representation of the one sector log 0x11 (SATA Phy Event Counters) to file log.bin.

nvmelog,PAGE,SIZE − [NVMe only] prints a hex dump of the first SIZE bytes from the NVMe log with identifier PAGE. PAGE is a hexadecimal number in the range from 0x1 to 0xff. SIZE is a hexadecimal number in the range from 0x4 to 0x4000 (16 KiB). **WARNING: Do not specify the identifier of an unknown log page. Reading a log page may have undesirable side effects.**

ssd − [ATA] prints the Solid State Device Statistics log page. This has the same effect as '−l devstat,7', see above.

ssd − [SCSI] prints the Solid State Media percentage used endurance indicator. A value of 0 indicates as new condition while 100 indicates the device is at the end of its lifetime as projected by the manufacturer. The value may reach 255.

−v ID,FORMAT[:BYTEORDER][,NAME], −−vendorattribute=ID,FORMAT...

[ATA only] Sets a vendor-specific raw value print FORMAT, an optional BYTEORDER and an optional NAME for Attribute ID. This option may be used multiple times.

The Attribute ID can be in the range 1 to 255. If 'N' is specified as ID, the settings for all Attributes are changed.

The optional BYTEORDER consists of 1 to 8 characters from the set '012345rvwz'. The characters '0' to '5' select the byte 0 to 5 from the 48-bit raw value, 'r' selects the reserved byte of the attribute data block, 'v' selects the normalized value, 'w' selects the worst value and 'z' inserts a zero byte. The default BYTEORDER is '543210' for all 48-bit formats, 'r543210' for the 54-bit formats, and '543210wv' for the 64-bit formats. For example, '−v 5,raw48:012345' prints the raw value of attribute 5 with big endian instead of little endian byte ordering.

The NAME is a string of letters, digits and underscore. Its length should not exceed 23 characters. The '−P showall' option reports an error if this is the case.

−v help − Prints (to STDOUT) a list of all valid arguments to this option, then exits.

Valid arguments for FORMAT are:

raw8 − Print the Raw value as six 8-bit unsigned base-10 integers. This may be useful for decoding the meaning of the Raw value.

raw16 − Print the Raw value as three 16-bit unsigned base-10 integers. This may be useful for decoding the meaning of the Raw value.

raw48 − Print the Raw value as a 48-bit unsigned base-10 integer. This is the default for most attributes.

hex48 − Print the Raw value as a 12 digit hexadecimal number. This may be useful for decoding the meaning of the Raw value.

raw56 − Print the Raw value as a 54-bit unsigned base-10 integer. This includes the reserved byte which follows the 48-bit raw value.

hex56 − Print the Raw value as a 14 digit hexadecimal number. This includes the reserved byte which follows the 48-bit raw value.

raw64 − Print the Raw value as a 64-bit unsigned base-10 integer. This includes two bytes from the normalized and worst attribute value. This raw format is used by some SSD devices with Indilinx controller.

hex64 − Print the Raw value as a 16 digit hexadecimal number. This includes two bytes from the normalized and worst attribute value. This raw format is used by some SSD devices with Indilinx controller.

min2hour − Raw Attribute is power-on time in minutes. Its raw value will be displayed in the form "Xh+Ym". Here X is hours, and Y is minutes in the range 0–59 inclusive. Y is always printed with two digits, for example "06" or "31" or "00".

sec2hour − Raw Attribute is power-on time in seconds. Its raw value will be displayed in the form "Xh+Ym+Zs". Here X is hours, Y is minutes in the range 0–59 inclusive, and Z is seconds in the range 0–59 inclusive. Y and Z are always printed with two digits, for example "06" or "31" or "00".

halfmin2hour − Raw Attribute is power-on time, measured in units of 30 seconds. This format is used by some Samsung disks. Its raw value will be displayed in the form "Xh+Ym". Here X is

hours, and Y is minutes in the range 0–59 inclusive. Y is always printed with two digits, for example "06" or "31" or "00".

msec24hour32 − Raw Attribute is power-on time measured in 32-bit hours and 24-bit milliseconds since last hour update. It will be displayed in the form "Xh+Ym+Z.Ms". Here X is hours, Y is minutes, Z is seconds and M is milliseconds.

tempminmax − Raw Attribute is the disk temperature in Celsius. Info about Min/Max temperature is printed if available. This is the default for Attributes 190 and 194. The recording interval (lifetime, last power cycle, last soft reset) of the min/max values is device specific.

temp10x − Raw Attribute is ten times the disk temperature in Celsius.

raw16(raw16) − Print the raw attribute as a 16-bit value and two optional 16-bit values if these words are nonzero. This is the default for Attributes 5 and 196.

raw16(avg16) − Raw attribute is spin-up time. It is printed as a 16-bit value and an optional "Average" 16-bit value if the word is nonzero. This is the default for Attribute 3.

raw24(raw8) − Print the raw attribute as a 24-bit value and three optional 8-bit values if these bytes are nonzero. This is the default for Attribute 9.

raw24/raw24 − Raw Attribute contains two 24-bit values. The first is the number of load cycles. The second is the number of unload cycles. The difference between these two values is the number of times that the drive was unexpectedly powered off (also called an emergency unload). As a rule of thumb, the mechanical stress created by one emergency unload is equivalent to that created by one hundred normal unloads.

raw24/raw32 − Raw attribute is an error rate which consists of a 24-bit error count and a 32-bit total count.

The following old arguments to '−v' are also still valid:

9,minutes − same as: *9,min2hour,Power_On_Minutes.*

9,seconds − same as: *9,sec2hour,Power_On_Seconds.*

9,halfminutes − same as: *9,halfmin2hour,Power_On_Half_Minutes.*

9,temp − same as: *9,tempminmax,Temperature_Celsius.*

192,emergencyretractcyclect − same as: *192,raw48,Emerg_Retract_Cycle_Ct*

193,loadunload − same as: *193,raw24/raw24.*

194,10xCelsius − same as: *194,temp10x,Temperature_Celsius_x10.*

194,unknown − same as: *194,raw48,Unknown_Attribute.*

197,increasing − same as: *197,raw48,Total_Pending_Sectors.* Also means that Attribute number 197 (Current Pending Sector Count) is not reset if uncorrectable sectors are reallocated (see **smartd.conf**(5) man page).

198,increasing − same as: *198,raw48,Total_Offl_Uncorrectabl.* Also means that Attribute number 198 (Offline Uncorrectable Sector Count) is not reset if uncorrectable sectors are reallocated (see **smartd.conf**(5) man page).

198,offlinescanuncsectorct − same as: *198,raw48,Offline_Scan_UNC_SectCt.*

200,writeerrorcount − same as: *200,raw48,Write_Error_Count.*

201,detectedtacount − same as: *201,raw48,Detected_TA_Count.*

220,temp − same as: *220,tempminmax,Temperature_Celsius.*

−F TYPE, −−firmwarebug=TYPE

[ATA only] Modifies the behavior of **smartctl** to compensate for some known and understood device firmware or driver bug. This option may be used multiple times. The valid arguments are:

none − Assume that the device firmware obeys the ATA specifications. This is the default, unless the device has presets for '−F' in the drive database. Using this option on the command line will override any preset values.

nologdir − Suppresses read attempts of SMART or GP Log Directory. Support for all standard logs is assumed without an actual check. Some Intel SSDs may freeze if log address 0 is read.

samsung − In some Samsung disks (example: model SV4012H Firmware Version: RM100-08) some of the two- and four-byte quantities in the SMART data structures are byte-swapped (relative to the ATA specification). Enabling this option tells **smartctl** to evaluate these quantities in byte-reversed order. Some signs that your disk needs this option are (1) no self-test log printed, even though you have run self-tests; (2) very large numbers of ATA errors reported in the ATA error log; (3) strange and impossible values for the ATA error log timestamps.

samsung2 − In some Samsung disks the number of ATA errors reported is byte swapped. Enabling this option tells **smartctl** to evaluate this quantity in byte-reversed order. An indication that your Samsung disk needs this option is that the self-test log is printed correctly, but there are a very large number of errors in the SMART error log. This is because the error count is byte swapped. Thus a disk with five errors (0x0005) will appear to have 20480 errors (0x5000).

samsung3 − Some Samsung disks (at least SP2514N with Firmware VF100-37) report a self-test still in progress with 0% remaining when the test was already completed. Enabling this option modifies the output of the self-test execution status (see options '−c' or '−a' above) accordingly.

xerrorlba − Fixes LBA byte ordering in Extended Comprehensive SMART error log. Some disks use little endian byte ordering instead of ATA register ordering to specify the LBA addresses in the log entries.

swapid − Fixes byte swapped ATA identify strings (device name, serial number, firmware version) returned by some buggy device drivers.

−P TYPE, −−presets=TYPE

[ATA only] Specifies whether **smartctl** should use any preset options that are available for this drive. By default, if the drive is recognized in the **smartmontools** database, then the presets are used.

The argument *show* will show any preset options for your drive and the argument *showall* will show all known drives in the **smartmontools** database, along with their preset options. If there are no presets for your drive and you think there should be (for example, a −v or −F option is needed to get **smartctl** to display correct values) then please contact the **smartmontools** developers so that this information can be added to the **smartmontools** database. Contact information is at the end of this man page.

The valid arguments to this option are:

use − if a drive is recognized, then use the stored presets for it. This is the default. Note that presets will NOT override additional Attribute interpretation ('−v N,something') command-line options or explicit '−F' command-line options..

ignore − do not use presets.

show − show if the drive is recognized in the database, and if so, its presets, then exit.

showall − list all recognized drives, and the presets that are set for them, then exit. This also checks the drive database regular expressions and settings for syntax errors.

The '−P showall' option takes up to two optional arguments to match a specific drive type and firmware version. The command:

```
smartctl -P showall
```

lists all entries, the command:

```
smartctl -P showall 'MODEL'
```

lists all entries matching MODEL, and the command:

```
smartctl -P showall 'MODEL' 'FIRMWARE'
```

lists all entries for this MODEL and a specific FIRMWARE version.

-B [+]FILE, --drivedb=[+]FILE

[ATA only] Read the drive database from FILE. The new database replaces the built in database by default. If '+' is specified, then the new entries prepend the built in entries.

Optional entries are read from the file **EXEDIR/drivedb-add.h**. if this option is not specified.

If **EXEDIR/drivedb.h** is present, the contents of this file is used instead of the built in table.

Run **EXEDIR/update-smart-drivedb.exe** to update this file from the smartmontools SVN repository.

The database files use the same C/C++ syntax that is used to initialize the built in database array. C/C++ style comments are allowed. Example:

```
/* Full entry: */
{
  "Model family",     // Info about model family/series.
  "MODEL1.*REGEX",    // Regular expression to match model of device.
  "VERSION.*REGEX",   // Regular expression to match firmware version(s).
  "Some warning",     // Warning message.
  "-v 9,minutes"      // String of preset -v and -F options.
},
/* Minimal entry: */
{
  "",                 // No model family/series info.
  "MODEL2.*REGEX",    // Regular expression to match model of device.
  "",                 // All firmware versions.
  "",                 // No warning.
  ""                  // No options preset.
},
/* USB ID entry: */
{
  "USB: Device; Bridge", // Info about USB device and bridge name.
  "0x1234:0xabcd",    // Regular expression to match vendor:product ID.
  "0x0101",           // Regular expression to match bcdDevice.
  "",                 // Not used.
  "-d sat"            // String with device type option.
},
/* ... */
```

SMART RUN/ABORT OFFLINE TEST AND self-test OPTIONS:

-t TEST, --test=TEST

Executes TEST immediately. The '-C' option can be used in conjunction with this option to run the short or long (and also for ATA devices, selective or conveyance) self-tests in captive mode (known as "foreground mode" for SCSI devices). Note that only one test type can be run at a time, so only one test type should be specified per command line. Note also that if a computer is shutdown or power cycled during a self-test, no harm should result. The self-test will either be aborted or will resume automatically.

All '-t TEST' commands can be given during normal system operation unless captive mode ('-C' option) is used. A running self-test can, however, degrade performance of the drive. Frequent I/O requests from the operating system increase the duration of a test. These impacts may vary from device to device.

If a test failure occurs then the device may discontinue the testing and report the result immediately.

[ATA] Note that the ATA command SMART EXECUTE OFF-LINE IMMEDIATE (the command to start a test) was declared obsolete in ATA ACS-4 Revision 10 (Nov 2015).

The valid arguments to this option are:

offline − [ATA] runs SMART Immediate Offline Test. This immediately starts the test described above. This command can be given during normal system operation. The effects of this test are visible only in that it updates the SMART Attribute values, and if errors are found they will appear in the SMART error log, visible with the '−l error' option.

If the '−c' option to **smartctl** shows that the device has the "Suspend Offline collection upon new command" capability then you can track the progress of the Immediate Offline test using the '−c' option to **smartctl**. If the '−c' option show that the device has the "Abort Offline collection upon new command" capability then most commands will abort the Immediate Offline Test, so you should not try to track the progress of the test with '−c', as it will abort the test.

offline − [SCSI] runs the default self test in foreground. No entry is placed in the self test log.

short − [ATA] runs SMART Short Self Test (usually under ten minutes). This command can be given during normal system operation (unless run in captive mode − see the '−C' option below). This is a test in a different category than the immediate or automatic offline tests. The "Self" tests check the electrical and mechanical performance as well as the read performance of the disk. Their results are reported in the Self Test Error Log, readable with the '−l selftest' option. Note that on some disks the progress of the self-test can be monitored by watching this log during the self-test; with other disks use the '−c' option to monitor progress.

short − [SCSI] runs the "Background short" self-test.

long − [ATA] runs SMART Extended Self Test (tens of minutes to several hours). This is a longer and more thorough version of the Short Self Test described above. Note that this command can be given during normal system operation (unless run in captive mode − see the '−C' option below).

long − [SCSI] runs the "Background long" self-test.

conveyance − [ATA only] runs a SMART Conveyance Self Test (minutes). This self-test routine is intended to identify damage incurred during transporting of the device. This self-test routine should take on the order of minutes to complete. Note that this command can be given during normal system operation (unless run in captive mode − see the '−C' option below).

select,N−M, *select,N+SIZE* − [ATA only] runs a SMART Selective Self Test, to test a **range** of disk Logical Block Addresses (LBAs), rather than the entire disk. Each range of LBAs that is checked is called a "span" and is specified by a starting LBA (N) and an ending LBA (M) with N less than or equal to M. The range can also be specified as N+SIZE. A span at the end of a disk can be specified by N−**max**.

For example the commands:

```
smartctl −t select,10−20 /dev/sda
smartctl −t select,10+11 /dev/sda
```

both runs a self test on one span consisting of LBAs ten to twenty (inclusive). The command:

```
smartctl −t select,100000000−max /dev/sda
```

run a self test from LBA 100000000 up to the end of the disk. The '−t' option can be given up to five times, to test up to five spans. For example the command:

```
smartctl −t select,0−100 −t select,1000−2000 /dev/sda
```

runs a self test on two spans. The first span consists of 101 LBAs and the second span consists of 1001 LBAs. Note that the spans can overlap partially or completely, for example:

```
smartctl −t select,0−10 −t select,5−15 −t select,10−20 /dev/sda
```

The results of the selective self-test can be obtained (both during and after the test) by printing the SMART self-test log, using the '−l selftest' option to smartctl.

Selective self tests are particularly useful as disk capacities increase: an extended self test (smartctl −t long) can take several hours. Selective self-tests are helpful if (based on SYSLOG error messages, previous failed self-tests, or SMART error log entries) you suspect that a disk is having

problems at a particular range of Logical Block Addresses (LBAs).

Selective self-tests can be run during normal system operation (unless done in captive mode − see the '−C' option below).

The following variants of the selective self-test command use spans based on the ranges from past tests already stored on the disk:

select,redo[+SIZE] − [ATA only] redo the last SMART Selective Self Test using the same LBA range. The starting LBA is identical to the LBA used by last test, same for ending LBA unless a new span size is specified by optional +SIZE argument.

For example the commands:

```
smartctl −t select,10−20 /dev/sda
smartctl −t select,redo /dev/sda
smartctl −t select,redo+20 /dev/sda
```

have the same effect as:

```
smartctl −t select,10−20 /dev/sda
smartctl −t select,10−20 /dev/sda
smartctl −t select,10−29 /dev/sda
```

select,next[+SIZE] − [ATA only] runs a SMART Selective Self Test on the LBA range which follows the range of the last test. The starting LBA is set to (ending LBA +1) of the last test. A new span size may be specified by the optional +SIZE argument.

For example the commands:

```
smartctl −t select,0−999 /dev/sda
smartctl −t select,next /dev/sda
smartctl −t select,next+2000 /dev/sda
```

have the same effect as:

```
smartctl −t select,0−999 /dev/sda
smartctl −t select,1000−1999 /dev/sda
smartctl −t select,2000−3999 /dev/sda
```

If the last test ended at the last LBA of the disk, the new range starts at LBA 0. The span size of the last span of a disk is adjusted such that the total number of spans to check the full disk will not be changed by future uses of '−t select,next'.

select,cont[+SIZE] − [ATA only] performs a 'redo' (above) if the self test status reports that the last test was aborted by the host. Otherwise it run the 'next' (above) test.

afterselect,on − [ATA only] perform an offline read scan after a Selective self-test has completed. This option must be used together with one or more of the *select,N−M* options above. If the LBAs that have been specified in the Selective self-test pass the test with no errors found, then read scan the **remainder** of the disk. If the device is powered-cycled while this read scan is in progress, the read scan will be automatically resumed after a time specified by the pending timer (see below). The value of this option is preserved between selective self-tests.

afterselect,off − [ATA only] do not read scan the remainder of the disk after a Selective self-test has completed. This option must be use together with one or more of the *select,N−M* options above. The value of this option is preserved between selective self-tests.

pending,N − [ATA only] set the pending offline read scan timer to N minutes. Here N is an integer in the range from 0 to 65535 inclusive. If the device is powered off during a read scan after a Selective self-test, then resume the test automatically N minutes after power-up. This option must be use together with one or more of the *select,N−M* options above. The value of this option is preserved between selective self-tests.

vendor,N − [ATA only] issues the ATA command SMART EXECUTE OFF-LINE IMMEDIATE with subcommand N in LBA LOW register. The subcommand is specified as a hex value in the range 0x00 to 0xff. Subcommands 0x40–0x7e and 0x90–0xff are reserved for vendor specific use, see table 61 of T13/1699-D Revision 6a (ATA8-ACS). Note that the subcommands 0x00–0x04,

0x7f, 0x81–0x84 are supported by other smartctl options (e.g. 0x01: '−t short', 0x7f: '−X', 0x82: '−C −t long').

WARNING: Only run subcommands documented by the vendor of the device.

Example for some Intel SSDs only: The subcommand 0x40 ('−t vendor,0x40') clears the timed workload related SMART attributes (226, 227, 228). Note that the raw values of these attributes are held at 65535 (0xffff) until the workload timer reaches 60 minutes.

force − start new self-test even if another test is already running. By default a running self-test will not be interrupted to begin another test.

−C, −−captive

[ATA] Runs self-tests in captive mode. This has no effect with '−t offline' or if the '−t' option is not used.

WARNING: Tests run in captive mode may busy out the drive for the length of the test. Only run captive tests on drives without any mounted partitions!

[SCSI] Runs the self-test in "Foreground" mode.

−X, −−abort

Aborts non-captive SMART Self Tests. Note that this command will abort the Offline Immediate Test routine only if your disk has the "Abort Offline collection upon new command" capability.

ATA, SCSI command sets and SAT

In the past there has been a clear distinction between storage devices that used the ATA and SCSI command sets. This distinction was often reflected in their device naming and hardware. Now various SCSI transports (e.g. SAS, FC and iSCSI) can interconnect to both SCSI disks (e.g. FC and SAS) and ATA disks (especially SATA). USB and IEEE 1394 storage devices use the SCSI command set externally but almost always contain ATA or SATA disks (or flash). The storage subsystems in some operating systems have started to remove the distinction between ATA and SCSI in their device naming policies.

99% of operations that an OS performs on a disk involve the SCSI INQUIRY, READ CAPACITY, READ and WRITE commands, or their ATA equivalents. Since the SCSI commands are slightly more general than their ATA equivalents, many OSes are generating SCSI commands (mainly READ and WRITE) and letting a lower level translate them to their ATA equivalents as the need arises. An important note here is that "lower level" may be in external equipment and hence outside the control of an OS.

SCSI to ATA Translation (SAT) is a standard (ANSI INCITS 431-2007) that specifies how this translation is done. For the other 1% of operations that an OS performs on a disk, SAT provides two options. First is an optional ATA PASS-THROUGH SCSI command (there are two variants). The second is a translation from the closest SCSI command. Most current interest is in the "pass-through" option.

The relevance to smartmontools (and hence smartctl) is that its interactions with disks fall solidly into the "1%" category. So even if the OS can happily treat (and name) a disk as "SCSI", smartmontools needs to detect the native command set and act accordingly. As more storage manufacturers (including external SATA drives) comply with SAT, smartmontools is able to automatically distinguish the native command set of the device. In some cases the '−d sat' option is needed on the command line.

There are also virtual disks which typically have no useful information to convey to smartmontools, but could conceivably in the future. An example of a virtual disk is the OS's view of a RAID 1 box. There are most likely two SATA disks inside a RAID 1 box. Addressing those SATA disks from a distant OS is a challenge for smartmontools. Another approach is running a tool like smartmontools inside the RAID 1 box (e.g. a Network Attached Storage (NAS) box) and fetching the logs via a browser.

EXAMPLES

smartctl −a /dev/sda

Print a large amount of SMART information for drive /dev/sda.

smartctl −s off /dev/sdd

Disable SMART monitoring and data log collection on drive /dev/sdd.

smartctl −−smart=on −−offlineauto=on −−saveauto=on /dev/sda
Enable SMART on drive /dev/sda, enable automatic offline testing every four hours, and enable autosaving of SMART Attributes. This is a good start-up line for your system's init files. You can issue this command on a running system.

smartctl −t long /dev/sdc
Begin an extended self-test of drive /dev/sdc. You can issue this command on a running system. The results can be seen in the self-test log visible with the '−l selftest' option after it has completed.

smartctl −s on −t offline /dev/sda
Enable SMART on the disk, and begin an immediate offline test of drive /dev/sda. You can issue this command on a running system. The results are only used to update the SMART Attributes, visible with the '−A' option. If any device errors occur, they are logged to the SMART error log, which can be seen with the '−l error' option.

smartctl −A −v 9,minutes /dev/sda
Shows the vendor Attributes, when the disk stores its power-on time internally in minutes rather than hours.

smartctl −q errorsonly −H −l selftest /dev/sda
Produces output only if the device returns failing SMART status, or if some of the logged self-tests ended with errors.

smartctl −q silent −a /dev/sda
Examine all SMART data for device /dev/sda, but produce no printed output. You must use the exit status (the **$?** shell variable) to learn if any Attributes are out of bound, if the SMART status is failing, if there are errors recorded in the self-test log, or if there are errors recorded in the disk error log.

smartctl −a −d 3ware,0 /dev/twl0
Examine all SMART data for the first SATA (not SAS) disk connected to a 3ware RAID 9750 controller card.

smartctl −t long −d areca,4 /dev/sg2
Start a long self-test on the fourth SATA disk connected to an Areca RAID controller addressed by /dev/sg2.

smartctl −a −d hpt,1/3 /dev/sda (under Linux)
smartctl −a −d hpt,1/3 /dev/hptrr (under FreeBSD)
Examine all SMART data for the (S)ATA disk directly connected to the third channel of the first HighPoint RocketRAID controller card.

smartctl −t short −d hpt,1/1/2 /dev/sda (under Linux)
smartctl −t short −d hpt,1/1/2 /dev/hptrr (under FreeBSD)
Start a short self-test on the (S)ATA disk connected to second pmport on the first channel of the first HighPoint RocketRAID controller card.

smartctl −t select,10−100 −t select,30−300 −t afterselect,on −t pending,45 /dev/sda
Run a selective self-test on LBAs 10 to 100 and 30 to 300. After the these LBAs have been tested, read-scan the remainder of the disk. If the disk is power-cycled during the read-scan, resume the scan 45 minutes after power to the device is restored.

smartctl −a −d cciss,0 /dev/cciss/c0d0
Examine all SMART data for the first SCSI disk connected to a cciss RAID controller card.

EXIT STATUS

The exit statuses of **smartctl** are defined by a bitmask. If all is well with the disk, the exit status (return value) of **smartctl** is 0 (all bits turned off). If a problem occurs, or an error, potential error, or fault is detected, then a non-zero status is returned. In this case, the eight different bits in the exit status have the following meanings for ATA disks; some of these values may also be returned for SCSI disks.

Bit 0: Command line did not parse.

Bit 1: Device open failed, device did not return an IDENTIFY DEVICE structure, or device is in a low-power mode (see '−n' option above).

Bit 2: Some SMART or other ATA command to the disk failed, or there was a checksum error in a SMART data structure (see '−b' option above).

Bit 3: SMART status check returned "DISK FAILING".

Bit 4: We found prefail Attributes <= threshold.

Bit 5: SMART status check returned "DISK OK" but we found that some (usage or prefail) Attributes have been <= threshold at some time in the past.

Bit 6: The device error log contains records of errors.

Bit 7: The device self-test log contains records of errors. [ATA only] Failed self-tests outdated by a newer successful extended self-test are ignored.

To test within the shell for whether or not the different bits are turned on or off, you can use the following type of construction (which should work with any POSIX compatible shell):
smartstat=$(($? & 8))
This looks at only at bit 3 of the exit status **$?** (since 8=2^3). The shell variable $smartstat will be nonzero if SMART status check returned "disk failing" and zero otherwise.

This shell script prints all status bits:

```
val=$?; mask=1
for i in 0 1 2 3 4 5 6 7; do
  echo "Bit $i: $(((val & mask) && 1))"
  mask=$((mask << 1))
done
```

AUTHORS

Bruce Allen (project initiator),
Christian Franke (project manager, Windows port and all sort of things),
Douglas Gilbert (SCSI subsystem),
Volker Kuhlmann (moderator of support and database mailing list),
Gabriele Pohl (wiki & development team support),
Alex Samorukov (FreeBSD port and more, new Trac wiki).

Many other individuals have made contributions and corrections, see AUTHORS, ChangeLog and repository files.

The first smartmontools code was derived from the smartsuite package, written by Michael Cornwell and Andre Hedrick.

REPORTING BUGS

To submit a bug report, create a ticket in smartmontools wiki:
<**https://www.smartmontools.org/**>.
Alternatively send the info to the smartmontools support mailing list:
<**https://listi.jpberlin.de/mailman/listinfo/smartmontools-support>.**

SEE ALSO

smartd(8).
update-smart-drivedb(8).

REFERENCES

Please see the following web site for more info: <**https://www.smartmontools.org/**>

An introductory article about smartmontools is *Monitoring Hard Disks with SMART*, by Bruce Allen, Linux Journal, January 2004, pages 74–77. See <**https://www.linuxjournal.com/article/6983**>.

If you would like to understand better how SMART works, and what it does, a good place to start is with

Sections 4.8 and 6.54 of the first volume of the 'AT Attachment with Packet Interface-7' (ATA/ATAPI-7) specification Revision 4b. This documents the SMART functionality which the **smartmontools** utilities provide access to.

The functioning of SMART was originally defined by the SFF-8035i revision 2 and the SFF-8055i revision 1.4 specifications. These are publications of the Small Form Factors (SFF) Committee.

Links to these and other documents may be found on the Links page of the **smartmontools** Wiki at <**https://www.smartmontools.org/wiki/Links**>.

PACKAGE VERSION

smartmontools-7.1 2019-03-31 r4903
$Id: smartctl.8.in 4882 2018-12-29 21:26:45Z chrfranke $