NAME

smartd.conf − SMART Disk Monitoring Daemon Configuration File

FULL PATH

/etc/smartd.conf

PACKAGE VERSION

smartmontools-5.43 2012-06-30 r3573

DESCRIPTION

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

/etc/smartd.conf is the configuration file for the **smartd** daemon, which monitors the Self-Monitoring, Analysis and Reporting Technology (SMART) system built into many ATA-3 and later ATA, IDE and SCSI-3 hard drives.

If the configuration file **/etc/smartd.conf** is present, **smartd** reads it at startup, before **fork**(2)ing into the background. If **smartd** subsequently receives a **HUP** signal, it will then re-read the configuration file. If **smartd** is running in debug mode, then an **INT** signal will also make it re-read the configuration file. This signal can be generated by typing **<CONTROL-C>** in the terminal window where **smartd** is running.

CONFIGURATION FILE /etc/smartd.conf

In the absence of a configuration file **smartd** will try to open all available devices. Under linux **smartd** will try to open the 20 ATA devices **/dev/hd[a-t]** and the 26 SCSI devices **/dev/sd[a-z].**

This can be annoying if you have an ATA or SCSI device that hangs or misbehaves when receiving SMART commands. Even if this causes no problems, you may be annoyed by the string of error log messages about block-major devices that can´t be found, and SCSI devices that can´t be opened.

One can avoid this problem, and gain more control over the types of events monitored by **smartd**, by using the configuration file **/etc/smartd.conf.** This file contains a list of devices to monitor, with one device per line. An example file is included with the **smartmontools** distribution. You will find this sample configuration file in **/usr/share/doc/smartmontools/**. For security, the configuration file should not be writable by anyone but root. The syntax of the file is as follows:

- There should be one device listed per line, although you may have lines that are entirely comments or white space.
- Any text following a hash sign ´#´ and up to the end of the line is taken to be a comment, and ignored.
- Lines may be continued by using a backslash ´\´ as the last non-whitespace or non-comment item on a line.
- Note: a line whose first character is a hash sign ´#´ is treated as a white-space blank line, **not** as a non-existent line, and will **end** a continuation line.

Here is an example configuration file. It´s for illustrative purposes only; please don´t copy it onto your system without reading to the end of the **DIRECTIVES** Section below!

```
################################################
# This is an example smartd startup config file
# /etc/smartd.conf for monitoring three
# ATA disks, three SCSI disks, six ATA disks
# behind two 3ware controllers, three SATA disks
# directly connected to the HighPoint Rocket-
# RAID controller, two SATA disks connected to
# the HighPoint RocketRAID controller via a pmport
# device, four SATA disks connected to an Areca
# RAID controller, and one SATA disk.
#
# First ATA disk on two different interfaces. On
# the second disk, start a long self-test every
# Sunday between 3 and 4 am.
#
  /dev/hda -a -m admin@example.com,root@localhost
  /dev/hdc -a -I 194 -I 5 -i 12 -s L/../../7/03
#
# SCSI disks. Send a TEST warning email to admin on
# startup.
#
  /dev/sda
  /dev/sdb -m admin@example.com -M test
#
# Strange device. It´s SCSI. Start a scheduled
# long self test between 5 and 6 am Monday/Thursday
  /dev/weird -d scsi -s L/../../(1|4)/05
#
# An ATA disk may appear as a SCSI device to the
# OS. If a SCSI to ATA Translation (SAT) layer
# is between the OS and the device then this can be
# flagged with the ’-d sat’ option. This situation
# may become common with SATA disks in SAS and FC
# environments.
  /dev/sda -a -d sat
#
# Three disks connected to a MegaRAID controller
# Start short self-tests daily between 1-2, 2-3, and
# 3-4 am.
  /dev/sda -d megaraid,0 -a -s S/../.././01
  /dev/sda -d megaraid,1 -a -s S/../.././02
  /dev/sda -d megaraid,2 -a -s S/../.././03
#
# Four ATA disks on a 3ware 6/7/8000 controller.
# Start short self-tests daily between midnight and 1am,
# 1-2, 2-3, and 3-4 am. Starting with the Linux 2.6
# kernel series, /dev/sdX is deprecated in favor of
# /dev/tweN. For example replace /dev/sdc by /dev/twe0
# and /dev/sdd by /dev/twe1.
  /dev/sdc -d 3ware,0 -a -s S/../.././00
  /dev/sdc -d 3ware,1 -a -s S/../.././01
  /dev/sdd -d 3ware,2 -a -s S/../.././02
  /dev/sdd -d 3ware,3 -a -s S/../.././03
#
# Two ATA disks on a 3ware 9000 controller.
# Start long self-tests Sundays between midnight and
# 1am and 2-3 am
  /dev/twa0 -d 3ware,0 -a -s L/../../7/00
  /dev/twa0 -d 3ware,1 -a -s L/../../7/02
#
```

```
# Two SATA (not SAS) disks on a 3ware 9750 controller.
# Start long self-tests Sundays between midnight and
# 1am and 2-3 am
  /dev/twl0 -d 3ware,0 -a -s L/../../7/00
  /dev/twl0 -d 3ware,1 -a -s L/../../7/02
#
# Three SATA disks on a HighPoint RocketRAID controller.
# Start short self-tests daily between 1-2, 2-3, and
# 3-4 am.
# under Linux
  /dev/sde -d hpt,1/1 -a -s S/../.././01
  /dev/sde -d hpt,1/2 -a -s S/../.././02
  /dev/sde -d hpt,1/3 -a -s S/../.././03
#
# Two SATA disks connected to a HighPoint RocketRAID
# via a pmport device. Start long self-tests Sundays
# between midnight and 1am and 2-3 am.
# under Linux
  /dev/sde -d hpt,1/4/1 -a -s L/../../7/00
  /dev/sde -d hpt,1/4/2 -a -s L/../../7/02
# Three SATA disks connected to an Areca
# RAID controller. Start long self-tests Sundays
# between midnight and 3 am.
  /dev/sg2 -d areca,1 -a -s L/../../7/00
  /dev/sg2 -d areca,2 -a -s L/../../7/01
  /dev/sg2 -d areca,3 -a -s L/../../7/02
#
# The following line enables monitoring of the
# ATA Error Log and the Self-Test Error Log.
# It also tracks changes in both Prefailure
# and Usage Attributes, apart from Attributes
# 9, 194, and 231, and shows continued lines:
#
  /dev/hdd -l error \
           -l selftest \
           -t \      # Attributes not tracked:
           -I 194 \  # temperature
           -I 231 \  # also temperature
           -I 9      # power-on hours
#
################################################
```

CONFIGURATION FILE DIRECTIVES

If a non-comment entry in the configuration file is the text string **DEVICESCAN** in capital letters, then **smartd** will ignore any remaining lines in the configuration file, and will scan for devices. **DEVICESCAN** may optionally be followed by Directives that will apply to all devices that are found in the scan. Please see below for additional details.

[NEW EXPERIMENTAL SMARTD FEATURE] If an entry in the configuration file starts with **DEFAULT** instead of a device name, then all directives in this entry are set as defaults for the next device entries.

This configuration:

```
  DEFAULT -a -R5! -W 2,40,45 -I 194 -s L/../../7/00 -m admin@example.com
```

```
 /dev/sda
 /dev/sdb
 /dev/sdc
 DEFAULT -H -m admin@example.com
 /dev/sdd
 /dev/sde -d removable
```

has the same effect as:

```
 /dev/sda -a -R5! -W 2,40,45 -I 194 -s L/../../7/00 -m admin@example.com
 /dev/sdb -a -R5! -W 2,40,45 -I 194 -s L/../../7/00 -m admin@example.com
 /dev/sdc -a -R5! -W 2,40,45 -I 194 -s L/../../7/00 -m admin@example.com
 /dev/sdd -H -m admin@example.com
 /dev/sde -d removable -H -m admin@example.com
```

The following are the Directives that may appear following the device name or **DEVICESCAN** or **DEFAULT** on any line of the **/etc/smartd.conf** configuration file. Note that **these are NOT command-line options for smartd**. The Directives below may appear in any order, following the device name.

For an ATA device, if no Directives appear, then the device will be monitored as if the ´−a´ Directive (monitor all SMART properties) had been given.

If a SCSI disk is listed, it will be monitored at the maximum implemented level: roughly equivalent to using the ´−H −l selftest´ options for an ATA disk. So with the exception of ´−d´, ´−m´, ´−l selftest´, ´−s´, and ´−M´, the Directives below are ignored for SCSI disks. For SCSI disks, the ´−m´ Directive sends a warning email if the SMART status indicates a disk failure or problem, if the SCSI inquiry about disk status fails, or if new errors appear in the self-test log.

If a 3ware controller is used then the corresponding SCSI (/dev/sd?) or character device (/dev/twe?, /dev/twa? or /dev/twl?) must be listed, along with the ´−d 3ware,N´ Directive (see below). The individual ATA disks hosted by the 3ware controller appear to **smartd** as normal ATA devices. Hence all the ATA directives can be used for these disks (but see note below).

If an Areca controller is used then the corresponding device (SCSI /dev/sg? on Linux or /dev/arcmsr0 on FreeBSD) must be listed, along with the ´−d areca,N´ Directive (see below). The individual SATA disks hosted by the Areca controller appear to **smartd** as normal ATA devices. Hence all the ATA directives can be used for these disks. Areca firmware version 1.46 or later which supports smartmontools must be used; Please see the **smartctl**(8) man page for further details.

−d TYPE

Specifies the type of the device. The valid arguments to this directive 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.

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

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

sat[,auto][,N] − the device type is SCSI to ATA Translation (SAT). This is for ATA disks that have a SCSI to ATA Translation (SAT) 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´.

[NEW EXPERIMENTAL SMARTD FEATURE] 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 - 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.

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

marvell − [Linux only] interact with SATA disks behind Marvell chip-set controllers (using the Marvell rather than libata driver).

megaraid,N − [Linux only] the device consists of one or more SCSI/SAS disks connected to a MegaRAID controller. The non-negative integer N (in the range of 0 to 127 inclusive) denotes which disk on the controller is monitored. This interface will also work for Dell PERC controllers. In log files and email messages this disk will be identified as megaraid_disk_XXX with XXX in the range from 000 to 127 inclusive. Please see the **smartctl**(8) man page for further details.

3ware,N − [FreeBSD and Linux only] the device consists of one or more ATA disks connected to a 3ware RAID controller. The non-negative integer N (in the range from 0 to 127 inclusive) denotes which disk on the controller is monitored. In log files and email messages this disk will be identified as 3ware_disk_XXX with XXX in the range from 000 to 127 inclusive.

Note that while you may use **any** of the 3ware SCSI logical devices /dev/tw* to address **any** of the physical disks (3ware ports), error and log messages will make the most sense if you always list the 3ware SCSI logical device corresponding to the particular physical disks. Please see the **smartctl**(8) man page for further details.

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. In log files and email messages this disk will be identifed as areca_disk_XX with XX in the range from 01 to 24 inclusive. Please see the **smartctl**(8) man page for further details.

areca,N/E − [FreeBSD, Linux, Windows and Cygwin only] [NEW EXPERIMENTAL SMARTD FEATURE] the device consists of one or more SATA 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 upcoming Areca SAS controller firmware version 1.51 or a recent beta version.

cciss,N − [FreeBSD and Linux only] the device consists of one or more SCSI/SAS or SATA disks connected to a cciss RAID controller. The non-negative integer N (in the range from 0 to 15 inclusive) denotes which disk on the controller is monitored. In log files and email messages this disk will be identified as cciss_disk_XX with XX in the range from 00 to 15 inclusive. Please see the **smartctl**(8) man page for further details.

hpt,L/M/N − [FreeBSD and Linux only] the device consists of one or more ATA disks connected to a HighPoint RocketRAID controller. The integer L is the controller id, the integer M is the channel number, and the integer N is the PMPort number if it is available. The allowed values of L are from 1 to 4 inclusive, M are from 1 to 16 inclusive and N from 1 to 4 if PMPort available. And also these values are limited by the model of the HighPoint RocketRAID controller. In log files and email messages this disk will be identified as hpt_X/X/X and X/X/X is the same as L/M/N, note if no N indicated, N set to the default value 1. Please see the **smartctl**(8) man page for further details.

removable − the device or its media is removable. This indicates to **smartd** that it should continue (instead of exiting, which is the default behavior) if the device does not appear to be present when **smartd** is started. This Directive may be used in conjunction with the other ´−d´ Directives.

−n POWERMODE[,N][,q]

[ATA only] This ´nocheck´ Directive is used to prevent a disk from being spun-up when it is periodically polled by **smartd**.

ATA disks have five different power states. In order of increasing power consumption they are: ´OFF´, ´SLEEP´, ´STANDBY´, ´IDLE´, and ´ACTIVE´. Typically in the OFF, SLEEP, and STANDBY modes the disk´s platters are not spinning. But usually, in response to SMART commands issued by **smartd**, the disk platters are spun up. So if this option is not used, then a disk which is in a low−power mode may be spun up and put into a higher−power mode when it is periodically polled by **smartd**.

Note that if the disk is in SLEEP mode when **smartd** is started, then it won’t respond to **smartd** commands, and so the disk won’t be registered as a device for **smartd** to monitor. If a disk is in any other low−power mode, then the commands issued by **smartd** to register the disk will probably cause it to spin−up.

The ´**−n**´ (nocheck) Directive specifies if **smartd**´s periodic checks should still be carried out when the device is in a low−power mode. It may be used to prevent a disk from being spun−up by periodic **smartd** polling. The allowed values of POWERMODE are:

never − **smartd** will poll (check) the device regardless of its power mode. This may cause a disk which is spun−down to be spun−up when **smartd** checks it. This is the default behavior if the ’−n’ Directive is not given.

sleep − check the device unless it is in SLEEP mode.

standby − 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 laptop disk from spinning up each time that **smartd** polls, this is probably what you want.

idle − 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.

Maximum number of skipped checks (in a row) can be specified by appending positive number ´,N´ to POWERMODE (like ´−n standby,15´). After N checks are skipped in a row, powermode is ignored and the check is performed anyway.

When a periodic test is skipped, **smartd** normally writes an informal log message. The message can be suppressed by appending the option ´,q´ to POWERMODE (like ´−n standby,q´). This prevents a laptop disk from spinning up due to this message.

Both ´,N´ and ´,q´ can be specified together.

−T TYPE

Specifies how tolerant **smartd** should be of SMART command failures. The valid arguments to this Directive are:

normal − do not try to monitor the disk if a mandatory SMART command fails, but continue if an optional SMART command fails. This is the default.

permissive − try to monitor the disk even if it appears to lack SMART capabilities. This may be required for some old disks (prior to ATA−3 revision 4) that implemented SMART before the SMART standards were incorporated into the ATA/ATAPI Specifications. This may also be needed for some Maxtor disks which fail to comply with the ATA Specifications and don't properly indicate support for error− or self−test logging.

[Please see the **smartctl −T** command-line option.]

−o VALUE

[ATA only] Enables or disables SMART Automatic Offline Testing when **smartd** starts up and has no further effect. The valid arguments to this Directive are *on* and *off*.

The delay between tests is vendor-specific, but is typically four hours.

Note that SMART Automatic Offline Testing is **not** part of the ATA Specification. Please see the **smartctl −o** command-line option documentation for further information about this feature.

−S VALUE

Enables or disables Attribute Autosave when **smartd** starts up and has no further effect. The valid arguments to this Directive are *on* and *off*. Also affects SCSI devices. [Please see the **smartctl −S** command-line option.]

−H [ATA only] Check the SMART health status of the disk. If any Prefailure Attributes are less than or equal to their threshold values, then disk failure is predicted in less than 24 hours, and a message at loglevel **´LOG_CRIT´** will be logged to syslog. [Please see the **smartctl −H** command-line option.]

−l TYPE

Reports increases in the number of errors in one of three SMART logs. The valid arguments to this Directive are:

error − [ATA only] report if the number of ATA errors reported in the Summary SMART error log has increased since the last check.

xerror − [ATA only] report if the number of ATA errors reported in the Extended Comprehensive SMART error log has increased since the last check.

If both ´−l error´ and ´−l xerror´ are specified, smartd checks the maximum of both values.

[Please see the **smartctl −l xerror** command-line option.]

selftest − report if the number of failed tests reported in the SMART Self-Test Log has increased since the last check, or if the timestamp associated with the most recent failed test has increased. Note that such errors will **only** be logged if you run self-tests on the disk (and it fails a test!). Self-Tests can be run automatically by **smartd**: please see the **´−s´** Directive below. Self-Tests can also be run manually by using the **´−t short´** and **´−t long´** options of **smartctl** and the results of the testing can be observed using the **smartctl ´−l selftest´** command-line option. [Please see the **smartctl −l** and **−t** command-line options.]

[ATA only] Failed self-tests outdated by a newer successful extended self−test are ignored. The warning email counter is reset if the number of failed self tests dropped to 0. This typically happens when an extended self−test is run after all bad sectors have been reallocated.

offlinests[,ns] − [ATA only] report if the Offline Data Collection status has changed since the last check. The report will be logged as LOG_CRIT if the new status indicates an error. With some drives the status often changes, therefore ´−l offlinests´ is not enabled by ’−a´ Directive. Appending ´,ns´ (no standby) to this directive is not implemented on Linux.

selfteststs[,ns] − [ATA only] report if the Self-Test execution status has changed since the last check. The report will be logged as LOG_CRIT if the new status indicates an error. Appending ´,ns´ (no standby) to this directive is not implemented on Linux.

scterc,READTIME,WRITETIME − [ATA only] sets the SCT Error Recovery Control settings to the specified values (deciseconds) when **smartd** starts up and has no further effect. 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. [Please see the **smartctl −l scterc** command-line option.]

-e NAME[,VALUE]

[NEW EXPERIMENTAL SMARTD FEATURE] Sets non−SMART device settings when **smartd** starts up and has no further effect. [Please see the **smartctl −−set** command-line option.] Valid arguments are:

aam,[N|off] − [ATA only] Sets the Automatic Acoustic Management (AAM) feature.

apm,[N|off] − [ATA only] Sets the Advanced Power Management (APM) feature.

lookahead,[on|off] − [ATA only] Sets the read look-ahead feature.

security-freeze − [ATA only] Sets ATA Security feature to frozen mode.

standby,[N|off] − [ATA only] Sets the standby (spindown) timer and places the drive in the IDLE mode.

wcache,[on|off] − [ATA only] Sets the volatile write cache feature.

−s REGEXP

Run Self-Tests or Offline Immediate Tests, at scheduled times. A Self- or Offline Immediate Test will be run at the end of periodic device polling, if all 12 characters of the string **T/MM/DD/d/HH** match the extended regular expression **REGEXP**. Here:

T is the type of the test. The values that **smartd** will try to match (in turn) are: ´L´ for a **L**ong Self-Test, ´S´ for a **S**hort Self-Test, ´C´ for a **C**onveyance Self-Test (ATA only), and ´O´ for an **O**ffline Immediate Test (ATA only). As soon as a match is found, the test will be started and no additional matches will be sought for that device and that polling cycle.

To run scheduled Selective Self-Tests, use ´n´ for **n**ext span, ´r´ to **r**edo last span, or ´c´ to **c**ontinue with next span or redo last span based on status of last test. The LBA range is based on the first span from the last test. See the **smartctl −t select,[next|redo|cont]** options for further info.

Some disks (e.g. WD) do not preserve the selective self test log accross power cycles. If state persistence (´−s´ option) is enabled, the last test span is preserved by smartd and used if (and only if) the selective self test log is empty.

MM
is the month of the year, expressed with two decimal digits. The range is from 01 (January) to 12 (December) inclusive. Do **not** use a single decimal digit or the match will always fail!

DD is the day of the month, expressed with two decimal digits. The range is from 01 to 31 inclusive. Do **not** use a single decimal digit or the match will always fail!

d is the day of the week, expressed with one decimal digit. The range is from 1 (Monday) to 7 (Sunday) inclusive.

HH
is the hour of the day, written with two decimal digits, and given in hours after midnight. The range is 00 (midnight to just before 1am) to 23 (11pm to just before midnight) inclusive. Do **not** use a single decimal digit or the match will always fail!

Some examples follow. In reading these, keep in mind that in extended regular expressions a dot **´.´** matches any single character, and a parenthetical expression such as **´(A|B|C)´** denotes any one of the three possibilities **A**, **B**, or **C**.

To schedule a short Self-Test between 2-3am every morning, use:
 −s S/../.././02
To schedule a long Self-Test between 4-5am every Sunday morning, use:
 −s L/../../7/04
To schedule a long Self-Test between 10-11pm on the first and fifteenth day of each month, use:
 −s L/../(01|15)/./22
To schedule an Offline Immediate test after every midnight, 6am, noon,and 6pm, plus a Short Self-Test daily at 1-2am and a Long Self-Test every Saturday at 3-4am, use:
 −s (O/../.././(00|06|12|18)|S/../.././01|L/../../6/03)
If Long Self-Tests of a large disks take longer than the system uptime, a full disk test can be performed by several Selective Self-Tests. To setup a full test of a 1TB disk within 20 days (one 50GB span each day), run this command once:

```
  smartctl -t select,0-99999999 /dev/sda
```

To run the next test spans on Monday-Friday between 12-13am, run smartd with this directive:
 −s n/../../[1-5]/12

Scheduled tests are run immediately following the regularly-scheduled device polling, if the current local date, time, and test type, match **REGEXP**. By default the regularly-scheduled device polling occurs every thirty minutes after starting **smartd**. Take caution if you use the ´−i´ option to make this polling interval more than sixty minutes: the poll times may fail to coincide with any of the testing times that you have specified with **REGEXP**. In this case the test will be run following the next device polling.

Before running an offline or self-test, **smartd** checks to be sure that a self-test is not already running. If a self-test **is** already running, then this running self test will **not** be interrupted to begin another test.

smartd will not attempt to run **any** type of test if another test was already started or run in the same hour.

To avoid performance problems during system boot, **smartd** will not attempt to run any scheduled tests following the very first device polling (unless ´−q onecheck´ is specified).

Each time a test is run, **smartd** will log an entry to SYSLOG. You can use these or the ’-q showtests’ command-line option to verify that you constructed **REGEXP** correctly. The matching order (**L** before **S** before **C** before **O**) ensures that if multiple test types are all scheduled for the same hour, the longer test type has precedence. This is usually the desired behavior.

If the scheduled tests are used in conjunction with state persistence (´−s´ option), smartd will also try to match the hours since last shutdown (or 90 days at most). If any test would have been started during downtime, the longest (see above) of these tests is run after second device polling.

If the ´−n´ directive is used and any test would have been started during disk standby time, the longest of these tests is run when the disk is active again.

Unix users: please beware that the rules for extended regular expressions [regex(7)] are **not** the same as the rules for file−name pattern matching by the shell [glob(7)]. **smartd** will issue harmless informational warning messages if it detects characters in **REGEXP** that appear to indicate that you have made this mistake.

−m ADD

Send a warning email to the email address **ADD** if the ´−H´, ´−l´, ´−f´, ´−C´, or ´−O´ Directives detect a failure or a new error, or if a SMART command to the disk fails. This Directive only works in conjunction with these other Directives (or with the equivalent default ´−a´ Directive).

To prevent your email in-box from getting filled up with warning messages, by default only a single warning will be sent for each of the enabled alert types, ´−H´, ´−l´, ´−f´, ´−C´, or ´−O´ even if more than one failure or error is detected or if the failure or error persists. [This behavior can be modified; see the ´−M´ Directive below.]

To send email to more than one user, please use the following "comma separated" form for the address: **user1@add1,user2@add2,...,userN@addN** (with no spaces).

To test that email is being sent correctly, use the ´−M test´ Directive described below to send one test email message on **smartd** startup.

By default, email is sent using the system **mail** command. In order that **smartd** find the mail command (normally /bin/mail) an executable named **´mail´** must be in the path of the shell or environment from which **smartd** was started. If you wish to specify an explicit path to the mail executable (for example /usr/local/bin/mail) or a custom script to run, please use the ´−M exec´ Directive below.

Note also that there is a special argument **<nomailer>** which can be given to the ´−m´ Directive in conjunction with the ´−M exec´ Directive. Please see below for an explanation of its effect.

If the mailer or the shell running it produces any STDERR/STDOUT output, then a snippet of that output will be copied to SYSLOG. The remainder of the output is discarded. If problems are encountered in sending mail, this should help you to understand and fix them. If you have mail problems, we recommend running **smartd** in debug mode with the ´-d´ flag, using the ´-M test´ Directive described below.

−M TYPE

These Directives modify the behavior of the **smartd** email warnings enabled with the ´−m´ email Directive described above. These ´−M´ Directives only work in conjunction with the ´−m´ Directive and can not be used without it.

Multiple −M Directives may be given. If more than one of the following three −M Directives are given (example: −M once −M daily) then the final one (in the example, −M daily) is used.

The valid arguments to the −M Directive are (one of the following three):

once − send only one warning email for each type of disk problem detected. This is the default unless state persistence (´−s´ option) is enabled.

daily − send additional warning reminder emails, once per day, for each type of disk problem detected. This is the default if state persistence (´−s´ option) is enabled.

diminishing − send additional warning reminder emails, after a one-day interval, then a two-day interval, then a four-day interval, and so on for each type of disk problem detected. Each interval is twice as long as the previous interval.

If a disk problem is no longer detected, the internal email counter is reset. If the problem reappears a new warning email is sent immediately.

In addition, one may add zero or more of the following Directives:

test − send a single test email immediately upon **smartd** startup. This allows one to verify that email is delivered correctly. Note that if this Directive is used, **smartd** will also send the normal email warnings that were enabled with the ´−m´ Directive, in addition to the single test email!

exec PATH − run the executable PATH instead of the default mail command, when **smartd** needs to send email. PATH must point to an executable binary file or script.

By setting PATH to point to a customized script, you can make **smartd** perform useful tricks when a disk problem is detected (beeping the console, shutting down the machine, broadcasting warnings to all logged-in users, etc.) But please be careful. **smartd** will **block** until the executable PATH returns, so if your executable hangs, then **smartd** will also hang. Some sample scripts are included in /usr/share/doc/smartmontools/examplescripts/.

The return status of the executable is recorded by **smartd** in SYSLOG. The executable is not expected to write to STDOUT or STDERR. If it does, then this is interpreted as indicating that something is going wrong with your executable, and a fragment of this output is logged to SYSLOG to help you to understand the problem. Normally, if you wish to leave some record behind, the executable should send mail or write to a file or device.

Before running the executable, **smartd** sets a number of environment variables. These environment variables may be used to control the executable´s behavior. The environment variables exported by **smartd** are:

SMARTD_MAILER

is set to the argument of −M exec, if present or else to ´mail´ (examples: /bin/mail, mail).

SMARTD_DEVICE

is set to the device path (examples: /dev/hda, /dev/sdb).

SMARTD_DEVICETYPE

is set to the device type specified by ´-d´ directive or ´auto´ if none.

SMARTD_DEVICESTRING

is set to the device description. For SMARTD_DEVICETYPE of ata or scsi, this is the same as SMARTD_DEVICE. For 3ware RAID controllers, the form used is ´/dev/sdc [3ware_disk_01]´. For HighPoint RocketRAID controller, the form is ´/dev/sdd [hpt_1/1/1]´ under Linux or ´/dev/hptrr [hpt_1/1/1]´ under FreeBSD. For Areca controllers, the form is ´/dev/sg2 [areca_disk_09]´ on Linux or ´/dev/arcmsr0 [areca_disk_09]´ on FreeBSD. In these cases the device string contains a space and is NOT quoted. So to use $SMARTD_DEVICESTRING in a bash script you should probably enclose it in double quotes.

SMARTD_FAILTYPE

gives the reason for the warning or message email. The possible values that it takes and their meanings are:
EmailTest: this is an email test message.
Health: the SMART health status indicates imminent failure.
Usage: a usage Attribute has failed.
SelfTest: the number of self-test failures has increased.
ErrorCount: the number of errors in the ATA error log has increased.
CurrentPendingSector: one of more disk sectors could not be read and are marked to be reallocated (replaced with spare sectors).
OfflineUncorrectableSector: during off−line testing, or self−testing, one or more disk sectors could not be read.
Temperature: Temperature reached critical limit (see −W directive).
FailedHealthCheck: the SMART health status command failed.
FailedReadSmartData: the command to read SMART Attribute data failed.
FailedReadSmartErrorLog: the command to read the SMART error log failed.
FailedReadSmartSelfTestLog: the command to read the SMART self-test log failed.
FailedOpenDevice: the open() command to the device failed.

SMARTD_ADDRESS

is determined by the address argument ADD of the ´−m´ Directive. If ADD is **<nomailer>**, then **SMARTD_ADDRESS** is not set. Otherwise, it is set to the comma-separated-list of email addresses given by the argument ADD, with the commas replaced by spaces (example:admin@example.com root). If more than one email address is given, then this string will contain space characters and is NOT quoted, so to use it in a bash script you may want to enclose it in double quotes.

SMARTD_MESSAGE

is set to the one sentence summary warning email message string from **smartd**. This message string contains space characters and is NOT quoted. So to use $SMARTD_MESSAGE in a bash script you should probably enclose it in double quotes.

SMARTD_FULLMESSAGE

is set to the contents of the entire email warning message string from **smartd**. This message string contains space and return characters and is NOT quoted. So to use $SMARTD_FULLMESSAGE in a bash script you should probably enclose it in double quotes.

SMARTD_TFIRST

is a text string giving the time and date at which the first problem of this type was reported. This text string contains space characters and no newlines, and is NOT quoted. For example:
Sun Feb 9 14:58:19 2003 CST

SMARTD_TFIRSTEPOCH

is an integer, which is the unix epoch (number of seconds since Jan 1, 1970) for **SMARTD_TFIRST**.

The shell which is used to run PATH is system-dependent. For vanilla Linux/glibc it´s bash. For other systems, the man page for **popen**(3) should say what shell is used.

If the ´−m ADD´ Directive is given with a normal address argument, then the executable pointed to by PATH will be run in a shell with STDIN receiving the body of the email message, and with the same command-line arguments:
-s "$SMARTD_SUBJECT" $SMARTD_ADDRESS
that would normally be provided to ´mail´. Examples include:
-m user@home -M exec /bin/mail
-m admin@work -M exec /usr/local/bin/mailto
-m root -M exec /Example_1/bash/script/below

If the ´−m ADD´ Directive is given with the special address argument **<nomailer>** then the executable pointed to by PATH is run in a shell with **no** STDIN and **no** command-line arguments, for example:
-m <nomailer> -M exec /Example_2/bash/script/below
If the executable produces any STDERR/STDOUT output, then **smartd** assumes that something is going wrong, and a snippet of that output will be copied to SYSLOG. The remainder of the output is then discarded.

Some EXAMPLES of scripts that can be used with the ´−M exec´ Directive are given below. Some sample scripts are also included in /usr/share/doc/smartmontools/examplescripts/.

−f [ATA only] Check for ´failure´ of any Usage Attributes. If these Attributes are less than or equal to the threshold, it does NOT indicate imminent disk failure. It "indicates an advisory condition where the usage or age of the device has exceeded its intended design life period." [Please see the **smartctl −A** command-line option.]

−p [ATA only] Report anytime that a Prefail Attribute has changed its value since the last check. [Please see the **smartctl −A** command-line option.]

−u [ATA only] Report anytime that a Usage Attribute has changed its value since the last check. [Please see the **smartctl −A** command-line option.]

−t [ATA only] Equivalent to turning on the two previous flags ´−p´ and ´−u´. Tracks changes in *all* device Attributes (both Prefailure and Usage). [Please see the **smartctl** −A command-line option.]

−i ID [ATA only] Ignore device Attribute number **ID** when checking for failure of Usage Attributes. **ID** must be a decimal integer in the range from 1 to 255. This Directive modifies the behavior of the ´−f´ Directive and has no effect without it.

This is useful, for example, if you have a very old disk and don´t want to keep getting messages about the hours-on-lifetime Attribute (usually Attribute 9) failing. This Directive may appear multiple times for a single device, if you want to ignore multiple Attributes.

−I ID [ATA only] Ignore device Attribute **ID** when tracking changes in the Attribute values. **ID** must be a decimal integer in the range from 1 to 255. This Directive modifies the behavior of the ´−p´, ´−u´, and ´−t´ tracking Directives and has no effect without one of them.

This is useful, for example, if one of the device Attributes is the disk temperature (usually Attribute 194 or 231). It´s annoying to get reports each time the temperature changes. This Directive may appear multiple times for a single device, if you want to ignore multiple Attributes.

−r ID[!]

[ATA only] When tracking, report the *Raw* value of Attribute **ID** along with its (normally reported) *Normalized* value. **ID** must be a decimal integer in the range from 1 to 255. This Directive modifies the behavior of the ´−p´, ´−u´, and ´−t´ tracking Directives and has no effect without one of them. This Directive may be given multiple times.

A common use of this Directive is to track the device Temperature (often ID=194 or 231).

If the optional flag ´!´ is appended, a change of the Normalized value is considered critical. The report will be logged as LOG_CRIT and a warning email will be sent if ´-m´ is specified.

−R ID[!]

[ATA only] When tracking, report whenever the *Raw* value of Attribute **ID** changes. (Normally **smartd** only tracks/reports changes of the *Normalized* Attribute values.) **ID** must be a decimal integer in the range from 1 to 255. This Directive modifies the behavior of the ´−p´, ´−u´, and ´−t´ tracking Directives and has no effect without one of them. This Directive may be given multiple times.

If this Directive is given, it automatically implies the ´−r´ Directive for the same Attribute, so that the Raw value of the Attribute is reported.

A common use of this Directive is to track the device Temperature (often ID=194 or 231). It is also useful for understanding how different types of system behavior affects the values of certain Attributes.

If the optional flag ´!´ is appended, a change of the Raw value is considered critical. The report will be logged as LOG_CRIT and a warning email will be sent if ´-m´ is specified. An example is ´-R 5!´ to warn when new sectors are reallocated.

−C ID[+]

[ATA only] Report if the current number of pending sectors is non-zero. Here **ID** is the id number of the Attribute whose raw value is the Current Pending Sector count. The allowed range of **ID** is 0 to 255 inclusive. To turn off this reporting, use ID = 0. If the **−C ID** option is not given, then it defaults to **−C 197** (since Attribute 197 is generally used to monitor pending sectors). If the name of this Attribute is changed by a ´−v 197,FORMAT,NAME´ directive, the default is changed to **−C 0**.

If ´+´ is specified, a report is only printed if the number of sectors has increased between two check cycles. Some disks do not reset this attribute when a bad sector is reallocated. See also ´−v 197,increasing´ below.

The warning email counter is reset if the number of pending sectors dropped to 0. This typically happens when all pending sectors have been reallocated or could be read again.

A pending sector is a disk sector (containing 512 bytes of your data) which the device would like to mark as ‘‘bad" and reallocate. Typically this is because your computer tried to read that sector, and the read failed because the data on it has been corrupted and has inconsistent Error Checking and Correction (ECC) codes. This is important to know, because it means that there is some unreadable data on the disk. The problem of figuring out what file this data belongs to is operating system and file system specific. You can typically force the sector to reallocate by writing to it (translation: make the device substitute a spare good sector for the bad one) but at the price of losing the 512 bytes of data stored there.

−U ID[+]

[ATA only] Report if the number of offline uncorrectable sectors is non-zero. Here **ID** is the id number of the Attribute whose raw value is the Offline Uncorrectable Sector count. The allowed range of **ID** is 0 to 255 inclusive. To turn off this reporting, use ID = 0. If the **−U ID** option is not given, then it defaults to **−U 198** (since Attribute 198 is generally used to monitor offline uncorrectable sectors). If the name of this Attribute is changed by a ´−v 198,FORMAT,NAME´ (except ´−v 198,FORMAT,Offline_Scan_UNC_SectCt´), directive, the default is changed to **−U 0**.

If ´+´ is specified, a report is only printed if the number of sectors has increased since the last check cycle. Some disks do not reset this attribute when a bad sector is reallocated. See also ´−v 198,increasing´ below.

The warning email counter is reset if the number of offline uncorrectable sectors dropped to 0. This typically happens when all offline uncorrectable sectors have been reallocated or could be read again.

An offline uncorrectable sector is a disk sector which was not readable during an off−line scan or a self−test. This is important to know, because if you have data stored in this disk sector, and you need to read it, the read will fail. Please see the previous ´−C´ option for more details.

−W DIFF[,INFO[,CRIT]]

Report if the current temperature had changed by at least **DIFF** degrees since last report, or if new min or max temperature is detected. Report or Warn if the temperature is greater or equal than one of **INFO** or **CRIT** degrees Celsius. If the limit **CRIT** is reached, a message with loglevel **´LOG_CRIT´** will be logged to syslog and a warning email will be send if '-m' is specified. If only the limit **INFO** is reached, a message with loglevel **´LOG_INFO´** will be logged.

The warning email counter is reset if the temperature dropped below **INFO** or **CRIT**-5 if **INFO** is not specified.

If this directive is used in conjunction with state persistence (´−s´ option), the min and max temperature values are preserved across boot cycles. The minimum temperature value is not updated during the first 30 minutes after startup.

To disable any of the 3 reports, set the corresponding limit to 0. Trailing zero arguments may be omitted. By default, all temperature reports are disabled (´-W 0´).

To track temperature changes of at least 2 degrees, use:
−W 2
To log informal messages on temperatures of at least 40 degrees, use:
−W 0,40
For warning messages/mails on temperatures of at least 45 degrees, use:
−W 0,0,45
To combine all of the above reports, use:
−W 2,40,45

For ATA devices, smartd interprets Attribute 194 as Temperature Celsius by default. This can be changed to Attribute 9 or 220 by the drive database or by the ´-v´ directive, see below.

−F TYPE

[ATA only] Modifies the behavior of **smartd** to compensate for some known and understood device firmware bug. The arguments to this Directive are exclusive, so that only the final Directive given is used. The valid values are:

none − Assume that the device firmware obeys the ATA specifications. This is the default, unless the device has presets for ´−F´ in the device database.

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 **smartd** 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 **smartd** to evaluate this quantity in byte−reversed order.

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. If this directive is specified, **smartd** will not skip the next scheduled self−test (see Directive ´−s´ above) in this case.

Note that an explicit ´−F´ Directive will over−ride any preset values for ´−F´ (see the ´−P´ option below).

[Please see the **smartctl −F** command-line option.]

−v ID,FORMAT[:BYTEORDER][,NAME]

[ATA only] Sets a vendor−specific raw value print FORMAT, an optional BYTEORDER and an optional NAME for Attribute ID. This directive may be used multiple times. Please see **smartctl -v** command-line option for further details.

The following arguments affect smartd warning output:

197,increasing − Raw Attribute number 197 (Current Pending Sector Count) is not reset if uncorrectable sectors are reallocated. This sets ´-C 197+´ if no other ´-C´ directive is specified.

198,increasing − Raw Attribute number 198 (Offline Uncorrectable Sector Count) is not reset if uncorrectable sector are reallocated. This sets ´-U 198+´ if no other ´-U´ directive is specified.

−P TYPE

[ATA only] Specifies whether **smartd** should use any preset options that are available for this drive. The valid arguments to this Directive are:

use − use any presets that are available for this drive. This is the default.

ignore − do not use any presets for this drive.

show − show the presets listed for this drive in the database.

showall − show the presets that are available for all drives and then exit.

[Please see the **smartctl −P** command-line option.]

−a Equivalent to turning on all of the following Directives: **´−H´** to check the SMART health status, **´−f´** to report failures of Usage (rather than Prefail) Attributes, **´−t´** to track changes in both Prefailure and Usage Attributes, **´−l error´** to report increases in the number of ATA errors, **´−l selftest´** to report increases in the number of Self-Test Log errors, **´−l selfteststs´** to report changes of Self-Test execution status, **´−C 197´** to report nonzero values of the current pending sector count, and **´−U 198´** to report nonzero values of the offline pending sector count.

Note that −a is the default for ATA devices. If none of these other Directives is given, then −a is assumed.

Comment: ignore the remainder of the line.

\ Continuation character: if this is the last non-white or non-comment character on a line, then the following line is a continuation of the current one.

If you are not sure which Directives to use, I suggest experimenting for a few minutes with **smartctl** to see what SMART functionality your disk(s) support(s). If you do not like voluminous syslog messages, a good choice of **smartd** configuration file Directives might be:
−H −l selftest −l error −f.

If you want more frequent information, use: **-a.**

ADDITIONAL DETAILS ABOUT DEVICESCAN

If a non-comment entry in the configuration file is the text string **DEVICESCAN** in capital letters, then **smartd** will ignore any remaining lines in the configuration file, and will scan for devices.

Configuration entries for devices not found by the platform−specific device scanning may precede the **DEVICESCAN** entry.

If **DEVICESCAN** is not followed by any Directives, then smartd will scan for both ATA and SCSI devices, and will monitor all possible SMART properties of any devices that are found.

DEVICESCAN may optionally be followed by any valid Directives, which will be applied to all devices that are found in the scan. For example
DEVICESCAN -m root@example.com
will scan for all devices, and then monitor them. It will send one email warning per device for any problems that are found.
DEVICESCAN -d ata -m root@example.com
will do the same, but restricts the scan to ATA devices only.
DEVICESCAN -H -d ata -m root@example.com
will do the same, but only monitors the SMART health status of the devices, (rather than the default −a, which monitors all SMART properties).

EXAMPLES OF SHELL SCRIPTS FOR ´−M exec´

These are two examples of shell scripts that can be used with the ´−M exec PATH´ Directive described previously. The paths to these scripts and similar executables is the PATH argument to the ´−M exec PATH´ Directive.

Example 1: This script is for use with ´−m ADDRESS -M exec PATH´. It appends the output of **smartctl -a** to the output of the smartd email warning message and sends it to ADDRESS.

```
#! /bin/bash

# Save the email message (STDIN) to a file:
cat > /root/msg

# Append the output of smartctl -a to the message:
/usr/sbin/smartctl -a -d $SMART_DEVICETYPE $SMARTD_DEVICE >> /root/msg

# Now email the message to the user at address ADD:
/bin/mail -s "$SMARTD_SUBJECT" $SMARTD_ADDRESS < /root/msg
```

Example 2: This script is for use with ´−m <nomailer> −M exec PATH´. It warns all users about a disk problem, waits 30 seconds, and then powers down the machine.

```
#! /bin/bash

# Warn all users of a problem
wall ´Problem detected with disk: ´ "$SMARTD_DEVICESTRING"
wall ´Warning message from smartd is: ´ "$SMARTD_MESSAGE"
wall ´Shutting down machine in 30 seconds... ´
```

```
# Wait half a minute
sleep 30

# Power down the machine
/sbin/shutdown -hf now
```

Some example scripts are distributed with the smartmontools package, in /usr/share/doc/smartmontools/examplescripts/.

Please note that these scripts typically run as root, so any files that they read/write should not be writable by ordinary users or reside in directories like /tmp that are writable by ordinary users and may expose your system to symlink attacks.

As previously described, if the scripts write to STDOUT or STDERR, this is interpreted as indicating that there was an internal error within the script, and a snippet of STDOUT/STDERR is logged to SYSLOG. The remainder is flushed.

AUTHOR

Bruce Allen smartmontools−support@lists.sourceforge.net
University of Wisconsin − Milwaukee Physics Department

CONTRIBUTORS

The following have made large contributions to smartmontools:
Casper Dik (Solaris SCSI interface)
Christian Franke (Windows interface, C++ redesign, USB support, ...)
Douglas Gilbert (SCSI subsystem)
Guido Guenther (Autoconf/Automake packaging)
Geoffrey Keating (Darwin ATA interface)
Eduard Martinescu (FreeBSD interface)
Frédéric L. W. Meunier (Web site and Mailing list)
Gabriele Pohl (Web site and Wiki, conversion from CVS to SVN)
Keiji Sawada (Solaris ATA interface)
Manfred Schwarb (Drive database)
Sergey Svishchev (NetBSD interface)
David Snyder and Sergey Svishchev (OpenBSD interface)
Phil Williams (User interface and drive database)
Shengfeng Zhou (Linux/FreeBSD HighPoint RocketRAID interface)
Many other individuals have made smaller contributions and corrections.

CREDITS

This code was derived from the smartsuite package, written by Michael Cornwell, and from the previous UCSC smartsuite package. It extends these to cover ATA−5 disks. This code was originally developed as a Senior Thesis by Michael Cornwell at the Concurrent Systems Laboratory (now part of the Storage Systems Research Center), Jack Baskin School of Engineering, University of California, Santa Cruz. **http://ssrc.soe.ucsc.edu/** .

HOME PAGE FOR SMARTMONTOOLS:

Please see the following web site for updates, further documentation, bug reports and patches: **http://smartmontools.sourceforge.net/**

SEE ALSO:

smartd(8), **smartctl**(8), **syslogd**(8), **syslog.conf**(5), **badblocks**(8), **ide−smart**(8), **regex**(7).

SVN ID OF THIS PAGE:

$Id: smartd.conf.5.in 3566 2012-06-19 21:26:28Z chrfranke $