

**NAME**

hddtemp - Utility to monitor hard drive temperature

**SYNOPSIS**

**hddtemp** [*options*] [*type:*]*disk*...

**DESCRIPTION**

**hddtemp** will give you the temperature of your hard drive by reading Self-Monitoring Analysis and Reporting Technology (S.M.A.R.T.) information on drives that support this feature. Only modern hard drives have a temperature sensor. **hddtemp** supports reading S.M.A.R.T. information from SCSI drives too. **hddtemp** can work as simple command line tool or as a daemon.

You can specify one or more device drive path, where each path can be prefixed with a **type** like PATA, SATA or SCSI to force **hddtemp** too use one of these type (because detection can fail).

**OPTIONS**

The program follows the usual GNU command line syntax, with long options starting with two dashes ('-'). A summary of options is included below.

**-b, --drivebase**

Display the database file that allows **hddtemp** to recognize a supported drive.

**-D, --debug**

Display various S.M.A.R.T. fields and their values. Useful for finding a value that seems to match the temperature and/or to send a report. (done for every drive supplied)

**-d, --daemon**

Execute **hddtemp** in TCP/IP daemon mode (port 7634 by default).

**-f, --file=*file***

Specify the database file to use.

**-F, --foreground**

Don't fork into the background even in daemon mode. This is useful when running under a process supervisor.

**-l, --listen=*addr***

Listen on a specific address. *addr* is a string containing a host name or a numeric host address string. The numeric host address string is a dotted-decimal IPv4 address or an IPv6 hex address.

**-n, --numeric**

Print only the temperature (without the unit).

**-p, --port=#**

Port number to listen to (in TCP/IP daemon mode).

**-s, --separator=*char***

Separator to use between fields (in TCP/IP daemon mode). The default separator is '|'.

**-S, --syslog=*s***

Switch to daemon mode and log temperatures to syslog every *s* seconds.

**-q, --quiet**

Don't check if the drive is supported.

**-u, --unit=*C|F***

Force output temperature either in Celsius or Fahrenheit.

**-v, --version**

Display **hddtemp** version number.

**-w, --wake-up**

Wake-up the drive if needed (ATA drives only).

- 4 Listen on IPv4 sockets only.
- 6 Listen on IPv6 sockets only.

## DRIVE DATABASE

If you know your drive has a temperature sensor but it is being reported unsupported, tell me which model and which manufacturer it is, and/or just add a new entry in `/etc/hddtemp.db`. Each line of `hddtemp.db` is either a comment, a blank line or a line containing:

- a regular expression that allow `hddtemp` to recognize a drive or a set of drives from its model name or from a generic model name,
- a value (`ATTRIBUTE_ID` from S.M.A.R.T.),
- a C or an F to set the unit to Celsius or Fahrenheit,
- a description.

Feedback is welcome (see the `REPORT` section below).

## TCP/IP DAEMON MODE

Example of type forcing:

```
# hddtemp SATA:/dev/sda PATA:/dev/hda
```

To test **hddtemp** in daemon mode, start it like this:

```
# hddtemp -d /dev/hd[abcd]
```

and use **telnet** or **netcat** (which is known as **nc** on some systems) to get a reply:

```
# netcat localhost 7634
```

The drive database is read only once at startup, so **hddtemp** must be restarted if the database is updated for the changes to take effect.

## REPORT

As I receive a lot of reports, things must be clarified. When running `hddtemp` with debug options, `hddtemp` will show sort of a dump of S.M.A.R.T. data. Each field corresponds to an information field. The standard field for drive temperature is 194. But this is not always the case (mostly for older drives). Even if your drive has S.M.A.R.T. capabilities, it doesn't necessarily mean that it can report its temperature. So, things must be determined through experimentation.

So, you can try to guess which field by is the good one by running `hddtemp` at regular intervals:

- just after starting up your PC/server/station,
- after opening a window (a physical window :),
- after opening the case,
- whatever you can think of...

and looking for a field's value that would increase or decrease depending on what effect you want to induce. Be careful, fields 4, 9, and 12 are often reported to match a temperature field but after some investigation they do not. But fields 194 (which is the standard field for temperature) and 231 are good candidates.

Then, you can send me a report with outputs from '`hddtemp --debug ...`', '`smartctl`' or '`hdparm -i ...`', and/or add an entry in `hddtemp.db` yourself.

## BUGS

If `hddtemp` crashes (yes, it might) for some unknown reasons, look for a file named `hddtemp.backtrace.<PID>.XXXXXX` (where `XXXXXX` is a random number generated at runtime) in `/tmp`. Then, you can then send me this file and the `hddtemp` binary. The backtrace functionality is currently supported on i386 architectures only.

**SEE ALSO**

[smartctl\(8\)](#), [syslog\(3\)](#), [syslogd\(8\)](#).

**AUTHOR**

Emmanuel Varagnat ([hddtemp@guzu.net](mailto:hddtemp@guzu.net)).

This manual page was originally written by Aurelien Jarno <[aurel32@debian.org](mailto:aurel32@debian.org)>, for the Debian GNU/Linux system (but may be used by others).