NAME

watch − execute a program periodically, showing output fullscreen

SYNOPSIS

watch [*options*] *command*

DESCRIPTION

watch runs *command* repeatedly, displaying its output and errors (the first screenfull). This allows you to watch the program output change over time. By default, *command* is run every 2 seconds and **watch** will run until interrupted.

OPTIONS

−d, **−−differences**[=*permanent*]
: Highlight the differences between successive updates. If the optional *permanent* argument is specified then **watch** will show all changes since the first iteration.

−n, **−−interval** *seconds*
: Specify update interval. The command will not allow quicker than 0.1 second interval, in which the smaller values are converted. Both '.' and ',' work for any locales. The WATCH_INTERVAL environment can be used to persistently set a non-default interval (following the same rules and formatting).

−p, **−−precise**
: Make **watch** attempt to run *command* every **−−interval** *seconds*. Try it with **ntptime** (if present) and notice how the fractional seconds stays (nearly) the same, as opposed to normal mode where they continuously increase.

−t, **−−no−title**
: Turn off the header showing the interval, command, and current time at the top of the display, as well as the following blank line.

−b, **−−beep**
: Beep if command has a non-zero exit.

−e, **−−errexit**
: Freeze updates on command error, and exit after a key press.

−g, **−−chgexit**
: Exit when the output of *command* changes.

−c, **−−color**
: Interpret ANSI color and style sequences.

−x, **−−exec**
: Pass *command* to **exec**(2) instead of **sh −c** which reduces the need to use extra quoting to get the desired effect.

−w, **−−no−linewrap**
: Turn off line wrapping. Long lines will be truncated instead of wrapped to the next line.

−h, **−−help**
: Display help text and exit.

−v, **−−version**
: Display version information and exit.

EXIT STATUS

0	Success.
1	Various failures.
2	Forking the process to watch failed.
3	Replacing child process stdout with write side pipe failed.
4	Command execution failed.

5 Closing child process write pipe failed.
7 IPC pipe creation failed.
8 Getting child process return value with **waitpid**(2) failed, or command exited up on error.
other The watch will propagate command exit status as child exit status.

ENVIRONMENT

The behaviour of **watch** is affected by the following environment variables.

WATCH_INTERVAL
Update interval, follows the same rules as the **−−interval** command line option.

NOTES

POSIX option processing is used (i.e., option processing stops at the first non−option argument). This means that flags after *command* don't get interpreted by **watch** itself.

BUGS

Upon terminal resize, the screen will not be correctly repainted until the next scheduled update. All **−−differences** highlighting is lost on that update as well.

Non-printing characters are stripped from program output. Use **cat -v** as part of the command pipeline if you want to see them.

Combining Characters that are supposed to display on the character at the last column on the screen may display one column early, or they may not display at all.

Combining Characters never count as different in **−−differences** mode. Only the base character counts.

Blank lines directly after a line which ends in the last column do not display.

−−precise mode doesn't yet have advanced temporal distortion technology to compensate for a *command* that takes more than **−−interval** *seconds* to execute. **watch** also can get into a state where it rapid-fires as many executions of *command* as it can to catch up from a previous executions running longer than **−−interval** (for example, **netstat** taking ages on a DNS lookup).

EXAMPLES

To watch for mail, you might do
watch −n 60 from
To watch the contents of a directory change, you could use
watch −d ls −l
If you're only interested in files owned by user joe, you might use
watch −d 'ls −l | fgrep joe'
To see the effects of quoting, try these out
watch echo $$
watch echo '$$'
watch echo "'"'$$'"'"
To see the effect of precision time keeping, try adding **−p** to
watch −n 10 sleep 1
You can watch for your administrator to install the latest kernel with
watch uname −r
(Note that **−p** isn't guaranteed to work across reboots, especially in the face of **ntpdate** (if present) or other bootup time-changing mechanisms)