

NAME

nano – Nano’s ANOther editor, inspired by Pico

SYNOPSIS

nano [*options*] [[*+line*[,*column*]] *file*]...

NOTICE

Starting with version 4.0, **nano** no longer hard-wraps an overlong line by default. It further uses smooth scrolling by default, and by default includes the line below the title bar into the editing area.

If you want the old, Pico behavior back, you can use **--breaklonglines**, **--jumpyscrolling**, and **--empty-line** (or **-bje** for short).

DESCRIPTION

nano is a small and friendly editor. It copies the look and feel of Pico, but is free software, and implements several features that Pico lacks, such as: opening multiple files, scrolling per line, undo/redo, syntax coloring, line numbering, and soft-wrapping overlong lines.

When giving a filename on the command line, the cursor can be put on a specific line by adding the line number with a plus sign (+) before the filename, and even in a specific column by adding it with a comma.

As a special case: if instead of a filename a dash (–) is given, **nano** will read data from standard input.

EDITING

Entering text and moving around in a file is straightforward: typing the letters and using the normal cursor movement keys. Commands are entered by using the Control (^) and the Alt or Meta (M–) keys. Typing ^K deletes the current line and puts it in the cutbuffer. Consecutive ^Ks will put all deleted lines together in the cutbuffer. Any cursor movement or executing any other command will cause the next ^K to overwrite the cutbuffer. A ^U will paste the current contents of the cutbuffer at the current cursor position.

When a more precise piece of text needs to be cut or copied, one can mark its start with ^6, move the cursor to its end (the marked text will be highlighted), and then use ^K to cut it, or M–6 to copy it to the cutbuffer. One can also save the marked text to a file with ^O, or spell check it with ^T.

On some terminals, text can be selected also by holding down Shift while using the arrow keys. Holding down the Ctrl or Alt key too will increase the stride. Any cursor movement without Shift being held will cancel such a selection.

The two lines at the bottom of the screen show some important commands; the built-in help (^G) lists all the available ones. The default key bindings can be changed via a *nanorc* file -- see **nanorc(5)**.

OPTIONS

-A, --smarthome

Make the Home key smarter. When Home is pressed anywhere but at the very beginning of non-whitespace characters on a line, the cursor will jump to that beginning (either forwards or backwards). If the cursor is already at that position, it will jump to the true beginning of the line.

-B, --backup

When saving a file, back up the previous version of it, using the current filename suffixed with a tilde (~).

-C directory, --backupdir=directory

Make and keep not just one backup file, but make and keep a uniquely numbered one every time a file is saved -- when backups are enabled (**-B**). The uniquely numbered files are stored in the

specified *directory*.

-D, --boldtext

Use bold text instead of reverse video text.

-E, --tabstospaces

Convert typed tabs to spaces.

-F, --multibuffer

Read a file into a new buffer by default.

-G, --locking

Use vim-style file locking when editing files.

-H, --historylog

Save the last hundred search strings and replacement strings and executed commands, so they can be easily reused in later sessions.

-I, --ignorercfiles

Don't look at the system's *nanorc* nor at the user's *nanorc*.

-J *number*, --guidestripe=*number*

Draw a vertical stripe at the given column, to help judge the width of the text. (The color of the stripe can be changed with **set stripecolor** in your *nanorc* file.)

-K, --rawsequences

Interpret escape sequences directly (instead of asking **ncurses** to translate them). If you need this option to get your keyboard to work properly, please report a bug. Using this option disables **nano**'s mouse support.

-L, --nonewlines

Don't automatically add a newline when a text does not end with one. (This can cause you to save non-POSIX text files.)

-M, --trimblanks

Snip trailing whitespace from the wrapped line when automatic hard-wrapping occurs or when text is justified.

-N, --noconvert

Disable automatic conversion of files from DOS/Mac format.

-O, --morespace

Obsolete and ignored option, since the line below the title bar is included into the editing space by default. If you prefer to keep this line blank, use **-e** or **--emptyline**.

-P, --positionlog

For the 200 most recent files, log the last position of the cursor, and place it at that position again upon reopening such a file.

-Q "*regex*", --quotestr="*regex*"

Set the regular expression for matching the quoting part of a line. The default value is `"^([\\t]*([!#%&;>}]|/))+"`. (Note that `\\t` stands for an actual Tab.) This makes it possible to rejustify blocks of quoted text when composing email, and to rewrap blocks of line comments when writing source code.

-R, --restricted

Restricted mode: don't read or write to any file not specified on the command line. This means: don't read or write history files; don't allow suspending; don't allow spell checking; don't allow a file to be appended to, prepended to, or saved under a different name if it already has one; and don't make backup files. Restricted mode can also be activated by invoking **nano** with any name beginning with 'r' (e.g. "rnano").

- S, --smooth**
Obsolete and ignored option, since smooth scrolling has become the default. If you prefer the chunk-by-chunk scrolling behavior, use **-j** or **--jumpyscrolling**.
- T *number*, --tabsize=*number***
Set the size (width) of a tab to *number* columns. The value of *number* must be greater than 0. The default value is 8.
- U, --quickblank**
Do quick status-bar blanking: status-bar messages will disappear after 1 keystroke instead of 25. Note that option **-c** (**--constantshow**) overrides this.
- V, --version**
Show the current version number and exit.
- W, --wordbounds**
Detect word boundaries differently by treating punctuation characters as part of a word.
- X "*characters*", --wordchars="*characters*"**
Specify which other characters (besides the normal alphanumeric ones) should be considered as part of a word. This overrides option **-W** (**--wordbounds**).
- Y *name*, --syntax=*name***
Specify the name of the syntax highlighting to use from among the ones defined in the *nanorc* files.
- Z, --zap**
Let an unmodified Backspace or Delete erase the marked region (instead of a single character, and without affecting the cutbuffer).
- a, --atblanks**
When doing soft line wrapping, wrap lines at whitespace instead of always at the edge of the screen.
- b, --breaklonglines**
Automatically hard-wrap the current line when it becomes overlong. (This option is the opposite of **-w** (**--nowrap**) -- the last one given takes effect.)
- c, --constantshow**
Constantly show the cursor position on the status bar. Note that this overrides option **-U** (**--quickblank**).
- d, --rebinddelete**
Interpret the Delete and Backspace keys differently so that both Backspace and Delete work properly. You should only use this option when on your system either Backspace acts like Delete or Delete acts like Backspace.
- e, --emptyline**
Do not use the line below the title bar, leaving it entirely blank.
- g, --showcursor**
Make the cursor visible in the file browser (putting it on the highlighted item) and in the help viewer. Useful for braille users and people with poor vision.
- h, --help**
Show a summary of the available command-line options and exit.
- i, --autoindent**
Automatically indent a newly created line to the same number of tabs and/or spaces as the previous line (or as the next line if the previous line is the beginning of a paragraph).
- j, --jumpyscrolling**
Scroll the buffer contents per half-screen instead of per line.

-k, --cutfromcursor

Make the 'Cut Text' command (normally **^K**) cut from the current cursor position to the end of the line, instead of cutting the entire line.

-l, --linenumbers

Display line numbers to the left of the text area.

-m, --mouse

Enable mouse support, if available for your system. When enabled, mouse clicks can be used to place the cursor, set the mark (with a double click), and execute shortcuts. The mouse will work in the X Window System, and on the console when **gpm** is running. Text can still be selected through dragging by holding down the Shift key.

-n, --noread

Treat any name given on the command line as a new file. This allows **nano** to write to named pipes: it will start with a blank buffer, and will write to the pipe when the user saves the "file". This way **nano** can be used as an editor in combination with for instance **gpg** without having to write sensitive data to disk first.

-o directory, --operatingdir=directory

Set the operating directory. This makes **nano** set up something similar to a chroot.

-p, --preserve

Preserve the XON and XOFF sequences (**^Q** and **^S**) so they will be caught by the terminal.

-r number, --fill=number

Set the target width for justifying and automatic hard-wrapping at this *number* of columns. If the value is 0 or less, wrapping will occur at the width of the screen minus *number* columns, allowing the wrap point to vary along with the width of the screen if the screen is resized. The default value is -8.

-s program, --speller=program

Use this alternative spell checker command.

-t, --tempfile

Save a changed buffer without prompting (when exiting with **^X**).

-u, --unix

Save a file by default in Unix format. This overrides nano's default behavior of saving a file in the format that it had. (This option has no effect when you also use **--noconvert**.)

-v, --view

Just view the file and disallow editing: read-only mode. This mode allows the user to open also other files for viewing, unless **--restricted** is given too.

-w, --nowrap

Do not automatically hard-wrap the current line when it becomes overlong. This is the default. (This option is the opposite of **-b** (**--breaklonglines**) -- the last one given takes effect.)

-x, --nohelp

Don't show the two help lines at the bottom of the screen.

-y, --afterends

Make **Ctrl+Right** stop at word ends instead of beginnings.

-z, --suspend

Enable the suspend ability.

-\$, --softwrap

Enable 'soft wrapping'. This will make **nano** attempt to display the entire contents of any line, even if it is longer than the screen width, by continuing it over multiple screen lines. Since '\$' normally refers to a variable in the Unix shell, you should specify this option last when using other

options (e.g. `'nano -wS$'`) or pass it separately (e.g. `'nano -wS -$'`).

TOGGLES

Several of the above options can be switched on and off also while **nano** is running. For example, **M-L** toggles the hard-wrapping of long lines, **M-S** toggles soft-wrapping, **M-N** toggles line numbers, **M-M** toggles the mouse, **M-I** auto-indentation, and **M-X** the help lines. See at the end of the **^G** help text for a complete list.

FILES

nano will read two configuration files: first the system's *nanorc* (if it exists), and then the user's *nanorc* (if it exists), either `~/.nanorc` or `$XDG_CONFIG_HOME/nano/nanorc` or `~/.config/nano/nanorc`, whichever is encountered first. See **nanorc(5)** for more information on the possible contents of those files.

NOTES

If no alternative spell checker command is specified on the command line nor in one of the *nanorc* files, **nano** will check the **SPELL** environment variable for one.

In some cases **nano** will try to dump the buffer into an emergency file. This will happen mainly if **nano** receives a **SIGHUP** or **SIGTERM** or runs out of memory. It will write the buffer into a file named *nano.save* if the buffer didn't have a name already, or will add a ".save" suffix to the current filename. If an emergency file with that name already exists in the current directory, it will add ".save" plus a number (e.g. ".save.1") to the current filename in order to make it unique. In multibuffer mode, **nano** will write all the open buffers to their respective emergency files.

BUGS

The recording and playback of keyboard macros works correctly only on a terminal emulator, not on a Linux console (VT), because the latter does not by default distinguish modified from unmodified arrow keys.

Please report any other bugs that you encounter via:
<https://savannah.gnu.org/bugs/?group=nano>.

When nano crashes, it will save any modified buffers to emergency .save files. If you are able to reproduce the crash and you want to get a backtrace, define the environment variable **NANO_NOCATCH**.

HOMEPAGE

<https://nano-editor.org/>

SEE ALSO

nanorc(5)

/usr/share/doc/nano/ (or equivalent on your system)