

NAME

gnome-terminal – A terminal emulator for GNOME

SYNOPSIS

gnome-terminal [*OPTION...*] [*— PROGRAM [ARG...]*]

DESCRIPTION

gnome-terminal is a terminal emulator application for accessing a UNIX shell environment which can be used to run programs available on your system. It supports several profiles, multiple tabs and implements several keyboard shortcuts.

OPTIONS

—help, -h

Show a brief overview of all the options.

—help-all

Show all the options in detail.

—help-gtk

Show all the GTK options.

—help-terminal

Show all the options to select between new terminal tabs or windows.

—help-terminal-options

Show all the options to change the attributes of terminals regardless of whether they are in separate tabs or windows.

—help-window-options

Show all the options to change the attributes of windows containing terminals.

—load-config=FILE

Restore the application to a previously saved state by loading it from a configuration file.

—preferences

Show the preferences window.

—print-environment, -p

Print the environment variables to interact with newly created terminals.

—quiet, -q

Suppress diagnostics.

—verbose, -v

Increase diagnostic verbosity.

—tab

Open a new tab containing a terminal in the last-opened window with the default profile.

—window

Open a new window with a tab containing a terminal with the default profile.

—command, -e=COMMAND

Split the argument to this option into a program and arguments in the same way a shell would, and execute the resulting command-line inside the terminal.

This option is deprecated. Instead, use **—** to terminate the options, and put the program and arguments to execute after it: for example, instead of **gnome-terminal -e "python3 -q"**, prefer to use **gnome-terminal — python3 -q**.

Note that the *COMMAND* is not run via a shell: it is split into words and executed as a program. If shell syntax is required, use the form **gnome-terminal — sh -c '...'**.

—execute PROGRAM [ARGS], -x PROGRAM [ARGS]

Stop parsing options at this point, and interpret all subsequent options as a program and arguments to

execute inside the terminal.

This option is deprecated: use `--` instead. For example, instead of `gnome-terminal -x python3 -q`, prefer to use `gnome-terminal -- python3 -q`.

--fd=FD

Forward file descriptor.

--profile=PROFILE-NAME

Use the given profile instead of the default profile.

--title, -t=TITLE

Set the initial terminal title.

--wait

Wait until the terminal's child exits.

--working-directory=DIRNAME

Set the terminal's working directory.

--zoom=ZOOM

Set the terminal's zoom factor. 1.0 is normal size.

--active

Set the last specified tab as the active one in its window.

--full-screen

Full-screen the window.

--geometry=GEOMETRY

Set the window size as COLSxROWS+X+Y. For example, 80x24 or 80x24+200+200.

--hide-menubar

Turn off the menubar for the window.

--show-menubar

Turn on the menubar for the window.

--maximize

Maximize the window.

--role=ROLE

Set the X window role.

--class=CLASS

Program class as used by the window manager.

--display=DISPLAY

X display to use.

--g-fatal-warnings

Make all warnings fatal.

--gdk-debug=FLAGS

GDK debugging flags to set.

--gdk-no-debug=FLAGS

GDK debugging flags to unset.

--gtk-debug=FLAGS

GTK debugging flags to set.

--gtk-no-debug=FLAGS

GTK debugging flags to unset.

--gtk-module=MODULES

Load additional GTK modules.

--name=NAME

Program name as used by the window manager.

BUGS

Please read <https://wiki.gnome.org/Apps/Terminal/ReportingBugs> on how to report bugs.

EXAMPLES

To run a terminal containing an interactive Python prompt:

```
gnome-terminal --title=Python -- python3 -i
```

To interpret shell syntax in a terminal, either write it in a separate shell script, or use `sh -c`:

```
gnome-terminal -- sh -c 'if [ "$(id -u)" = 0 ]; then ...'
```

SEE ALSO

For further information, visit the website <https://wiki.gnome.org/Apps/Terminal>. There's a list of frequently asked questions at <https://wiki.gnome.org/Apps/Terminal/FAQ>.