



Lesser-Known gedit Keyboard Shortcuts

Serge Y. Stroobandt

Copyright 2013–2016, licensed under [Creative Commons BY-NC-SA](#)

Introduction

Once I really like GNOME's [gedit text editor](#), even though I have since become a [Vim](#) adept. And since the advent of the spoon-fed (i.e. nonforked) [GNOME 3 debacle](#), I switched my desktop to [XFCE](#). Nevertheless, gedit 3 works just fine in XFCE 4.11 regardless of the fact that gedit 3 employs version 3 of the [GIMP](#) toolkit (GTK+), whereas XFCE 4.11 still relies on [GTK+](#) version 2.

Its plugins, both *“official”* and *“less official”* ([LaTeX](#), [collaboration](#),...) are what make gedit a truly complete editor; yet in a very unobtrusive way. As for eye candy, I can wholeheartedly recommend my own dark, relaxing [Oblivio](#) colour scheme.

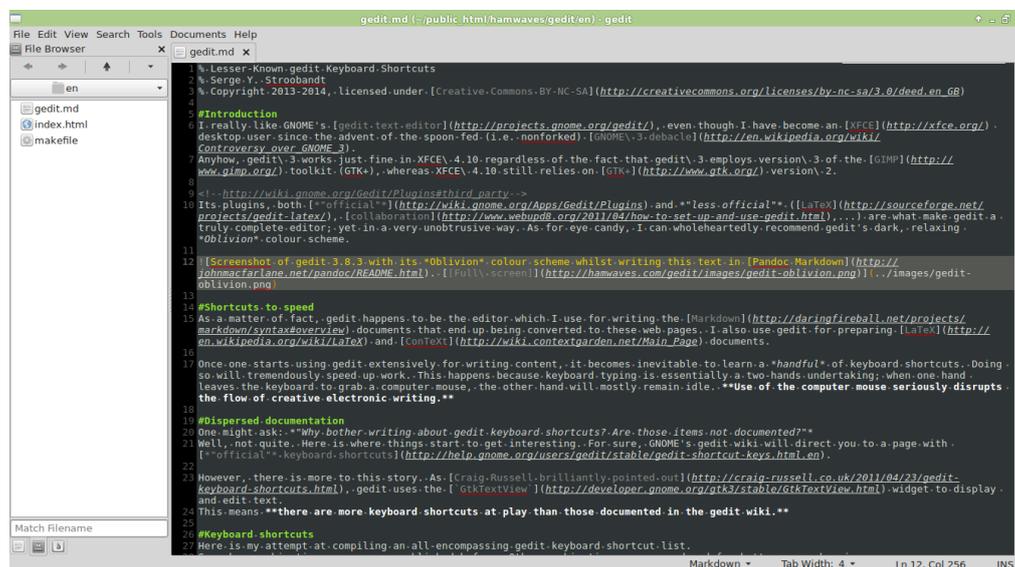


Figure 1: Screenshot of gedit 3.8.3 with its *Oblivion* colour scheme whilst writing this text in [Pandoc Markdown](#). [\[Full screen\]](#)

Shortcuts to speed

As a matter of fact, gedit happens to be the editor which I use for writing the **Markdown** documents that end up being converted to these web pages. I also use gedit for preparing **LaTeX** and **ConTeXt** documents.

Once one starts using gedit extensively for writing content, it becomes inevitable to learn a *handful* of keyboard shortcuts. Doing so will tremendously speed up work. This happens because keyboard typing is essentially a two-hands undertaking; when one hand leaves the keyboard to grab a computer mouse, the other hand will mostly remain idle. Use of the computer mouse seriously disrupts the flow of creative electronic writing.

Dispersed documentation

One might ask: “*Why bother writing about gedit keyboard shortcuts? Are those items not documented?*” Well, not quite. Here is where things start to get interesting. For sure, GNOME’s gedit wiki will direct you to a page with “*official*” **keyboard shortcuts**.

However, there is more to this story. As **Craig Russell** brilliantly pointed out, gedit uses the **GtkTextView** widget to display and edit text. This means there are more keyboard shortcuts at play than those documented in the gedit wiki.

Keyboard shortcuts

Here is my attempt at compiling an all-encompassing gedit keyboard shortcut list. Some key combinations were never published before. Other combinations were reordered for better comprehension. However, by no means I pretend this list to be complete. For example, there is this **specification of GtkTextIter**, which is a component of GtkTextView providing text manipulation. In it, **sentence navigation** gets mentioned. Unfortunately, key-bindings for these sentence functions are nowhere to be found... Finally, I did not include published shortcuts that failed to work in gedit 3.8.3, nor shortcuts that are specific to optionally installed plugins, unless otherwise stated.

Cursor navigation

Table 1: Cursor navigation Ctrl

shortcut	action
Ctrl →	Move the cursor forward by one word .
Ctrl ←	Move the cursor backwards by one word .
Ctrl PageUp or Home	Move the cursor to the beginning of the line .
Ctrl PageDn or End	Move the cursor to the end of the line .
Ctrl ↑	Move the cursor to the beginning of the previous line .
Ctrl ↓	Move the cursor to the end of the next line .
Ctrl 	Move the cursor to a specific line number .
Ctrl Home	Move the cursor to the beginning of the document .
Ctrl End	Move the cursor to the end of the document .
Ctrl Alt PageUp	Switch to the document/side pane tab to the left .
Ctrl Alt PageDn	Switch to the document/side pane tab to the right .
Alt 1 ... 9 , 0 [*]	Jump to the 1st... 10th document tab .
Alt Tab [†]	Switch to the workspace application to the right .
Shift Alt Tab [†]	Switch to the workspace application to the left .
Ctrl Alt → [†]	Switch to the workspace to the right .
Ctrl Alt ← [†]	Switch to the workspace to the left .

Notes:

^{*} Keyboard digits only; does not work with Numpad digits.

[†] GTK+ shortcut added for completeness; has no relation to gedit.

Selecting

Table 2: Selecting Shift

shortcut	action
Shift Ctrl →	Select forward by words .
Shift Ctrl ←	Select backwards by words .
Shift Ctrl PageUp	Select to the beginning of the line .
Shift Ctrl PageDn	Select to the end of the line .
Shift Ctrl ↑	Select forward by lines .
Shift Ctrl ↓	Select backwards by lines .

shortcut	action
Shift Ctrl Home	Select to the beginning of the document .
Shift Ctrl End	Select to the end of the document .
Ctrl A	Select all .

For selecting other custom text that *can* be specified by a **regular expression**, use the *Click Config* plugin. Regular expressions can match many items, but evidently **not everything** as **regex recursion** remains unavailable.

Moving

Table 3: Moving Alt

shortcut	action
Alt →	Move the selected word forward by one word .
Alt ←	Move the selected word backwards by one word .
Alt ↑	Move the selected line up by one line .
Alt ↓	Move the selected line down by one line .

Deleting

Table 4: Deleting

shortcut	action
Ctrl BkSpce	Delete the word behind the cursor.
Ctrl Del	Delete the word ahead of the cursor.
Shift Ctrl BkSpce	Delete from the cursor to the beginning of the line .
Shift Ctrl Del	Delete from the cursor to the end of the line .
Ctrl D	Delete the current line .

Editing

Table 5: Editing

shortcut	action
Ctrl X	Cut the selected text or region and place it on the clipboard.
Ctrl C	Copy the selected text or region onto the clipboard.

shortcut	action
Ctrl V	Paste the contents of the clipboard.
Ctrl Z	Undo the last action.
Shift Ctrl Z	Redo the last undone action.
Shift F7	Check spelling .

Special characters

On GNU/Linux machines, special characters can be entered by their UTF code using the key combination **Shift Ctrl U**. Finish off with **Enter** or **Space**. [UTF-8 code](#) for some of the most common special characters is [listed here](#).

Searching

Table 6: Searching

shortcut	action
Ctrl F	Find a string.
Ctrl G	Find the next instance of the searched string.
Shift Ctrl G	Find the previous instance of the searched string.
Shift Ctrl K	Clear the search result highlighting.
Ctrl H	Search and replace .

Document operations

Table 7: Document operations

shortcut	action
Ctrl N	Create a new document.
Ctrl O	Open a document.
Ctrl Alt O	Quickly open a document.*
Ctrl S	Save the current document.
Shift Ctrl S	Save as a new file.
Shift Ctrl L	Save all open documents.
Ctrl P	Print the current document.
Shift Ctrl P	Print preview .

shortcut	action
Ctrl W	Close the current document.
Shift Ctrl W	Close all open documents.

Note:

* Requires the *Quick Open* plugin.

Application

Table 8: Application

shortcut	action
F1	Open the gedit user guide .
F9	Show/hide the side pane .
F11	View fullscreen .
Ctrl Q	Quit gedit.

Olvido colour scheme

I use gedit with my own dark colour scheme called *Olvido*. It is almost indistinguishable from the *Oblivion* colour scheme, if not for the LaTeX expressions which are handled better. Reminder: “*El olvido*” is Spanish for “*Oblivion*”.

Compared to other editors

Gedit 3.10 and its commercial counterpart *Sublime Text 3* can be made to act pretty much the same with the aid of plugins or packages. Nonetheless, both applications continue to suffer from a number of annoyances and glaring omissions. After extensively testing both applications in a Markdown content scenario, my personal balance eventually tipped in favour of *Vim*.

Table 9: gedit, Sublime Text 3 and Vim compared

action	gedit	Sublime Text 3	Vim
Comment code	Ctrl M	Ctrl / or 7 depending on keyboard layout. Adds an undesired space.	<code>\cc</code> using NERD Commenter , <code>,cc</code> on non-English keyboards!
Uncomment code	Shift Ctrl M	Shift Ctrl / , i.e. not possible on my Swiss keyboard!	<code>\uc</code> using NERD Commenter , <code>,uc</code> on non-English keyboards!
Select between brackets & parenthesis	No	Shift Ctrl M	<code>vi[</code> and <code>vi(</code>
Select current word	No	Ctrl D	<code>iw</code>
Select current line	No	Ctrl L	<code>V</code>
Move (drag) current line	Alt ↑ / ↓	Shift Ctrl ↑ / ↓	Alt ↑ / ↓ after configuring ~/.vimrc
Delete current line	Ctrl D	Shift Ctrl K	<code>dd</code>
Duplicate current line	No	Shift Ctrl D	<code>yyp</code> for yank line and put
Insert at beginning of current line	Begin	Begin	<code>I</code> for insert
Append at end of current line	End	End	<code>A</code> for append
Move (drag) text	By word with Alt ← / → However, it performs poorly and is therefore hardly useful.	No, only by character with MoveText and therefore hardly useful!	By word with Alt ← / → , by character with Ctrl ← / → after configuring ~/.vimrc
Spell check of Markdown with suggestions	Yes	No; fails between brackets!	Yes; <code>z=</code> for suggestions
Spell check language persistence by file	Yes	No	No, all configured dictionaries are used simultaneously.
Open URL	No	Ctrl Alt ← with Clickable URLs	<code>gx</code> , but it does not work with Markdown .
Enter character by UTF code	Shift Ctrl U	No	Shift Ctrl U
Word suggestions	As you type.	Tab	Tab with SuperTab



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-nc-sa/4.0/).

Other licensing available on request.

Unattended CSS typesetting with . The Prince logo features the word 'Prince' in a stylized purple font with 'Print with CSS' written in smaller text above it.

This work is published at <https://hamwaves.com/gedit/en/>.

Last update: Monday, March 1, 2021.