↖ Topic Index

Emacs support for editing Changelog files 🚧

Description	Keystroke	Function	Note
Changelog Files Last updated on:	Emacs natively supports the **change-log-mode** for editing **Changelog files**. PEL enhance Emacs' support when **pel-use-changelog** user-option is set, providing some extra key bindings and providing access to this PDF. 2025-05-07	See **Emacs Change Log Commands**	This mode is related to VCS and CVS version control systems. 🚧 The documentation here does not include information about those and might have to be updated. ⚠️
Open this PDF file. See also: **Help/Info**	**<f12> <f1>**	(**pel-changelog-help** &optional OPEN-WEB-PAGE)	Open the **Changelog File** local PDF. If the prefix argument (like **C-u** or **M--**) is used, then it opens the remote GitHub hosted raw PDF instead. If the **pel-flip-help-pdf-arg** user-option is set it's the other way around.
ChangeLog Control			
Use change-log-mode	M-x change-log-mode	(**change-log-mode**)	Major mode for editing change logs; like Indented Text mode. • Prevents numeric backups and sets 'left-margin' to 8 and 'fill-column' to 74. • New log entries are usually made with **M-x add-change-log-entry** or **C-x 4 a**. • Each entry behaves as a paragraph, and the entries for one day as a page.

Add new entry in the change-log	**C-x 4 a**	(**add-change-log-entry** &optional WHOAMI CHANGELOG-FILE-NAME OTHER-WINDOW NEW-ENTRY PUT-NEW-ENTRY-ON-NEW-LINE)	Find ChangeLog buffer, add an entry for today and an item for this file.
			• With an optional prefix WHOAMI (such as **C-u**) prompt for user name and email (the default values are stored in '**add-log-full-name**' and '**add-log-mailing-address**' user-options).
			• The CHANGELOG-FILE-NAME is the file name of the change log.
			  • Interactively, it's nil; so the function uses the value of '**change-log-default-name**' user-option.
			  • If the file thus named exists, it is used for the new entry. If it doesn't exist, it is created, unless '**add-log-dont-create-changelog-file**' is t, in which case a suitably named buffer that doesn't visit any file is used for keeping entries pertaining to CHANGELOG-FILE-NAME's directory.
			• Third arg OTHER-WINDOW non-nil means visit in other window. Interactively it's nil and therefore uses the current window.
			• Fourth arg NEW-ENTRY non-nil means always create a new entry at the front; never append to an existing entry. Option '**add-log-keep-changes-together**' otherwise affects whether a new entry is created. Interactively it's nil so it creates a new entry at the front (the top) of the file.
			• Fifth arg PUT-NEW-ENTRY-ON-NEW-LINE non-nil means that if a new entry is created, put it on a new line by itself, do not put it after a comma on an existing line.
			• Option '**add-log-always-start-new-record**' non-nil means always create a new record, even when the last record was made on the same date and by the same person.
			• The change log file can start with a copyright notice and a copying permission notice. The first blank line indicates the end of these notices.
			• Today's date is calculated according to '**add-log-time-zone-rule**' if non-nil, otherwise in local time.

Go to source location of change log tag	**C-c C-c**	(**change-log-goto-source**)	Go to source location of "change log tag" near 'point'. • A change log tag is a symbol within a parenthesized, comma-separated list. • If no suitable tag can be found nearby, try to visit the file for the change under 'point' instead.
Visit changed file	**C-c C-f**	(**change-log-find-file**)	Visit the file for the change under point.
Move to next note	**C-c C-n**	(**add-log-edit-next-comment** ARG)	Cycle forward through Log-Edit mode comment history. • With a numeric prefix ARG, go forward ARG comments.
Move to previous note	**C-c C-p**	(**add-log-edit-prev-comment** ARG)	Cycle backward through Log-Edit mode comment history. • With a numeric prefix ARG, go back ARG comments.