↖ Topic Index

Compilation Mode

Action	Key binding	Command	Note
Compilation-Mode ○ Help • Compile • Navigate in compilation log buffer • next-error-follow-minor-mode • Navigate to next/previous error • kill compilation, quit buffer	Emacs provides a special mode to display the result of commands, such as compilers, that list errors and warnings detected on specific lines and columns: the compilation-mode. • Some major modes provide another mode derived from the compilation-mode that is specialized for the programming language supported by the mode. • For example, emacs-lisp-mode uses the emacs-lisp-compilation-mode. • These derived modes normally provide the same commands and key bindings. Any difference is described in the page for the mode. In all cases you can always request information for the mode by typing `C-h m` or `<f1> m` in the window using the compilation-mode or its derived counterpart.		
Last updated on:	2025-05-01	See also: ⅀ **Hide/Show**	
Open this PDF file (⅀ Compilation Mode) See also: ⅀ **Help/Info**	• `<f11> ? p` • `<f11> p`	(**pel-help-pdf-select** &optional OPEN-WEB-PAGE)	Prompt for a PEL PDF and open it. Prompt supports tab completion. For this PDF type `compilation <RET>` at the prompt. • 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.
To Compile	Several major modes have their own command to launch the compilation. • The compilation commands are described in the page describing the major mode. • PEL tries to bind the command for compilation to the `<f12> c` key sequence. In some cases that is a prefix to more commands related to compilation.		
Compile	`M-x compile`		Refer to the major mode for the compilation command specific to the mode.
compilation-mode buffer commands	The following commands are issued from the compilation-mode log buffer that opens when the compilation command detected errors and or warnings.		
Move to beginning of buffer See also: ⅀ **Navigation**	• `<` • `M-<` • `C-<home>`	(**beginning-of-buffer** &optional ARG)	Move point to the beginning of the buffer. • With numeric arg N, put point N/10 of the way from the beginning. • If the buffer is narrowed, this command uses the beginning of the accessible part of the buffer. • Push mark at previous position, unless either a `C-u` prefix is supplied, or Transient Mark mode is enabled and the mark is active.
Move to end of buffer See also: ⅀ **Navigation**	`>`	(**end-of-buffer** &optional ARG)	Move point to the end of the buffer. • With numeric arg N, put point N/10 of the way from the end. • If the buffer is narrowed, this command uses the end of the accessible part of the buffer. • Push mark at previous position, unless either a `C-u` prefix is supplied, or Transient Mark mode is enabled and the mark is active.
Toggle next-error-follow-minor-mode When this mode is active, navigating across errors in the compilation log buffer causes automatic display of the faulty line in the source code window.	`C-c C-f`	(**next-error-follow-minor-mode** &optional ARG)	Toggles next-error-follow-minor-mode for compilation, occur and diff modes. • If the prefix argument is positive, enable the mode, and if it is zero or negative, disable the mode. • When turned on, cursor motion in the compilation, grep, occur or diff buffer causes automatic display of the corresponding source code location. To check whether the minor mode is enabled in the current buffer, type the following key sequence: `M-: next-error-follow-minor-mode <RET>`
Display the source of current error inside another window	`C-o`	(**compilation-display-error**)	Display the source for current error in another window. • Useful if the compilation-mode log buffer uses the complete Emacs frame.
Visit source file for current error message	`C-c C-c`	(**compile-goto-error** &optional EVENT)	Visit the source code for the current error message: move point inside the source code file at the line and column identified by the message.
Next error no select (Emacs >= 28.1)	`n`	(**next-error-no-select** &optional N)	Move point to the next error in the 'next-error' buffer and highlight match. • Prefix arg N says how many error messages to move forwards (or backwards, if negative). Finds and highlights the source line like `` C-x ` ``, but does not select the source buffer.
Compilation next error	`M-n`	(**compilation-next-error** N &optional DIFFERENT-FILE PT)	Move point to the next error in the compilation buffer. • Prefix arg N says how many error messages to move forwards (or backwards, if negative). This does NOT find the source line like `` C-x ` ``.
Next Error and visit file	• `` C-x ` `` • `M-g n` • `M-g M-n`	(**next-error** &optional ARG RESET)	Move to the next error message and visit the corresponding source code. • If all the error messages parsed so far have been processed already, the message buffer is checked for new ones. • A prefix ARG specifies how many error messages to move; negative means move back to previous error messages. • Just `C-u` as a prefix means reparse the error message buffer and start at the first error.
Previous error no select (Emacs >= 28.1)	`p`	(**previous-error-no-select** &optional N)	Move point to the previous error in the 'next-error' buffer and highlight match. • Prefix arg N says how many error messages to move forwards (or backwards, if negative). Finds and highlights the source line like `M-g p`, but does not select the source buffer.
Compilation previous error	`M-p`	(**compilation-previous-error** N)	Move point to the previous error in the compilation buffer. • Prefix arg N says how many error messages to move backwards (or forwards, if negative). This does NOT find the source line like `M-g p`.
Previous Error and visit file	• `M-g p` • `M-g M-p`	(**previous-error** &optional N)	Visit previous previous error message and corresponding source code. • Prefix arg N says how many error messages to move backwards (or forwards, if negative).
Compilation error on previous file	`M-{`	(**compilation-previous-file** N)	Move point to the previous error for a different file than the current one. Prefix arg N says how many files to move backwards (or forwards, if negative).
Compilation error on next file	`M-}`	(**compilation-next-file** N)	Move point to the next error for a different file than the current one. Prefix arg N says how many files to move forwards (or backwards, if negative).
Quit window	• `q` • `C-x w q`	(**quit-window** &optional KILL WINDOW)	Quit WINDOW and bury its buffer. • With prefix argument KILL non-nil, kill the buffer instead of burying it.
Kill compilation	`C-c C-k`	(**kill-compilation**)	Kill the compilation process.