File Management

	File Management			
Operation	<u>Keystroke</u>	Function	<u>Note</u>	
File Management See also: • ∑ Dired • ∑ Customize	content, displaying dire The directory editing There are also severa Use Emacs custor PEL <f11> f key p Once you have mo</f11>	arge set of commands to open files (Emacs documentation uses the term "finding" files for that), saving files searching for file directory content, etc These are listed in this table. Iting (dired) commands are mainly listed in the Dired table. It were all Emacs internal and external packages that provide useful commands. PEL supports several of them, listed below. It is stomize system to modify their values to activate, deactivate and modify the behaviour of these packages. It is prefix followed by either 12 to access PEL activation group and 13 to access the external package customization groups. It is modified the relevant user-option values, apply or save them and then either execute M-x pel-init or restart Emacs. It is modified the following Emacs built-in libraries or functionalities:		
• <u>》Key-Chords</u>	 Library ffap			
Open this PDF file. See also: <u>∑ Help/Info</u>	<f11> f <f1> 1</f1></f11>	(pel-help-pdf &optional OPEN-WEB-PAGE)	Open the <u>S File-mngt</u> 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.	
Customize PEL File/ Directory Management	<f11> f <f2> 1</f2></f11>	(pel-customize-pel &optional OTHER-WINDOW)	Customize PEL support for file management. • If OTHER-WINDOW is non-nil (use C-u), display in other window.	
Customize Emacs file management support	<f11> f <f3></f3></f11>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs support for file management. Includes the following: files, recentf, popup-switched	
Customize Emacs support for file revert	<f11> f r <f3></f3></f11>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs support for file automatic revert management.	
Customize ffap (find file at point)	<f11> f a <f3></f3></f11>	(pel-customize-library &optional OTHER-WINDOW)	Customize Emacs support for management of ffap (find file at point).	
Open File in OS application	The following command	d opens file(s) outside Emacs, ι	using OS applications registered with the file type.	
Open currently file visited in current buffer with the default OS application. See: <u>Dired</u> , <u>Web</u>	<f11> f F</f11>	(pel-open-buffer-file-in-os- app &optional <u>FNAME</u>)	Open the file in the present buffer with the OS-registered application. If the buffer is modified, prompt to save buffer first. In dired-mode buffers, open each marked files in its S-registered applications. Inside a dired-mode buffer you can also type z to open the current file or all selected files.	
Opening file	The following commands are available to open/visit files in Emacs buffers. • For some of them the corresponding ido mode function is also shown. • The command used to 'visit' a file, find-file is Emacs default. It supports Emacs' basic tab completion. Packages that support other completion mechanisms can be installed and activated and then the command uses a different completion mechanism. • See the Completion mechanism dynamically. You can change it without restarting Emacs or event re-executing pel-init. • See the Completion/Input and Coustomize tables for more info.			
File Lock			the same file with a lock. If you attempt to edit the buffer of a locked file, or save a buffer of a locked lock (with 's'), 2) proceed ('p') to edit the file anyway or 3) quit ('q').	
Open file-open dialog Open (visit) a file/ directory See also:	 <f11> f f</f11> <m-f11> M-f</m-f11> M-f With PEL, the <f11> f and <m-f11> M-f M-f key bindings are always av</m-f11></f11> 		on macOS in graphics mode only: open a file, select the file name via an OS File dialog. Prompt for the file or directory name to open. Open the selected file/directory in a buffer with the appropriate mode. For directory, the buffer opens in Dired-mode. • With PEL, the <f11> f f and <m-f11> M-f M-f key bindings are always available, regardless of what completion mechanism is in use. It can be used as a fallback when testing</m-f11></f11>	
Description Section S	C-x C-f		various completion packages. I have seen some of them fail and break Ido. Note that <m-f12> M-f M-f is also available in some major modes to open files in a way that takes the major mode into account, like providing a list of files in the project. Refer to major mode pages for specific information.</m-f12>	
		(ido-find-file)	Same as above with Ido completion See Completion/Input for completion modes available at the prompt. The ido-use-filename-at-point user-options control whether ido-find-file uses the file name at point as the basis for selecting the file name to open. PEL provides the <f11> f M key sequence to dynamically change the value. See below.</f11>	
• Prevent Ido expansion with c-j	 find-file is the original command and uses Emacs default completion. When Ido is used, the ido-find-file command is used instead. When <u>ido</u> mode is used, you can also: Type C-f or C-x f to change to original find-file mode and prevent Ido completion from trying to provide the name of an existing file when you want to specify the name of a file that does not exists yet. Type C-j to accept the file/directory name verbatim without replacement or suggestion. Also useful to open a directory in dired mode. To open a file in read-only mode you can: Use one of the commands below (C-x C-r, etc) Use C-x C-f then type C-x C-q to change the mode of the buffer to read-only mode. 			
Open in read-only:	PEL supports dyna	mic selection of completion inp	out that control the way this command operates to help you select a file name: (ido, ivy, helm).	
Open file via popup menu	<f11> f M-f</f11>	(pel-psw-navigate-files)	Open file from a pop-up menu listing files in current directory. Uses (psw-navigate-files "."). • Narrow menu list by typing part of the file name. You can also select directory names. • Requires popup-switcher PEL activates when pel-use-popup-switcher is t.	
Open another file in ouffer	C-x C-v	(find-alternate-file FILENAME & optional WILDCARDS) (ido-find-alternate-file)	Kills buffer and open the newly specified file in a new buffer same window. When ido-mode is used, the ido-find-alternate-file is used instead. Useful when just selected an empty file just selected by mistake.	
Open file in other window	• C-x 4 f • <f11> f o</f11>	(find-file-other-window FILENAME &optional WILDCARDS)	Edit file FILENAME, in another window. • Like C-x C-f, but creates a new window or reuses an existing one.	
Open file in other frame	C-x 5 f	(ido-find-file-other- window) (find-file-other-frame FILENAME &optional WILDCARDS)	Edit file FILENAME, in another frame. • Like C-x C-f, but creates a new frame or reuses an existing one.	
		(ido-find-file-other-frame)		

(ido-find-file-other-frame)

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>	
** Set whether ido- find-file uses filename at point	<f11> f M</f11>	(pel-set-ido-use-fname-at- point &optional GLOBALLY)	Set Ido's ability to use the filename at point as a starting point in the current buffer or globally. It can set it to one of: • disabled: don't use filename at point. • guess: try to identify an exiting file name from the name at point. • literal: use name at point in the Ido search for a file name.	
	 By default this commands sets ido-find-file behaviour for the current buffer only by setting a ido-use-filename-at-point buffer local variable. Use any prefix argument (eg. C-u) to modify the behaviour globally for the current Emacs session, it does not persist across Emacs sessions. For a persistent behaviour change you must customize ido-use-filename-at-point user-option variable. For that, use M-x customize-option. This affects the behaviour of all commands opening file using Ido completion: ido-find-file as the others. 			
Open in read-only	The following command	ds open files in read-only mode	. While in read-only mode, use Use C-x C-q to permit editing.	
Open a file in read-only mode	С-ж С-г	(find-file-read-only FILENAME &optional WILDCARDS) (ido-find-file-read-only)	Edit file FILENAME but don't allow changes. Like C-x C-f , but marks buffer as read-only. Use C-x C-q to permit editing.	
Open file in other window in read-only mode	• C-x 4 r • <f11> f 0</f11>	(find-file-read-only-other-window FILENAME & optional WILDCARDS) (ido-find-file-read-only-other-window)	(find-file-read-only-other-window FILENAME &optional WILDCARDS) Edit file FILENAME in another window but don't allow changes. Like C-x C-f, but marks buffer as read-only. Use C-x C-q to permit editing.	
Open as root	On Unix/Linux/macOS	some files are write protected a	and can only be opened with root privilege with su or sudo. Use the following command for those.	
Open file with root privilege	<f11> f R</f11>	(pel-edit-as-root &optional ARG)	Open a file as root with sudo. • If already visiting a file and a prefix ARG is specified then edit currently visited file as root. • The function uses tramp and will prompt for password if necessary.	
Open Literally	Note that when using	g Ido completion, it is possible	d in the Fundamental mode: the major mode normally associated with the file type is not used. to use a command during completion to force Ido to open the file literally. However, if you are using the only way to open a file literally.	
Visit a file literally: with no encoding support and conversion	<f11> f M-1</f11>	(find-file-literally FILENAME)	Visit file FILENAME with no conversion of any kind.	
	The major mode usedAutomatic uncompre	d is Fundamental mode regardlession and adding a newline at	are both disabled, and multibyte characters are disabled in the resulting buffer. less of the file name, and local variable specifications in the file are ignored. the end of the file due to 'require-final-newline' is also disabled. e, this command asks you whether to visit it literally instead.	
Open binary	Open a file in hex binary	y mode. There are also comma	ands to convert current buffer to hexadecimal editing, like nhexl (described in <u>Natural Buffers</u>).	
Open a file in hexl-mode See also: <u>Nauffers</u>	<f11> f M-x</f11>	(hexl-find-file FILENAME)	Edit file FILENAME as a binary file in hex dump format, using the 'hexl-mode'. • Switch to a buffer visiting file FILENAME, creating one if none exists.	
Recently opened			PEL ensures that Emacs remembers the list of recently opened files and provides:	
S Completion/Input	 the pel-initial-recent-f-function user-option identifies which function use used to open the recently opened files: ido-recentf-open uses the current Ido prompt or Ido enhanced mechanism. Use <f11> M-c ? to list them and see which one is active.</f11> counsel-recentf uses a vertical list prompt. Requires counsel psw-switch-recentf: uses a popup menu The menu bar includes a File->Open Recent menu entry. Some other functions are activated by their respective user options. 			
Open recently opened files, using active method	<f11> f M-r M-r</f11>	(pel-find-recent-file)	Open the recent file prompt using the currently active function. The function is selected by pel-initial-recent-f-function . It can be modified in the current editing session by pel-select-recentf-function, bound to <f11> f M-r M-R. When basic Ido is used, type <tab> to get possible expansions listed in a separate buffer. Ido completion is selectable. Use <f11> M-c ? to list them and see which one is active. When counsel-recentf is used, you can type C-c C-o to copy the list of files inside a special buffer.</f11></tab></f11>	
Display the name of the function used to prompt for recently opened file	<f11> f M-r M-?</f11>	(pel-show-recentf-function &optional AFTER- SELECTION-P)	Display what function is used to visit recently opened files. • The argument is for internal use, it is not available interactively.	
Select the function used to list/prompt the recently opened files.	<f11> f M-r M-R</f11>	(pel-select-recentf- function & optional RECENTF-FUNCTION SILENT)	Select the function to visit recently opened files. Modifies what is used in the current editing session, not the persistent value selected by the pel-initial-recent-f-function user-option. • The arguments are for internal use, they are not available interactively.	
Edit list of recently opened files	<f11> f M-r M-e</f11>	(recentf-edit-list)	Show a dialog to delete selected files from the recent list. • Use this to remove some of the files from the list.	
Open file at point	The following commands, followed by the flap commands, allow opening files from the file name taken at point (the cursor location). They work regardless of the input completion method currently used. Note that when using the Ido completion mode, it is possible to instruct Ido to use a file name at point as the basis for the file name to open. This Ido behaviour is controlled by the ido-use-filename-at-point user-option. With PEL you can control it globally or locally with <f11> f M</f11>			
Open filename at point in a browser See also: •	• <f11> f / • 6u</f11>	(pel-browse-filename-at- point)	Open the file name at point inside the system's browser. • If point is at a directory name, open the systems application that browses directories (eg. macOS Finder, Windows Explorer). • This is the same as using pel-open-at-point with the argument N set to 9. It is easier to type and PEL assigns its own key-chord for it.	
Open URL at point in a browser See also: • <u>Sey-Chords</u> , <u>Sey-Chords</u>	• <f11> f M-/ • 7u</f11>	(browse-url-at-point &optional ARG)	Ask a WWW browser to load the URL at or before point. • Variable 'browse-url-browser-function' says which browser to use. • With prefix argument inverts the value of the option 'browse-url-new-window-flag'. • Use <f11> <f2> E u to open the browse-url group that contains relevant user options.</f2></f11>	
Copy URL at point in temporary file and visit the file	<f11> f M-u</f11>	(pel-open-url-at-point)	Copy the URL at point to a local temporary file and visit that file. • A The download copy of the file does not have the same name and may not open with the proper mode because it won't have an extension. The HTML formatted files will be recognized by Emacs but most of the files won't be. • Save the file somewhere else using the C-x C-w key sequence and identify the proper extension to activate the required major mode.	
With goto-address-mode	C-c C-f		This binding is only available when point is over the URL and the <u>goto-address-mode</u> minor mode is active. Use <f11> f u or <f11> f U to activate this mode.</f11></f11>	
Set base directory for pel-open-at-point relative file names	<f11> f ;</f11>	(pel-set-open-at-point-dir)	Set the behaviour of 'pel-open-at-point' in current buffer. • Select how it determines the directory from which a relative file name is built. Prompts to allow selection of one of the following methods: • Use visited file parent directory (the default). • Use buffer's current working directory. • Use a specified directory. Prompts for the directory name. Supports completion.	

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>	
Open file or web-page whose name is at point	• C-^ • <f11> f . • <m-f11> M-f</m-f11></f11>	(pel-open-at-point &optional N)	Open the file, library or the URL, named at point, with potential line & column #s. • Will find source files in current project as specified by pel-filename-at-point-finders user-option. Supports glob characters, partial directory path. When multiple files are found it prompts using	
**	М		the method selected by pel-prompt-read-method user-option.	
See also:	• <u>6y</u>		The <u>6v</u> key-chord is available if pel-use-key-chord is non-nil.	
• <u>∑ Key-Chords</u> • <u>M reStructuredText</u>				
• <u>Bī - C</u>	 This command works generically but is also specialized for some major modes, like C, C++, Erlang, reStructuredText. See their respective pages for the mode specific features. 			
• <u>ֆմ - C++</u> • ֆմ - Erlang	When executed from	• When executed from with a buffer in sh-mode, the '=' and ':' characters are used as additional delimiters. Shell variables (such as \$HOME) are		
y. Lining	In general the com- directory. If the file identified by the p	 In general the command extracts the file (or directory) name (and possibly line and column numbers) from text at point and tries to open the file or directory. If the file is not found then the command searches the file inside a directory tree holding the current file. The root of that directory tree is identified by the presence of a project marker file, one of the file identified in the pel-project-root-identifiers user option. Something like .git, .hg, .project or .pel-project by default. 		
	If embedded space(s	are allowed in the filename, the	text at point. The name is either surrounded by white space characters or the delimiters listed below. hen point must be located at the first of the 2 delimiter characters. by by the point tab, newline and: "`' ()[]{ $<$ ''" [] () $<$ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\ \\	
	If embedded space	e in the file name is not allowe	d, then the file name must also be enclosed in the above delimiters, the space acts as an extra	
	If the string identifies		e page in the systems' default browser.	
Change relative path base directory selection			lob characters (even though this is not used in a #include "" or #include <> statements). ectory or the buffer's current working directory by default.	
method 🖛		•	y executing the pel-set-open-at-point-dir command (<f11> f ;</f11>) in the buffer. en-file-at-point-dir user-option. Use <f11> f <f2></f2></f11> to open the customization buffer to modify it.	
	Otherwise the comm	and attempts to open the file r	name with the specified name. If that file does not exists it then proceed to search for it. nbers the point is moved to that position.	
Select prompt method ■		•	•	
			I lists them and prompts using the method selected by pel-prompt-read-method user-option. Ited by PEL. You can select a more powerful ivy prompting instead.	
			pel-use-ivy to t pand <u>Ivy mode</u> will be installed automatically when you restart Emacs.	
	This allows you	to detect potential duplication	the specified search method, it does not only use the first one found. in header file names in large include paths.	
Select target window	' '		ng the find file (with completion), search for libraries files (type 1) according to current file type.	
-	Select target windo Without argument:	:		
	If no window ho	olds that file, select the target v	int to that window and to the line column coordinates if specified following the file name at point. window based on the number of editable windows in frame: if 1, split that window and use the new	
N>20 : open the directory	With numeric argu		re, use the current window.	
directory *	• (abs N) > 20: th		of the file. Interpret the window position from the N value adjusted: N-20 (or N+20 if N is negative)	
		other' (the next) window. ove (excluding 8, 9 and 10): se	elect the target window based on the number of editable windows in frame:	
	if 1 window: if 2 windows.		nd use the new window, w,	
	if 3 or more windows: use the current window. N is: 8: up, 2: down, 4:left, 5:current, 6:right.			
See function docstring	N is 9: open the file in the system's browser (with N=29 or N=-29, open the directory of the file name in the system's browser), open a directory name at point with directory browsing (eq. macOS Finder, Windows Explorer).			
for more info.	 N is 10: open the URL at point in the system's browser. Selecting Minibuffer, inexistent or dedicated window is not allowed. 			
ffap commands	Emacs provides the ffap (find file at point) command set. The ffap command is similar to pel-find-file-at-point-in-window but does not support line and numbers, does not support identifying a window with command arguments and is not designed to support multiple programming languages. It does however support other facilities and can be installed to replace the behaviour of standard file management command bindings such as C-x C-f.			
	PEL activates the E	macs built-in ffap library when	the pel-use-ffap user option is set to either t or to ffap-bindings . In both cases these activate the	
		is set to ffap-bindings, then PE	EL also activates the standard ffap bindings which take over the behaviour of the main file finding and	
	dired commands. This means that Ido, Ivy or Helm are no longer available for these commands. If pel-use-ffap is only set to t then the standard ffap bindings is not activated.			
Find file/URL at point	<f11> f a p</f11>	(ffap &optional FILENAME)	Find FILENAME, guessing a default from text around point. • If 'ffap-url-regexp' is not nil, the FILENAME may also be an URL. Web URL opens in browser.	
			 With a prefix, this command behaves exactly like 'ffap-file-finder'. If 'ffap-require-prefix' is set, the prefix meaning is reversed. 	
			See also the variables 'ffap-dired-wildcards', 'ffap-newfile-prompt', 'ffap-url-unwrap-local', 'ffap-url-unwrap-remote', and the functions ffap-file-at-point' and 'ffap-url-at-point'.	
Find file/URL at point - read only	<f11> f a P</f11>	(ffap-read-only)	Like 'ffap', but mark buffer as read-only.	
Find another file/URL at point in window Find file/URL in other	<f11> f a v</f11>	(ffap-atternate-file) (ffap-other-window)	Like 'ffap' and 'find-alternate-file': kills current buffer and open new file in the same window. Like 'ffap', but put buffer in another window.	
window Find file/URL in other	<f11> f a w</f11>	(ffap-other-frame)	Like 'ffap', but put buffer in another frame.	
frame Find file/URL in other	<f11> f a f</f11>	(ffap-read-only-other-	Like 'ffap', but put buffer in another frame. Like 'ffap', but put buffer in another window and mark as read-only.	
window - read only Find file/URL in other	<f11> f a W</f11>	(πap-read-only-other- window) (ffap-read-only-other-	Like 'ffap', but put buffer in another frame and mark as read-only. Like 'ffap', but put buffer in another frame and mark as read-only.	
frame - read only Start Dired with file at		frame) (dired-at-point &optional	Start Dired, defaulting to file at point. See 'ffap'.	
point Start Dired with file at	<f11> f a d</f11>	FILENAME)		
Start Dired with file at point in other window Start Dired with file at	<f11> f a D</f11>	(ffap-dired-other-window)	Like 'dired-at-point', but put buffer in another window.	
point in other frame	<f11> f a M-d</f11>	(ffap-dired-other-frame)	Like 'dired-at-point', but put buffer in another frame.	
List directory of file at point	<f11> f a l</f11>	(ffap-list-directory)	Like 'dired-at-point' and 'list-directory'.	
Open a menu of all files, URL in current buffer.	<f11> f a m</f11>	(ffap-menu &optional RESCAN)	Put up a menu of files and URLs mentioned in this buffer. Set mark, jump to choice, and try to fetch it. The menu is cached in 'ffap-menu-alist', and rebuilt by 'ffap-menu-rescan'. • With prefix argument: forces a rebuild. Searches with 'ffap-menu-regexp'.	

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>
Open <u>Dired</u> (Directory Editor)	When "opening" (visiting) a directory Emacs opens a buffer in Dired mode, that looks like a ls -l output, which allows several operations. If you specify a directory path to Cx C-f then Dired-mode is used. You can also use the following commands to open buffer in Dired mode. It's also possible to browse a file directory tree with file tree browsers, like NeoTree and Ztree, described below in this table. The Sependbar can also be used.		can also use the following commands to open buffer in Dired mode.
Open a directory editor See also: <u>> Dired</u> > Completion/Input	• C-x d • %-D	(dired DIRNAME &optional SWITCHES) (ido-dired)	Opens a Dired-mode buffer on the specified directory. Prompt for the directory name. PEL activates ido when the pel-use-ido-mode user option is set to t. See Completion/Input for completion modes available at the prompt.
Run Dired in other (next) window	C-x 4 d	(dired-other-window)	Opens a Dired-mode buffer on the specified directory inside another window. Prompt for the directory name.
List Directory See also: Completion/Input	C-x C-d	(list-directory DIRNAME &optional VERBOSE)	Display a list of files in or matching DIRNAME, a la 'ls'. • DIRNAME is globbed by the shell if necessary. • Prefix arg (C-u) means supply -l switch to 'ls'.
Activating URLs to browse and open files	Emacs provides the goto-url-mode and the goto-url-prog-mode that turn URLs found in the current buffer into clickable buttons. • Once the mode is active the following key sequences are available wheel point is over a URL button: • C-c RET or the mouse to click on the button. • If the URL is an email address a buffer to write an email to that address opens. • If the URL is a web or FTP address the system browser is invoked to open the address. • C-c C-n : move point to the end of the next URL in the buffer. • C-c C-p : move point to to the previous URL in the buffer. • C-c C-f : download the file identified by the URL into a local temporary file and visit the file. See (pel-open-url-at-point) above.		
Toggle goto-address- mode	<f11> f u</f11>	(goto-address-mode &optional ARG)	Minor mode to buttonize URLs and e-mail addresses in the current buffer. With a prefix argument ARG, enable the mode if ARG is positive, and disable it otherwise.
Toggle goto-addrress- prog-mode	<f11> f U</f11>	(goto-address-prog-mode &optional ARG)	Like 'goto-address-mode', but only for comments and strings.
Open the URL (email or web page)	C-c RET	(goto-address-at-point &optional EVENT)	Open the URL at point: If URL is a web page: open it in a browser If URL is a mail address: Send mail to address at point: Find e-mail address around or before point. Then search backwards to beginning of line for the start of an e-mail address. If no email address is found there, then load the URL at or before point.
Move to end of next URL in buffer See also: <u>Navigation</u>	C-c C-n <f6> C-n</f6>	(pel-goto-next-url)	Move point forward to the end of the next URL located in the current buffer. • The global <f6> C-n key binding activates the goto-address-mode if it is not already active.</f6>
Move to beginning of previous URL in buffer See also: <u>Navigation</u>	C-c C-p <f11> C-p</f11>	(pel-goto-previous-url)	Move point backward to the beginning of the previous URL located in the current buffer. • The global <f6> C-p key binding activates the goto-address-mode if it is not already active.</f6>
Insert text of another file at point	The following commands can be used to insert text from other files at point in the current buffer.		
Insert file at point	• C-x i • <f11> f i</f11>	• (insert-file FILENAME) • (ido-insert-file)	Insert contents of file FILENAME into buffer after point. • Set mark after the inserted text.
Insert file literally at point	<f11> f I</f11>	(insert-file-literally FILENAME)	Insert contents of file FILENAME into buffer after point with no conversion. Set mark after the inserted text.
Write text into specified file	The following command	ls can be used to write text se	lected from current buffer into specified file.
Write region text to file	<f11> f w</f11>	(write-region START END FILENAME &optional APPEND VISIT LOCKNAME MUSTBENEW)	Write current region into specified file. • Prompts for the specified file.
Append region text to file	<f11> f W</f11>	(append-to-file START END FILENAME)	Append the contents of the region to the end of file FILENAME. • Prompts for the specified file.
Set file mode	<f11> f m</f11>	(set-file-modes FILENAME MODE)	Set mode bits of file named FILENAME to MODE (an integer). Only the 12 low bits of MODE are used. Prompts for file name and then for chmod-like file mode value.
Reverting Files	If the file's content changed on the disk and you want to refresh the Emacs buffer visiting that file, you need to "revert" the file. If you want to use Emacs to monitor the content of a file that is continuously modified by an external process (like a log file) set the <i>revert-without-query</i> variable to a list of regular expressions describing the field it'll apply to. You can also activate the auto-revert mode for the current buffer or globally and restart its timer.		
Revert a buffer See also: Diff & Merge	• <f11> f r f • ₩-u</f11>	(revert-buffer & optional IGNORE-AUTO NOCONFIRM PRESERVE-MODES)	Replace current buffer text with the text of the visited file on disk. • This undoes all changes since the file was visited or saved. • With a prefix argument, offer to revert from latest auto-save file, if that is more recent than the visited file. • This is also the command to use to reload a file that was modified on the file system. • You can use ediff-current-file to see the difference between the buffer and its disk file. PEL binding for this is <f11> e b f.</f11>
Toggle auto-revert mode	<f11> f r a</f11>	(auto-revert-mode &optional ARG)	Toggle reverting buffer when the file changes (Auto-Revert Mode). With a prefix argument ARG, enable Auto-Revert Mode if ARG is positive, and disable it otherwise. • Auto-Revert Mode is a minor mode that affects only the current buffer. When enabled, it reverts the buffer when the file on disk changes. • When a buffer is reverted, a message is generated. This can be suppressed by setting 'auto-revert-verbose' to nil.
Toggle auto-revert tail mode	• <f11> t • <f11> f r t</f11></f11>	(auto-revert-tail-mode &optional ARG)	Toggle reverting tail of buffer when the file grows. • With a prefix argument ARG, enable Auto-Revert Tail Mode if ARG is positive, and disable it
See also: ∑ Scrolling			 otherwise. When Auto-Revert Tail Mode is enabled, the tail of the file is constantly followed, as with the shell command 'tail -f'. This means that whenever the file grows on disk (presumably because some background process is appending to it from time to time), this is reflected in the current buffer. You can edit the buffer and turn this mode off and on again as you please. But make sure the background process has stopped writing before you save the file!
Cancel/restart auto- revert timer	<f11> f r SPC</f11>	(pel-auto-revert-set-timer)	Restart or cancel the timer used by Auto-Revert Mode. If such a timer is active, cancel it.
	Restarting the timer e : pel-auto-revert-se	ensures that Auto-Revert Mode t-timer is a thin wrapper over a	ve or if Auto-Revert Mode is active in some buffer. will use an up-to-date value of 'auto-revert-interval' (which is normally 5 seconds by default). uto-revert-set-timer that displays a warning if executed when the buffer is not already in auto-revert- when auto-revert-set-timer is executed.

<u>Operation</u>	<u>Keystroke</u>	Function		<u>Note</u>	
Saving Files	PEL supports the foll	nands to save the content of owing controllable actions or ed with a command for the fi	file save. Each of these actions are activ	vated via an action-specific PEL user-optic heir associated user-option and command	on, and can I are listed here:
	Action Delete trailing space override it for som Update time stamp o	and lines on save pel-del e major modes: pel-m n save pel-upo	te-trailing-whitespace pel-tog des-preventing-delete-trailing-whitespac ate-time-stamp pel-tog	ggle-update-time-stamp-on-save	Key Sequence <f11> M-W <f11> M-T</f11></f11>
Save file to disk	 Update copyright not C-x C-s 	(save-buffer &optional ARG	1, 5	ggle-update-copyright-on-save By default, it makes the previous version is	<f11> M-C</f11>
Safe ine to disk	• %-s	(save-same) deplication of the	previously requested or if this is the fire With C-u: marks this version to bec With C-u C-u: makes the previous With C-u C-u C-u: marks this version into a bec with prefix 0: never make the previous on macOS in graphics mode only:	rst save. come a backup when the next save is done s version into a backup file rersion to become a backup when the next ackup file. bus version into a backup file.	e save is done, and
Save all/some files	С-х s	(save-some-buffers &optional ARG PRED)	Prompt for files that are modified. Opt y : save n : don't save C-r : look at the buffer in question d : view differences with diff-buff		
Write buffer to specified file Save As	C-x C-w	(write-file FILENAME &optional CONFIRM) (ido-write-file)	Similar to "Save-As": prompt for the fill Can also be yanked in the mini buffe Use that command to rename the fi	er, use M-n to edit it.	
Changed current buffer changed state	M-~	(not-modified &optional ARG)	Mark current buffer as unmodified, not • With C - u prefix ARG, mark buffer as	•	
Toggle copyright update on save	<f11> M-@</f11>	(pel-toggle-update- copyright-on-save &optional GLOBALLY)	current Emacs editing session (the c To modify the global state permaner user option via the 'pel-pkg-for-filem'		ssions). el-update-copyright'
Toggle timestamp update on save	<f11> M-T</f11>	(pel-toggle-update-time- stamp-on-save &optional GLOBALLY)	 Toggle time-stamp update on file save and display current state. By default change behaviour for local buffer only. When GLOBALLY argument is non-nil, using any prefix argument, change it for all buffers for the current Emacs editing session (the change does not persist across Emacs sessions). To modify the global state permanently modify the customized value of the 'pel-update-time-stamp' user option via the 'pel-pkg-for-filemng' group customize buffer with <f11> f <f2> 1.</f2></f11> This command is only available when the pel-update-time-stamp is set to t. 		
Toggle delete trailing space on save See also: Whitespace	• <f11> M-W • <f11> t w M-W</f11></f11>	(pel-toggle-delete-trailing- space-on-save &optional GLOBALLY)	current Emacs editing session (the case Trailing whitespace deletion is autowhitespace user-option is set to t. Use To modify the global state permaner		ssions). pel-delete-trailing- ivate it. el-delete-trailing-
Inserting & Automatically Updating Copyrights	Emacs has built-in support for insertion and update of copyright notices inside files. • Two commands, shown below, are provided to manually insert or update the file's copyright notice. • The copyright notice can be automatically updated by adding the copyright-update function to the list of before-save-hook variable with the following code: (add-hook 'before-save-hook 'copyright-update) To be automatically updated, the copyright notice must be placed within an area at the beginning of the file specified by the value of the copyright-limit variable, normally defined as the first 2000 characters. This variable is customizable.				
Insert copyright notice at point See also: <u>∑ Inserting</u> Text	<f11> i C</f11>	(copyright &optional STR ARG)	Insert a copyright by \$ORGANIZATION	N notice at cursor. variable is not available, Emacs prompts f	or it.
Update file's copyright notice	M-x copyright- update	(copyright-update &optional ARG INTERACTIVEP)		e current year. in the notice rather than adding the current ppl-version' is set, any copying permissions	•
	 Even when used interactively copyright-update does not warn if there is no copyright in the current buffer to update. It does not create a missing notice. If you want to be prompted automatically to update an existing but out-of-date copyright notice, write the following inside your init.el file: (add-hook 'before-save-hook 'copyright-update) 				

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>		
Automatic File			es. It must be activated by adding the time-stamp function to the before-save-hook variable.		
Time Stamp on file save	The time stamp will be added to files that contain, inside their first 8 lines, a line that looks like one of the following: Time-stamp: <>				
References:	• Time-stamp: " "				
<u>TimeStamps @</u> <u>EmacsWiki</u>	You can, however change these defaults and get Emacs to update all sorts of time stamp formats, even inside source code statements: Emacs controls automatic insertion of timestamp with the following variables:				
Change time stamp format in:	• time-stamp-pattern consists of 4 parts, each one controlled by a variable:				
 markdown file reStucturedText file 	 time-stamp-line-limit: identifies where in the file the time stamp can be located. Defaults to 8: the first 8 lines. time-stamp-start: identifies the text pattern that precedes the time stamp. time-stamp-formed according the format to the stamp. 				
See also: Inserting	• Som	 time-stamp-format specifies the format of the time stamp. Something like "%:y-%02m-%02d %02H:%02M:%02S %u" to specify the date and time in ISO format, with the user login's name. 			
Text	 time-stamp-time-zone specifies the time zone selection: nil: Emacs local time 				
	• t: Unive • wall : sy	rsal time vstem wall clock time			
		ntrolled by a TZ environment va p-format and time-stamp-tim	ariable e-zone variables can be set in your init file or via the Emacs customization system.		
	n *	d in the time-stamp customiza	tion group. ling or after the automatically updated time stamp, it is best to use file local variables: this will allow		
			various formats. As an example, see the top and end of the <u>PEL manual raw format file</u> .		
			within the first 8 lines of the file, otherwise it will not be updated automatically. set the time-stamp-line-limit file local variable.		
	PEL provides the ex	tra user-option to control the a	utomatic generation of time-stamps:		
		The state of the s	ether time-stamps are automatically update time stamps in all files where a valid time-stamp pove. Set it to t (the default) to allow automatic time stamp updates. Set it to nil to prevent them. You		
			session by using the <f11> f M-t key sequence. kage provides a set of text insert commands which include inserting a time stamp.</f11>		
		ng Text table for the appropria			
Update file time stamp	<f11> f t</f11>	(time-stamp)	Force update the time stamp string(s) in the current buffer. • Updates a time stamp of format recognized by <i>Emacs current settings</i> even when automatic time-		
			stamp update is off. • More information about the "Emacs current settings" in the description block above.		
Toggle time stamp	<f11> f M-t</f11>	(time-stamp-toggle-active	Toggle 'time-stamp-active', setting whether <f11> f t updates a buffer.</f11>		
automatic update		&optional ARG)	With ARG, turn time stamping on if and only if arg is positive.		
RFC-Mode		_	mode commands. Requires <u>rfc-mode</u> activated by <u>pel-use-rfc-mode</u> ,		
Read a specific RFC Browse RFCs	<f11> B r <f11> B R</f11></f11>	(rfc-mode-read NUMBER)	Read the RFC document NUMBER. Offer the number at point as default. Browse through all RFC documents referenced in the index.		
		(rfc-mode-browse)			
Browsers	 Emacs supports mechanisms to browse file directories. This includes: Emacs built-in ∑ Dired directory editor, along with several extensions. You can have several different Dired buffers in an Emacs session. The Emacs built-in ∑ Speedbar and its extensions. There can only be one instance of a Speedbar buffer and that can be inside another frame. Several other external packages: Neotree, treemacs and Ztree 				
View Directory	The NeoTree extern	al package provides a Vim-Nei	rdTree like tree-view of a directory with expansion/collapse.		
Tree with NeoTree		en pel-use-neotree is set to t . 2> opens the PEL customizat	ion group to set pel-use-neatree		
	 <f11> B N <f2> opens the PEL customization group to set pel-use-neotree.</f2></f11> <f11> B N <f3> prompts, select neotree to open the neotree customization group.</f3></f11> There is only one NeoTree window. It is a dedicated window. 				
	lcons used in the tre	e can be changed:	o arrows to use arrows instead of '+'.		
	In graphics mod	de, if pel-neotree-font-in-graphi	ics is set to icons then the icons provided by all-the-icons package is used.		
		The state of the s	ackage it does not install the fonts. executing: M-x all-the-icons-install-fonts		
View directory tree with	<f11> B N N</f11>	(neotree-toggle)	Toggle show/hide the NeoTree window.		
<u>NeoTree</u>		In the NeoTree buffer the follo			
	 n next line, p previous line. > end of buffer, < top buffer 				
		 SPC or RET or TAB : Open U Go up a directory 	current item if it is a file, Fold/Unfold current item if it is a directory.		
		 g Refresh A Maximize/Minimize the N	leoTree Window		
		• H Toggle display hidden file	es. Controlled by neo-hidden-regexp-list user option.		
		 O Recursively open a direct C-c C-n Create a file or c 	tory breate a directory if filename ends with a '/'		
		 C-c C-d Delete a file or a C-c C-r Rename a file or 	•		
		• C-c C-c Change the root • C-c C-p Copy a file or a	directory.		
Open NeoTree for dir of current buffer	<f11> B N F</f11>	(neotree-find &optional PATH DEFAULT-PATH)	Open a NeoTree window using the directory of the current buffer. No prompt.		
Open NeoTree for specified directory	<f11> B N D</f11>	(neotree-dir PATH)	Prompt for a directory. Open a Neotree window for that directory.		
Close NeoTree window	<f11> B N H</f11>	(neotree-hide)	Close the NeoTree window.		
Show NeoTree window	<f11> B N S</f11>	(neotree-show)	Show the NeoTree window.		
<u>Treemacs</u>			pace/project oriented tree-based view with expansion/collapse and actions of directories and files.		
Manipulate directory			cs user-option is turned on (set to t).		
trees associated as projects/workspaces	Treemacs has a large number of user-options in the treemacs customization group and sub-groups. PEL <f11> B <f3> key sequence gives access to the customization group.</f3></f11>				
 Manipulate the directories and files 		the treemacs buffer with the <			
**	 In graphics mode the mouse provides access to most commands. In terminal (and graphics) mode when pain is inside the treemacs dedicated window, the treemacs major mode key-bindings, listed below, are available. 				
See: ∑X Treemacs	The treemacs-mode an	d extensions have an extensive	e command set. See ∑X Treemacs for the complete list		
Open/close treemacs	<f11> B T</f11>	(treemacs)	Initialise or toggle treemacs. See ∑x Treemacs for treemacs-mode commands.		
	_	·	If the treemacs window is visible hide it.		
			If a treemacs buffer exists, but is not visible show it. If no treemacs buffer exists for the current frame create and show it. If the street and show it. If the first analysis and the street and show it.		
			If the workspace is empty additionally ask for the root path of the first project to add.		

Operation	<u>Keystroke</u>	Function	<u>Note</u>			
View Directory	The ztree external package provides a text-based tree-view of a directory with expansion/collapse.					
Tree with ZTree	PEL ztree customization:					
	• <f11> B <f2> opens the PEL customization group (select the tree subgroup) . See also: Customize.</f2></f11>					
		es it when pel-use-ztree is set				
		following PEL provided custon				
	 pel-ztree-dir-n pel-ztree-dir-f 		ve focus to new entry when <ret> is typed. egexp to ignore more file. Do not enter quote for string.</ret>			
	per-zu ce-un-i		b ignore the .pyc files, enter ^.*pyc on a line.			
	•		olay filtered files until H is typed. Normally they are not shown until H is typed.			
			the ztree customization group itself.			
	1. Execute M-x pel	-init after settling and apply	ing new values to activate the new values.			
View directory as tree	<f11> B Z</f11>	(ztree-dir PATH)	Open an interactive buffer with the directory tree of the PATH given.			
with ztree-dir			 ▶ Opens the tree buffer in the current window. ▶ There can be several buffers with different ztree-dir trees. 			
		In the Ztree Dir buffer the follo				
		> : narrow/display director	· ·			
		• d : Open Dired at point.	, ,			
			d files. Controlled by regexp in the ztree-dir-filter-list user option.			
		• x : Toggle expand/collapse				
			n large directory trees it takes a long time. I have see Emacs hang when typing ${f x}$ again during that			
		time. Minvestigate.				
Searching/Finding		ds can be used to search for fil				
Files	A	#6 : searching and finding files.				
See also:	Use man to get moon locate: <f1< li=""></f1<>					
• <u>∑ Help/Info</u>	• on find: <f1< td=""><td></td><td></td></f1<>					
• <u>National Direct</u>	You can manipulate	e the result in Dired with Dired o	commands. For instance type (to toggle the display of more than the file names.			
Search for file with	<f11> f L</f11>	(locate SEARCH-STRING	Prompt for a search pattern and search for filenames using the system locate command line utility			
locate		&optional FILTER ARG)	through the sell to search a database of all pathnames that match the specified search pattern. The			
			database is recomputed periodically.			
			The search result is shown in a '*Locate*' buffer.With prefix arg ARG, prompt for the exact shell command to run instead. This way you can specify			
			options to the locate command line utility.			
Run grep via find	• <f11> f g</f11>	(find-grep COMMAND-	Run grep via find, with user-specified args COMMAND-ARGS.			
	• <f11> g f</f11>	ARGS)	Collect output in a buffer.			
See also: <u>See also</u>			• While find runs asynchronously, you can use the C-x ` command to find the text that grep hits refer to.			
			This command uses a special history list for its arguments, so you can easily repeat a find			
			command.			
Search for files with	<f11> f d</f11>	(find-dired DIR ARGS)	Prompts for the root to search from, and a find command to search for files with the Unix find.			
'find' and open Dired			Specify the arguments for the <u>find command</u> . For example to perform a case inspecifive search for all higher user in area. (**) high			
buffer			 For example, to perform a case insensitive search for all .h files, use: -iname "*\.h" Opens a Dired-mode buffer and show the files found in there. 			
Search directory for	<f11> f n</f11>	(find-name-dired DIR	Search DIR recursively for files matching the globbing pattern PATTERN, and run Dired on those files.			
files and open Dired		PATTERN)	PATTERN is a shell wildcard (not an Emacs regexp) and need not be quoted.			
buffer for those			The default command run (after changing into DIR) is:			
			findname 'PATTERN' -ls			
Find files in a directory	<f11> f h</f11>	(find-grep-dired DIR	Find files in DIR that contain matches for REGEXP and start Dired on output.			
and open Dired output		REGEXP)	The command run (after changing into DIR) is:			
			find.\(-typef-exec'grep-program'find-grep-options'-eREGEXP{}\;\)-ls			
			where the first string in the value of the variable 'find-ls-option' specifies what to use in place of "-ls"			
			as the final argument.			
Find Emacs Lisp files in	<f11> f 1</f11>	(find-lisp-find-dired DIR	Find Emacs Lisp files in DIR, matching REGEXP.			
directory tree		REGEXP)	Open *Find Lisp Dired* buffer on output.			

File Management — References

Topic & Link	Description
Emacs Display - Mode Line	Read first. Describes what the Emacs mode line displays.
GNU Emacs Manual - File Handling	Describes how to open and deal with files and directories in Emacs.
GNU EMACS Manual - Interactive Do	Describes the ido-mode, a nice addition that helps with completing file names at prompts.
Display path of file in status bar	In graphics mode, display the buffer name and the full path file in parenthesis inside the frame title bar.
How do I rename an open file in Emacs?	
Find files faster with the recent files package	Mickey Petersen article describing the recent file feature. PEL ido-recentf-open is taken from Mickey Peterson code.