File Management

		1 110	Management
<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>
File Handling See also: Dired Dired Tramp (edit remote computer files) Features: open file open file at point ffap commands Recently opened open modes (read-only, root, binary) Fuzzy file finder Dired (directory editor) Activate URLs in buffer Insert text from file Write text to file Reverting file Save to file, Rename Inserting copyright Automatic time stamp RFC mode Directory tree browsers Mode-specialized open	content, displaying dire There are also severa Use Emacs custor PEL <f11> f key p Once you have mo PEL provides integratio archive-rpm Library ffap Library recentf The fzf.el externa Automatic file time is</f11>	ctory content, etc These are all Emacs internal and external prize system to modify their validation of the content of the	rpm, provides ability to open RPM and CPIO archive files as you can do with tarball and zip files. wide several commands to open file at point.
Open this PDF file. See also: <u>Nelp/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 open remote GitHub hosted raw PDF instead. If pel-flip-help-pdf-arg user-option is set it's the other way around.
<u>S</u> 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, fzf, recentf, popupswitcher, x509 (see M X.509 Certificates).
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).
Show file mngt status	<f11> f ?</f11>	(pel-show-filemng-status)	Display status of various file management controls: encoding, resolving relative path method, etc
Open File in OS application	The following command opens file(s) outside Emacs, using OS applications registered with the file type. See: Dired, Web		
Open currently file visited in current buffer with the default OS application.	<f11> f C-o</f11>	(pel-open-buffer-file-in-os- app &optional <u>FNAME</u>)	Open the file in the current buffer with the OS-registered application. If the current buffer holds a HTML file, that's a quick way to open the file in your browser. 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 See ∑ Completion/Input.	The following commands are available to open/visit files in Emacs buffers. For some of them the corresponding <u>ido</u> 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. PEL customization system allows you to specify whether you want to use one or several other completion mechanisms. It also has a command to change the completion mechanism dynamically. You can change it without restarting Emacs or event re-executing pel-init. See the <u>Completion/Input</u> and <u>Customize</u> tables for more info. File Lock: Emacs protects against multiple processes modifying the same file with a lock. If you attempt to edit the buffer of a locked file, or save a buffer		
Open file-open dialog	of a locked file, Emacs	(ns-open-file-using-panel)	steal the lock (with 's'), 2) proceed ('p') to edit the file anyway or 3) quit ('q'). 6 On macOS in graphics mode only: open a file, select the file name via an OS File dialog.
Open (visit) a file/ directory See also:	• <f11> f f • M-<f11> M-f M-f • C-x C-f</f11></f11>	(find-file FILENAME &optional WILDCARDS)	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 various completion packages. I have seen some of them fail and break Ido. I 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. See major mode pages. Same as above with Ido completion. See <u>S Completion/Input</u> for available completion modes.</f12></f11></f11>
C-x C-f /sudo::/ path/to/file .			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. Use <f11> f M to dynamically change it.</f11>
C-x C-f /su::/ path/to/file with C-f Change input completion method	 find-file is the original command and uses Emacs default completion. When Ido is used, the ido-find-file command is used instead. When ido 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. Use <f11> M-c <f4> to select another input completion method. See Completion/Input.</f4></f11> 		
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 buffer	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) (ido-find-file-other- window)	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	(find-file-other-frame FILENAME & optional WILDCARDS) (ido-find-file-other-frame)	Edit file FILENAME, in another frame. • Like C-x C-f, but creates a new frame or reuses an existing one.
	L	·	4

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>		
Open same file in other directory	• <f11> f M-d • M-<f11> M-f</f11></f11>	(pel-open-file-in-other-dir)	Open file of same name as current one present in another directory. Subsett to create clone of current file, or to compare files located inside 2 different VCS dir trees.		
	 First prompt with the name of the directory of currently visited file using the default completion mechanism ('ido' by default). Use the prompt to select the name of the other directory (which must already exist). Use C-f to edit the dir path without completion. Select dir name, hit <ret> to open the same file in the selected other directory.</ret> 				
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.		
See also: <u>See Completion/</u> Input	Use any prefix arguments persistent behavio	 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 file at point	Note that when usin	g the Ido completion mode, it i	e taken at point (the cursor location). They work regardless of the current input completion method. is possible to instruct Ido to use a file name at point as the basis for the file name to open. This Ido oint user-option. With PEL you can control it globally or locally with <f11> f M</f11>		
Set base directory for pel-open-at-point	<f11> f ;</f11>	(pel-set-open-at-point-dir)	Set the behaviour of 'pel-open-at-point' in current buffer. Which defaults to value selected by pel-open-file-at-point-dir user-option.		
relative file names		Use visited file parent dieUse buffer's current work			
Open file or web-page whose name is at point	• M- <f6> • <f11> f .</f11></f6>	(pel-open-at-point &optional N)	Open the file, library or the URL, named at point, with potential line & column #s. • If necessary will search source code files in current project as specified by pel-filename-at-point-		
** Command is generic and	• <u>6y</u>		finders user-option. Type <f12> <f4> ? to show used file search method in supporting modes. Supports glob characters, partial directory path. When multiple files are found it prompts using</f4></f12>		
is also specialized for: • <u>M reStructuredText</u>	M- <f6> overriden in iedit-mode</f6>		the method selected by pel-prompt-read-method user-option. The <u>6y</u> key-chord is available if pel-use-key-chord is non-nil. See <u>x Key-Chords</u> .		
• <u>\$\text{\$\titt{\$\text{\$\titt{\$\text{\$\}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}}</u>	~	s generically but is also specially but it is also specially but it also specially but it also specially but it is also specially but	alized for some major modes, like C, C++, Erlang, reStructuredText, shell scripts.		
 \$\mathbb{P}\tilde{\pi} - \text{Erlang}\$ \$\mathbb{P}\tilde{\pi} - \text{UNIX Shell}\$ 			e '=' and ':' characters are used as additional delimiters. Shell variables (such as \$HOME) are		
Delimiting characters ■	The generic mode ex newline and: "`' ()	traction works by identifying the contraction works by identification	name, and possibly line and column numbers, from text at point and tries to open the file or directory. The beginning & end of the file/directory/library/URL name string by delimiter characters, one of: tab, (()) (()) (()) (()) (()) (()) (()) ((
File identification heuristic <f11> f <f2> <f11> f ; <f11> f ;</f11></f11></f2></f11>	The name may include The command uses a U In the file/dir name is it builds a absolute particular.	 If embedded space(s) are allowed in the name, point must be located at the first of the 2 delimiter chars. Otherwise point can be anywhere in the name. The name may include glob characters (but not in C/C++ in #include "" or #include <> statements). The command uses a URL unchanged but uses the following heuristic to identify the exact location of the file/directory: In the file/dir name is an absolute path it uses that. Otherwise it builds a absolute path using the extracted relative path name inside the directory identified by the pel-open-file-at-point-dir user-option, which can be 1) use parent directory of currently visited file, or use current working directory, 2) use current working directory, or 3) use user-specified directory. It uses 			
Select multi-file selection method ▼	the found file/dir name if it exists. Otherwise it searches for the relative file/dir name in directory tree under the root marker file identified by the pel-project-root-identifiers user-option which is something like .git, .hg, .project, .pel-project (the default). If it can find such a file in the above directories it searches the tree under the found root. If it finds several files it prompts using the current completion mode to allow selection of the appropriate name (see below) and opens the selected one. If it finds only one it opens that file. Otherwise, it prompts showing the name searched and provide the following choices: 1) create the file with specified name, 2) edit the name to search again, 3) use the name found and search for an Emacs library file with that name, or 4) quit. The command opens the extracted name according to this heuristic: If the string is a properly formatted URL, it opens it using the OS default browser (even if a optional numeric argument specified otherwise), otherwise if the string is a file or directory name it opens it. If the file name is followed by line and column numbers the point is moved to that position in the buffer. When finding several file names, the command lists them and prompts using the method selected by pel-prompt-read-method user-option. The default is a very primitive function implemented by PEL. You can select a more powerful ivy prompting instead. With ivy selected, PEL will automatically set pel-use-ivy to pel-use-ivy to pel-use-ivy to totoe not only use the first one found.				
N>20 : open the directory	 Use this to detect potential duplication in header file names in large include paths. The command opens the file in the window selected by the following logic controlled by presence or absence of typed numerical prefix arguments: Select target window: Without argument: If file or directory is already opened in a window, move point to that window and to the line column coordinates if specified. If no window holds that file, select the target window according to the number of editable windows in frame: if 1, split that window and use the new window, if 2: use the other window, if 3 or more, use the current window. With prefix numeric argument N: N < 0: create a new window and use that. (abs N) > 20: then open the directory instead of the file. Interpret the window position from the N value adjusted: N-20 (or N+20 if N is negative) N = 0: use the 'other' (the next) window. N = 1, 3, 7or above (excluding 8, 9 and 10): select the target window based on the number of editable windows in frame: 				
	 if 1 window: split that window and use the new window, if 2 windows: use the other window, if 3 or more windows: use the current window. N is: 8: up, 2: down, 4:left, 5:current, 6:right. 8 := up				
See function docstring for more info.	 N is 9: force opening the file in the OS associated application (eg. macOS Finder, Windows Explorer). If this is a URL, open it in the OS default web browser. 				
Open filename at point	Selecting Minibuffe <f11> f /</f11>	er, inexistent or dedicated wind (pel-browse-filename-at-	dow is not allowed. Open the file name (or URL) at point inside the system's web browser.		
in a browser See also: <u>\(\sigma\) Key-Chords</u> , <u>\(\sigma\) Web</u>	• <u>6u</u>	point)	If point is at dir name, open the OS app. browsing dirs (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: • \(\subseteq \text{Key-Chords} \), • \(\subseteq \text{Web} \)	• <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. • In 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		s This binding is only available when point is over the URL and the goto-address-mode minor mode is active. Use <f11> f u or <f11> f U to activate this mode.</f11></f11>		

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>
ffap commands	numbers, does not suphowever support other PEL activates the Erkey bindings shown bel	port identifying a window with of facilities and can be installed to macs built-in ffap library when ow.	et. The ffap command is similar to pel-find-file-at-point-in-window but does not support line and command arguments and is not designed to support multiple programming languages. It does be replace the behaviour of standard file management command bindings such as C-x C-f. The pel-use-ffap user option is set to either t or to ffap-bindings. In both cases these activate the Lalso activates the standard ffap bindings which take over the behaviour of the main file finding and
	dired commands. Th		are no longer available for these commands.
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	<f11> f a v</f11>	(ffap-alternate-file)	Like 'ffap' and 'find-alternate-file': kills current buffer and open new file in the same window.
Find file/URL in other window	<f11> f a w</f11>	(ffap-other-window)	Like 'ffap', but put buffer in another window.
Find file/URL in other frame	<f11> f a f</f11>	(ffap-other-frame)	Like 'ffap', but put buffer in another frame.
Find file/URL in other window - read only	<f11> f a W</f11>	(ffap-read-only-other- window)	Like 'ffap', but put buffer in another window and mark as read-only.
Find file/URL in other frame - read only	<f11> f a F</f11>	(ffap-read-only-other-frame)	Like 'ffap', but put buffer in another frame and mark as read-only.
Start Dired with file at point	<f11> f a d</f11>	(dired-at-point &optional FILENAME)	Start Dired, defaulting to file at point. See 'ffap'.
Start Dired with file at point in other window	<f11> f a D</f11>	(ffap-dired-other-window)	Like 'dired-at-point', but put buffer in another window.
Start Dired with file at 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'.
Recently opened Solution Solution Solution Solution Solution Solution Solution Recently opened		:-f-function user-option identificing uses the current Ido promp	PEL ensures that Emacs remembers the list of recently opened files and provides: lies which function use used to open the recently opened files: t or Ido enhanced mechanism. Use <f11> M-c ? to list them and see which one is active.</f11>
		: uses a vertical list prompt. f: uses a popup menu es a File->Open Recent menu	Requires counsel external package described by pel-use-counsel entry. Some other functions are activated by their respective user options.
Open recently opened files, using active	<f11> f M-r M-r</f11>	(pel-find-recent-file)	Open the recent file prompt using the currently active function.
method		When basic Ido is used, typ Ido completion is selecta	pel-initial-recent-f-function. Change with pel-select-recentf-function, bound to <f11> f M-r M-R. De <tab> to get possible expansions listed in a separate buffer. Able. Use <f11> M-c ? to list them and see which one is active. ed, 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 a recently opened file searched by fzf	<f11> f M-r M-z</f11>	(fzf-recentf &optional WITH-PREVIEW)	Open a recently opened file selected by fzf search. With C-u show file preview. See (fzf) below. Page 1 Requires the fzf.el external package 2 activated by pel-use-fzf.
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 s		und can only be opened with root privilege with su or sudo. Use the following command for those. -x C-f /sudo::/path/to/file, as su with: C-x C-f /su::/path/to/file
Open file with root privilege	<f11> f R</f11>	(pel-edit-as-root &optional ARG)	Open a file as root with sudo. Prompt for password if necessary. If already visiting a file and a prefix ARG is specified then edit currently visited file as root.
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	<f11> f M-1</f11>	(find-file-literally FILENAME)	Visit file FILENAME with no conversion of any kind.
and conversion	The major mode usedAutomatic uncompre	d is Fundamental mode regardl ssion and adding a newline at t	are both disabled, and multibyte characters are disabled in the resulting buffer. ess 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		ands to convert current buffer to hexadecimal editing, like nhexl (described in <u>E Buffers</u>).
Open a file in hexl-mode See also: <u>∑</u> Buffers	<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.

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>
Fuzzy File Finders	install and configure the fzf.el ext	the fzf command line utility, ternal package	der that can be used within Emacs via the fzf.el emacs front-end. To use it inside Emacs, you must: and use one of the following package to use the corresponding commands: tivated by pel-use-fzf . The fzf commands below are available when this is active. tivated by pel-use-counsel . The counsel commands below are available when this is active.
Open file searched by fzf in current directory	• <f11> M-z M-z • <f11> f z</f11></f11>	(fzf &optional <u>WITH-</u> <u>PREVIEW</u>)	Open a file selected by fzf session in the current directory. Type partial file name, use fzf filter expressions. Select one file and hit return to open it inside current window. • Process current working directory or Projectile process root directory if available.
fzf & fzf-directory support fzf file preview	For fzf and fzf-directory: With optional prefix (eg. C-u) the currently selected file content or attribute is shown using the preview command identified by the fifther file content with cat, but that can be customized to use other mechanisms.		
Open file searched by fzf in specified directory	• <f11> M-z M-d • <f11> f d</f11></f11>	(fzf-directory &optional WITH-PREVIEW)	Prompt for a directory to perform the fzf file search, then open selected file inside current window. Directory prompt uses current completion mode. See <u>Sempletion/Input</u> .
Open fzf searched file in current or specified directory using ivy I/F	<f11> f c</f11>	(counsel-fzf &optional INITIAL-INPUT INITIAL- DIRECTORY FZF-PROMPT)	Open a file selected by ivy-style prompt using a fzf shell command. • With C-u prefix argument first prompts for the directory to perform the fzf search. • Much slower than (fzf) for large directories because counsel captures fzf output before showing it.
Switch buffer with fzf See also: <u>∑</u> Buffers	<f11> b z</f11>	(fzf-switch-buffer)	Switch buffer in current window by selecting it with fzf. • Uses the fzf command line utility for fast & flexible search. Requires the fzf.el external package activated by pel-use-fzf.
Search/open Git repo member files with fzf	<f11> f g</f11>	(fzf-git-files)	Search files committed current Git repository with fzf and open user selected file.
Search/open committed file in Git repo directory tree with fzf	<f11> f G</f11>	(fzf-git)	Search all files in current Git repository with fzf and open user selected file.
Search/open committed file in Mercurial repo tree with fzf	<f11> f h</f11>	(fzf-hg-files)	Search files committed current Mercurial repository with fzf and open user selected file.
Search/open file in Mercurial repo directory tree with fzf	<f11> f H</f11>	(fzf-hg)	Search all files in current Mercurial repository with fzf and open user selected file.
Search/open file in current projectile project with fzf. See XX Projectile	<f11> f <f8></f8></f11>	(fzf-projectile &optional WITH-PREVIEW)	Search all files in current projectile project with fzf and open selected file. With C-u show file preview. Projectile external package Activated by pel-use-fzf Requires the projectile external package Activated by pel-use-projectile
Grep search files with fzf for specified regex	<f11> g s</f11>	(fzf-grep)	Prompt for string to search and file grep selection expression, show grep results in a fzf session, select appropriate line to open the specific file at appropriate line.
Grep search files with fzf for specified regex in specified directory	<f11> g S</f11>	(fzf-grep-in-dir)	Prompt for directory, then for string to search and file grep selection expression, show grep results in a fzf session, select appropriate line to open the specific file at appropriate line.
Grep search Git repo member files with fzf for specified regex	<f11> g G</f11>	(fzf-git-grep)	Prompt for string to search and file grep selection expression, show grep results over current Git repo searched in a fzf session, select appropriate line to open the specific file at appropriate line. Limit This command does not seem to work properly, it searches but does not always open the file.
Open <u>Dired</u> (Directory Editor) See also: <u>Dired</u>	When "opening" (visiting) a directory Emacs opens a buffer in Dired mode, that looks like a ls -I output, which allows several operations. If you specify a directory path to C-x C-f then Dired-mode is used. You can also use the following commands to open buffer in Dired mode. • Prompt input completion can be changed for these. See <u>Completion/Input</u> It's also possible to browse a file directory tree with file tree browsers , like NeoTree and ztree (see below), or with <u>Speedbar</u> .		
Open a directory editor	• 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 .
Run Dired in other 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	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 -I switch to 'ls'.
Jump to file entry in dired buffer ★★ 《Leaves point on the file jumped to, allowing immediate Dired action, eg.: C-x C-j R renames the file.	С-х С-ј	(dired-jump &optional OTHER-WINDOW FILE- NAME)	Jump to Dired buffer corresponding to current buffer. If in a file, Dired the current directory and move to file's line. If in Dired already, pop up a level and goto old directory's line. In case the proper Dired file line cannot be found, refresh the dired buffer and try again. When OTHER-WINDOW is non-nil, jump to Dired buffer in other window. When FILE-NAME is non-nil, jump to its line in Dired. Interactively with prefix argument, read FILE-NAME.
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-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-address- 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, around point or before.
Move to end of next URL in buffer See also: Navigation	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	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	_	ls can be used to insert text fro	om 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.
	-		

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>	
Write text into specified file	The following command	ds 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. Solve 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 See also: ∑ Scrolling	• <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 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	Restarting the timer e pel-auto-revert-se	ensures that Auto-Revert Mode t-timer is a thin wrapper over a	Restart or cancel the timer used by Auto-Revert Mode. If such a timer is active, cancel it. 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.	
Saving Files To rename a file use one of: C-x C-j R C-x C-w	Use the following commands to save the content of a buffer to a filesystem file. • PEL supports the following controllable actions on file save. Each of these actions are activated via an action-specific PEL user-option, and can temporarily be disabled with a command for the file in the current buffer. The actions and their associated user-option and command are listed here: Action • Delete trailing space and lines on save • override it for some major modes: • Update time stamp on save • pel-modes-preventing-delete-trailing-whitespace • pel-toggle-update-time-stamp pel-toggle-update-time-stamp-on-save • \$\forall \text{F11> M-W}\$			
Save file to disk	Update copyright not C-x C-s %-s		save current buffer to associated file. By default, it makes the previous version into a backup file if previously requested or if this is the first save. • With C-u: marks this version to become a backup when the next save is done • With C-u C-u: makes the previous version into a backup file • With C-u C-u: marks this version to become a backup when the next save is done, and makes the previous version into a backup file. • With prefix 0: never make the previous version into a backup file. • With prefix 0: never make the previous version into a backup file. • On macOS in graphics mode only: %-s brings a OS file-save dialog. A save and activated on-file-save actions only occur when the buffer is in "changed" status. Use M to flip that status to force an action when it has just been activated.	
Save all/some files	C-x s	(save-some-buffers &optional ARG PRED)	Prompt for files that are modified. Options: • y : save • n : don't save • C-r : look at the buffer in question • d : view differences with diff-buffer-with-file	
Write buffer to specified file Save As/Rename	C-x C-w	(write-file FILENAME &optional CONFIRM) (ido-write-file)	Similar to "Save-As": prompt for the filename. • Can also be yanked in the mini buffer, use M-n to edit it. © Use that command to rename the file.	
Changed current buffer changed state	M-~	(not-modified &optional ARG)	Mark current buffer as unmodified, not needing to be saved. • With C - u prefix ARG, mark buffer as modified, so C - x C - s will save.	
Toggle copyright update on save	<f11> M-@</f11>	(pel-toggle-update- copyright-on-save &optional GLOBALLY)	Toggle copyright 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-copyright' user option via the 'pel-pkg-for-filemng' group customize buffer with <f11> f <f2> 1. ☑ This command is only available when the pel-update-copyright is set to t.</f2></f11>	
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. ☑ This command is only available when the pel-update-time-stamp is set to t.</f2></f11>	
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)	Toggle deletion of trailing spaces 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). Trailing whitespace deletion is automatically activated on file save when the pel-delete-trailing-whitespace user-option is set to t. Use this command to de-activate it or re-activate it. • To modify the global state permanently modify the customized value of the 'pel-delete-trailing-whitespace' user option via the 'pel-pkg-for-filemng' group customize buffer with <f11> f <f2> 1.</f2></f11>	

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>
Inserting & Automatically Updating Copyrights	The copyright notice code: (add-hook) To be automatically	can be automatically updated 'before-save-hook 'copy	must be placed within an area at the beginning of the file specified by the value of the copyright-limit
Insert copyright notice at point	<f11> i C</f11>	(copyright &optional STR ARG)	Insert a copyright by \$ORGANIZATION notice at cursor. ► See also: <u>Inserting Text</u> • If the ORGANIZATION environment variable is not available, Emacs prompts for it.
Update file's copyright notice	M-x copyright- update	(copyright-update &optional ARG INTERACTIVEP)	Update copyright notice to indicate the current year. With prefix ARG, replace the years in the notice rather than adding the current year after them. If necessary, and 'copyright-current-gpl-version' is set, any copying permissions following the copyright are updated as well.
	If you want to be pr	ompted automatically to update	bes not warn if there is no copyright in the current buffer to update. It does not create a missing notice. e an existing but out-of-date copyright notice, write the following inside your init.el file: t-update)
Automatic File Time Stamp on file save References: • TimeStamps @ EmacsWiki • Change time stamp format in: • markdown file • reStucturedText file See also: ∑ Inserting Text	Emacs has a built-in automatic time-stamping of files, It must be activated by adding the time-stamp function to the before-save-hook variable. This can either be done via Emacs customization system or explicitly inside your init file with the following code: (add-hook 'before-save-hook 'time-stamp) * 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:		
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 " <i>Emacs current settings</i> " in the description block above.
Toggle time stamp automatic update	<f11> f M-t</f11>	(time-stamp-toggle-active &optional ARG)	Toggle 'time-stamp-active', setting whether <f11> f t updates a buffer. • With ARG, turn time stamping on if and only if arg is positive.</f11>
RFC-Mode		RFC Files with the following rfc- > 1 to access its PEL customiz	-mode commands. Paquires <u>rfc-mode</u> activated by pel-use-rfc-mode, ration.
Read a specific RFC	<f11> B r</f11>	(rfc-mode-read NUMBER)	Read the RFC document NUMBER. Offer the number at point as default.
Browse RFCs	<f11> B R</f11>	(rfc-mode-browse)	Browse through all RFC documents referenced in the index.
Directory Tree Browsers	Emacs supports several mechanisms to browse file directories. This includes: • Emacs built-in ∑ <u>Dired</u> directory editor, along with several extensions. You can have several different Dired buffers in an Emacs session. • The Emacs built-in ∑ <u>Speedbar</u> and its extensions. There can only be one instance of a Speedbar buffer and that can be inside another frame. • Several other external packages: <u>dir-treeview</u> , <u>Neotree</u> , <u>treemacs</u> , <u>Isp-treemacs</u> and <u>Ztree</u> • Use <f11> B <f2> 1 to access their PEL customization and <f11> B <f3> to access the customization of these packages.</f3></f11></f2></f11>		
dir-treeview	The <u>dir-treeview</u> external package provide a simple to use expandable directory tree view in a buffer. Left PEL activates it when pel-use-dir-treeview is set to t. Access its configuration via <f11> B <f3> 1</f3></f11>		
Browse home (or default) directory tree	<f11> B D</f11>	(dir-treeview)	Display the default directory tree inside the current (or new) <i>Dir Treeview</i> buffer. • Open the directory identified by the dir-treeview-default-root user-option which defaults to the home directory.
Browse selected directory tree	<f11> B d</f11>	(dir-treeview-open &optional DIR)	Prompt for directory, then display its directory tree inside the current (or new) <i>Dir Treeview</i> buffer. • The pro pomp proposes the dir-treeview-default-root user-option which defaults to the home directory.
View Directory Tree with NeoTree	 The NeoTree external package provides a Vim-NerdTree like tree-view of a directory with expansion/collapse. PEL activates it when pel-use-neotree is set to t. <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. Icons used in the tree can be changed: In text mode set pel-neotree-font-in-terminal to arrows to use arrows instead of '+'. In graphics mode, if pel-neotree-font-in-graphics is set to icons then the icons provided by all-the-icons package is used. However, once PEL has installed the package it does not install the fonts. You must install the fonts manually by executing: M-x all-the-icons-install-fonts 		
View directory tree with NeoTree	<f11> B N N</f11>	 U Go up a directory. A Maximize/Minimize the N H Toggle display hidden file O Recursively open a direction 	g Refresh leoTree Window es. Controlled by neo-hidden-regexp-list user option. tory create a directory if filename ends with a '/' a directory. C-c C-r Rename 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.

<u>Operation</u>	<u>Keystroke</u>	Function	<u>Note</u>	
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.	
Manipulate directory trees associated as projects/workspaces Manipulate the directories and files	The treemacs and Isp-treemacs provides workspace/project oriented tree-based view with expansion/collapse and actions of directories and files. PEL activates treemacs when the pel-use-treemacs or pel-use-Isp-treemacs user-option is turned on (set to t). Treemacs has a large number of user-options in the treemacs customization group and sub-groups. Use <f11> B <f2>1 to access its PEL customization for it. and <f11> B <f3> 3 to access its customization group.</f3></f11></f2></f11>			
★★ See: ∑X Treemacs	 In graphics mode the 	On PEL, open (or close) the treemacs buffer with the <f11> B T key sequence. 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.</f11>		
	The treemacs-mode and	d extensions have an extensive	e command set. See <u>IX Treemacs</u> for the complete list	
Open/close treemacs	<f11> B T</f11>	(treemacs)	Initialise or toggle treemacs. See **\textit{\subset}\text{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 workspace is empty additionally ask for the root path of the first project to add.	
View Directory Tree with ZTree	PEL ztree customiza <f11> B <f2> PEL activate Modify one of the repel-ztree-dir-fil pel-ztree-dir-fil pel-ztree-show <f11> B <f3></f3></f11></f2></f11>	tion: opens the PEL customization is it when pel-use-ztree is set following PEL provided custom ove-focus : set to t to mov ter-list : add a list of re For example, to -filtered-files : set to t to disp prompts, select ztree to open t		
View directory as tree with ztree-dir	<f11> B Z</f11>	In the Ztree Dir buffer the follo > : narrow/display directory d : Open Dired at point. H : toggle display of filtered x : Toggle expand/collapse Luse x with care! On time. Investigate.	y on current line < : widen/display parent directory If files. Controlled by regexp in the ztree-dir-filter-list user option.	
Searching/Finding Files See also: •	See: Video: .Emacs # Use man to get mor on locate: <f1 <f11<="" find:="" on="" td=""><td colspan="3">The following commands can be used to search for file by name or content. • See: Video: .Emacs #6 : searching and finding files. Use man to get more information, • on locate: <f11> ? m locate • on find: <f11> ? m find</f11></f11></td></f1>	The following commands can be used to search for file by name or content. • See: Video: .Emacs #6 : searching and finding files. Use man to get more information, • on locate: <f11> ? m locate • on find: <f11> ? m find</f11></f11>		
• <u>\(\tilde{\tilde{D}}\) Dired</u>	You can manipulate		ommands. For instance type (to toggle the display of more than the file names.	
Search for file with locate	<f11> f L</f11>	(locate SEARCH-STRING &optional FILTER ARG)	Prompt for a search pattern and search for filenames using the system locate command line utility 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.	
		(counsel-locate &optional INITIAL-INPUT)	Call a "locate" style shell command with counsel listing and completion user-interface. • INITIAL-INPUT can be given as the initial minibuffer input. ☑ This binding activated when the pel-use-counsel user-option is turned on. ☑ When pel-use-ivy-hydra user-option is set you can activate the ivy-hydra with C-o. When Hydra is active, minibuffer editing is disabled and menus display short aliases: Short Normal Command name o C-g keyboard-escape-quit j C-n ivy-next-line k C-p ivy-previous-line h M-< ivy-beginning-of-buffer 1 M-> ivy-beginning-of-buffer d C-m ivy-done f C-j ivy-alt-done g C-M-m ivy-call u C-c C-o ivy-occur	
Run grep via find	• <f11> f f g</f11>	(find-grep COMMAND- ARGS)	Run grep via find, with user-specified args COMMAND-ARGS. • Collect output in a buffer.	
See also: <u>∑ Grep</u>	• <f11> g f</f11>	, u 100)	 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 'find' and open Dired buffer	<f11> f 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. • Specify the arguments for the <u>find command</u> . • 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 files and open Dired buffer for those	<f11> f f n</f11>	(find-name-dired DIR PATTERN)	Search DIR recursively for files matching the globbing pattern PATTERN, and run Dired on those files. • PATTERN is a shell wildcard (not an Emacs regexp) and need not be quoted. • The default command run (after changing into DIR) is: findname 'PATTERN' -ls	
Find files in a directory and open Dired output	<f11> f f h</f11>	(find-grep-dired DIR REGEXP)	Find files in DIR that contain matches for REGEXP and start Dired on output. The command run (after changing into DIR) is: find . \(-type f -exec 'grep-program' 'find-grep-options' -e REGEXP {} \; \) -ls where the first string in the value of the variable 'find-is-option' specifies what to use in place of "-ls"	
			as the final argument.	
Find Emacs Lisp files in directory tree	<f11> f f 1</f11>	(find-lisp-find-dired DIR REGEXP)	Find Emacs Lisp files in DIR, matching REGEXP. • Open *Find Lisp Dired* buffer on output.	

Operation	<u>Keystroke</u>	Function	<u>Note</u>
Mode Specialized File Open Commands	The following file open of major mode.	commands are only available fo	or some major modes. As such they are also described inside the page describing the corresponding
Open file with alternate extension Supports: • \$\psi \cdot C\$ • \$\psi \cdot C + +	M- <f12> M-f</f12>	(pel-open-file-alternate)	Open a file with same name but an alternate extension. The new extension depends on the current file extension. The list of alternate extensions is currently very limited and restricted to C and C++. If the alternate file is not found, save the file basename in the kill ring and prompt for the file name to open.

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.