File Management

| Size of the Z. Highlafted STEEL & F. 122 STEEL | | | | |
|--|--|--|--|--|
| Services: - 19 Ministed - 2 Customize - 3 Ministed - 3 Customize - 4 Customize - 4 Customize - 5 Ministed - 5 Customize - 6 Customize - 7 Customi | | | | |
| Open table for En. Service 3. Electronic File Fundamental File | See also: • <u>SM Dired</u> | content, displaying dire The directory editing There are also sever customize user optic behaviour of these p | ectory content, etc These are li (dired) commands are mainly list al Emacs internal and external pa on variables. They are listed below backages. PEL provides the <f11< b=""></f11<> | sted in this table. led in the <u>NM Dired</u> table. ckages that provide useful commands. PEL supports several of them, and activates them via w. Use Emacs customize system to modify their values to activate, deactivate and modify the > <1> f key binding to quickly gain access to the appropriate customize group where you can |
| File Lock Protection Management See also 2. Edutationize Grade Edutationize Management See also 2. Edutationize Mile Se | • | | (pel-help-pdf &optional OPEN- | Open the local copy of the <u>▼ File-mngt</u> PDF file unless a command prefix (like C-u) was used. |
| To the transparence Controller English of E | Management | <f11> f <f2> 1</f2></f11> | | Customize PEL support for file management. |
| Controller Fing (find file at point) Open application Open application Open currently file whited in current buffer with a class and the control of the control of the control of the control of the class and class | | <f11> f <f3> 1</f3></f11> | | Customize Emacs support for file management. |
| Open application ### | | <f11> f r <f3></f3></f11> | | Customize Emacs support for file automatic revert management. |
| Open currently file violated declaut 0.5 application. Committed of the properties of the propertie | | <f11> f a <f3></f3></f11> | | Customize Emacs support for management of ffap (find file at point). |
| ### Comments under with the decident OS application. **Discrete Oscillation of Comments are available to oper/vise titles in the received of the protection of the following commends are available to oper/vise titles in the received of the following commends are available to oper/vise titles in the received of the following commends are available to oper/vise titles in the received of the following commends are available to oper/vise titles in the received of the following commends are available to oper/vise titles in the received of the following commends are available to oper/vise titles in the received of the following commends are available of the following commends are available of the following commends are commended to commend the commend of the following commends are available of the following commends are commended to commend the commend of the commended of the | Open application | The following comman | ds open OS level applications | |
| For some of them the corresponding job mode function is also alrow. Note: Times uses the work vising "instead of propring" files: Note: Times uses the work vising" instead of propring" files: Note: Times uses the work vising "instead of propring" files: Note: Times uses the work vising "instead of propring" files: Propring the propring | in current buffer with the default OS application. See also: | <f11> f F</f11> | | ◆ Currently only works on Linux, macOS and Windows. • To open a file without first loading it inside an Emacs buffer, open a Dired buffer and use 'z' on |
| Copen file using OS file- open dialog | | For some of them the corresponding <u>ido</u> mode function is also shown. Note: Emacs uses the word "visiting" instead of "opening" files. 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. | | |
| open disiding C=x C=f - (find.file=FLENANE soptional WILDCARDS) See also: - **S Completion/Input - **SE Direct - **Se E Completion/input for completion modes available at the prompt. - **Se E Completion/input for completion modes available at the prompt. - **See E Completion/input for completion modes available at the prompt. - **See E Completion/input for completion modes available at the prompt. - **See E Completion/input for completion modes available at the prompt. - **See E Completion/input for completion modes available at the prompt. - **See E Completion/input for completion modes available at the prompt. - **See E Completion/input for completion from tread-only mode. - **See E Completion/input for completion from tread-only mode. - **See E Completion/input for completion from tread-only mode. - * | FIIE LOCK Protection | ocked file , Emacs will steal the lock (proceed ('p') to | prompt. You can then: with 's'), | le same me with a lock. In you attempt to edit the burier of a locked me, of save a burier of a |
| See also: ***Scamplation/Input** ***Subjusted** ***This can be replaced by the ido-find-file: it provides suggestions. ***Who figer mote is used, you can also: ***Uper C-f or C-x f lor change to original find-file mode and prevent Ido completion from trying to provide the name of a file that does not exists: ***Uper C-f or C-x f lor change to original find-file mode and prevent Ido completion from trying to provide the name of a file that does not exists: ***Uper C-f or C-x f lor change to original find-file mode and prevent Ido completion from trying to provide the name of a file that does not exists: ***Uper C-f or C-x f lor change to original find-file mode and prevent Ido completion from trying to provide the name of a file that does not exists: ***Uper C-f or C-x f lor change to original find-file mode and prevent Ido completion from trying to provide the name of a file that does not exists: ***Uper C-f or C-x f lor change to original find-file mode and prevent Ido completion from trying to provide the name of a file that does not exists: ***Uper C-f or C-x f lor change to original find-file mode and prevent Ido completion from trying to provide the name of a file that does not exists: ***Uper C-f or C-x f lor change to original find-file mode and prevent Ido completion from trying to provide the name of a file that does not exists: ***Uper C-f or C-x f lor change to original find-file mode and prevent Ido completion from trying to provide the name of a file that does not exists: ***Uper C-f or C-x f lor change to original find-file mode and prevent Ido completion input to the control the way this command operates to help you select a file name; (do, inj., help with not file in the way this command operates to help you select a file name; (do, inj., help with not file in the way this command operates to help you select a file name; (do, inj., help with not file in the file in the name) file in file in the name; (do, inj., help with not file in the name) file in file in the name; (do, inj., he | | %-o | (ns-open-file-using-panel) | 6 On macOS in graphics mode only: open a file, select the file name via an OS File dialog. |
| • This can be replaced by the Ido-mode by the Ido-find-file: it provides suggestions. • This can be replaced by the Ido-mode by the Ido-find-file: it provides suggestions. • 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-f or S-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 file in the canon and the suffer or read-only mode. • To open a file in each only mode you can: • Use C-x C-x f to change the mode of the buffer to read-only mode. • If the commands below (C-x C-x, etc) • Use C-x C-y (find-alternate-file) • FIEL supports dynamic selection of completion input that control the way this command operates to help you select a file name: (ido, ivy, helm). • C-x C-y (find-alternate-file) • FIEL Supports dynamic selection of completion input that control the way this command operates to help you select a file name: (ido, ivy, helm). • C-x C-y (find-alternate-file) • FIEL Supports dynamic selection of completion input that control the way this command operates to help you select a file name: (ido, ivy, helm). • C-x C-y (find-alternate-file) • FIEL Supports dynamic selection of completion input that control the way this command operates to help you select a file name: (ido, ivy, helm). • C-x C-y (find-file-alternate-file) • C-x C-y (find-file-alternate-file) • FIEL Supports dynamic selection of completion input that control the way this command operates to help you select a file name: (ido, ivy, helm). • C-x C-y (find-file-alternate-file) • C-x C-y (find-file-alternate-file) • FIEL Supports dynamic selection of completion input that control the way this command operates to help you select a file name: (ido, ivy, helm). • File FIEL Supports dynamic selection of completion from the way the command selection of the file name: (ido, i | See also: | C-x C-f | &optional WILDCARDS) | |
| FILENAME & optional WILDCARDS (do-find-alternate-file) When ido-mode is used, the ido-find-alternate-file is used instead. Useful when just selected an empty file just selected by mistake. | • <u>∑M Dired</u> | 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 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. This is often useful to open a directory in director (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. | | e mode and prevent Ido completion from trying to provide the name of an existing file when you sts yet. without replacement or suggestion. This is often useful to open a directory in directory editing .) mode of the buffer to read-only mode. |
| C-x 5 f File.NAME apptional WilDCARDS) Gido-find-file-other-window) File.NAME apptional WilDCARDS Gido-find-file-other-window) File.NAME apptional WilDCARDS Gido-find-file-other-frame Edit file File.NAME, in another frame. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new frame or reuses an existing one. Like C-x C-f., but creates a new fram | Open another file in buffer | C-x C-v | FILENAME &optional WILDCARDS) | When ido-mode is used, the ido-find-alternate-file is used instead. |
| FILENAME & optional WIDCARDS) Cido-find-file-other-frame) Like C-x C-f, but creates a new frame or reuses an existing one. | Open file in other window | | (find-file-other-window FILENAME & optional WILDCARDS) (ido-find-file-other- | |
| Open a file in read-only mode C-x C-r (find-file-read-only FILENAME & optional WILDCARDS) (ido-find-file-read-only) Open file in other window in read-only mode C-x 4 r (find-file-read-only-other-window FILENAME & optional WILDCARDS) (ido-find-file-read-only-other-window FILENAME & optional WILDCARDS) (ido-find-file-read-only-other-window FILENAME & optional WILDCARDS) (ido-find-file-read-only-other-window) Open Literally Open a file with no encoding conversion: file is opened in the Fundamental mode: the major mode normally associated with the file type is not used. Visit a file literally: with no encoding support and conversion oncoding support and conversion and character code conversion are both disabled, and multibyte characters are disabled in the resulting buffer. The major mode used is Fundamental mode regardless of the file name, and local variable specifications in the file are ignored. Automatic uncompression and adding a newline at the end of the file due to 'require-final-newline' is also disabled. If Emacs already has a buffer which is visiting the file, this command asks you whether to visiting the file, this command asks you whether to visiting the file, this command asks you whether to visiting the file, this command asks you whether to visiting the file, this command asks you whether to visiting the file, this command asks you whether to visiting the file, this command asks you whether to visiting the file file file file file file file fil | Open file in other frame | C-x 5 f | FILENAME & optional WILDCARDS) | |
| FILENAME & optional WILDCARDS) • (ido-find-file-read-only) Open file in other window in read-only mode • C-x 4 r | Open in read-only | The following comman | ds open files in read-only mode. | While in read-only mode, use Use C-x C-q to permit editing. |
| window FILENAME acoptional WILDCARDS) (ido-find-file-read-only-other-window) Open Literally Open a file with no encoding conversion: file is opened in the Fundamental mode: the major mode normally associated with the file type is not used. Visit a file literally: with no encoding support and conversion (find-file-literally FILENAME) (find-file | | С-х С-г | FILENAME & optional WILDCARDS) | |
| Visit a file literally: with no encoding support and conversion (find-file-literally FILENAME) (Find-file-literally FILENAME) (Find-file-literally FILENAME) (Visit file FILENAME with no conversion of any kind. • Format conversion and character code conversion are both disabled, and multibyte characters are disabled in the resulting buffer. • The major mode used is Fundamental mode regardless of the file name, and local variable specifications in the file are ignored. • Automatic uncompression and adding a newline at the end of the file due to 'require-final-newline' is also disabled. • If Emacs already has a buffer which is visiting the file, this command asks you whether to visiting the file, this command asks you whether to visiting the file, this command asks you whether to visiting the file, this command asks you whether to visiting the file, this command asks you whether to visiting the file, this command asks you whether to visiting the file of the f | | | window FILENAME &optional WILDCARDS) • (ido-find-file-read-only- | Edit file FILENAME in another window but don't allow changes. |
| Format conversion and character code conversion are both disabled, and multibyte characters are disabled in the resulting buffer. The major mode used is Fundamental mode regardless of the file name, and local variable specifications in the file are ignored. Automatic uncompression and adding a newline at the end of the file due to 'require-final-newline' is also disabled. If Emacs already has a buffer which is visiting the file, this command asks you whether to vis | Open Literally | Open a file with no end | coding conversion: file is opened i | n the Fundamental mode: the major mode normally associated with the file type is not used. |
| it ilterally iristead. | encoding support and | <f11> f M-1</f11> | (find-file-literally FILENAME) | Format conversion and character code conversion are both disabled, and multibyte characters are disabled in the resulting buffer. The major mode used is Fundamental mode regardless of the file name, and local variable specifications in the file are ignored. Automatic uncompression and adding a newline at the end of the file due to 'require-final- |

| <u>Operation</u> | <u>Keystroke</u> | Function | <u>Note</u> |
|--|--|---|---|
| Open binary | Open a file in hex binal See <u>∑ Buffers</u> . | ry mode. There are also comman | ds to convert current buffer to hexadecimal editing, like nhexl. |
| Open a file in hexl-mode See also: ∑ 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. |
| Recently opened | When the pel-use-recentf user option is set to t, Emacs remembers the list of recently opened files. The menu bar includes a File->Open Recent menu entry. Some other functions are activated by their respective user options. | | |
| Open recently opened files, listed with Ido | <f11> f f</f11> | (ido-recentf-open) | Open file. Prompt suggests recently opened files wit Ido-style completion. • Type <tab> to get possible expansions listed in a separate buffer. Available only when both pel-use-ido and pel-use-recentf are set to t. Credits: Mickey Petersen recentf article.</tab> |
| Open recently opened files, listed with Counsel | <f11> f R</f11> | (counsel-recentf) | List files recently opened in a counsel buffer. • The list of recently opened files are listed in a Counsel buffer . Select one and type return to open. • Type C-c C-o to copy the list of files inside a special buffer. • Requires Ivy mode completion with Counsel mode and recentf activated: set pel-use-counsel and pel-use-recentf to t |
| Edit list of recently opened files | <f11> f M-r</f11> | (recentf-edit-list) | Show a dialog to delete selected files from the recent list. Use it to remove some of the files from the list. |
| Open file at point | The following comman | ds, followed by the flap command | ds, allow opening files from the file name taken at point (the cursor location). |
| Open file or web-page whose name is at point ★★ | • C-^ • <f11> f . • 6y</f11> | (pel-open-at-point &optional N) | Open the file, library or the URL, named at point, with potential line & column #s. With PEL, the <u>6v</u> key-chord is available if pel-use-key-chord is non-nil. Command prefixes are supported with the key-chord. See <u>Ney-Chords</u> . |
| See also: • Key-Chords • M reStructuredText Onen filename at point in | the page is opened If embedded space(any of the following: The first 12 Tab and newline a If embedded space delimiter, and point of the string identifies Prompts for incomp Currently only of the string identifies If file is already of of the string identifies If file is already of of the string identifies If file is already of of the string identifies If file is already of of the string window, if 2: use With numeric argum N < 0: create an N = 0: use the 'ot N = 1,3,7or above select the targ window, if 3 or N is: 8: up, 2: do N is: 8: up, 2: do N is: 9: open the fi Finder, Windows N is: 10: open the Selecting Minibuffer, If the file name is fol When executed from expanded. More information availa | in the systems' browser). s) are allowed in the filename, the "`' ()[]{}<> ''"" 「」 are part of ASCII but not the rema are also delimiter characters. in the file name is not allowed, the can be positioned anywhere betw be a URL, the function opens the p lete file names, allowing editing the supports Emacs Lisp files. Plannir bened in a window, move point to lis that file, select the target windo the other window, if 3 or more, us ent N: ew window and use that ther' (the next) window be (excluding 9): et window based on the number of more, use the current window. wn, 4:left, 5:current, 6:right. lile in the system's browser, and fo Explorer). URL at point in the system's brow in with a buffer in sh-mode, the '=' able in the command's help docst | lage in the systems' default browser. It is find file (with completion), search for libraries files (type 1) according to current file type. In the support other programming languages with and without project management packages. It is that window and to the line column coordinates if specified following the file name at point. It is based on the number of editable windows in frame: if 1, split that window and use the new is the current window. In the distable windows in frame: if 1, split that window and use the new window, if 2: use the other or a directory name at point open the application associated with directory browsing (eg. macOS exer. In so not allowed. In the systems' default browser. In the support other programming languages with and without project management packages. In the support other programming languages with and without project management packages. In the support other programming languages with and without project management packages. In the support other programming languages with and without project management packages. In the support packages. In the support other programming languages with and without project management packages. In the support packages. |
| 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: | • <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. Optional prefix argument ARG non-nil inverts the value of the option 'browse-url-new-window-flag'. FEL provides the <f11> <f2> E u key sequence to open the browse-url group that contains relevant the user options.</f2></f11> |
| Copy URL at point in temporary file and visit the file | <f11> f M-u C-c C-f</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. |
| | | on (find file at a sixt) | Significant states of the state of the states of the stat |
| 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 Emacs 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 key bindings shown below. When pel-use-ffap is set to ffap-bindings, then PEL 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 | <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. |

| <u>Operation</u> | <u>Keystroke</u> | Function | <u>Note</u> |
|--|---|--|--|
| 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. Then set mark, jump to choice, and try to fetch it. The menu is cached in 'ffap-menu-alist', and rebuilt by 'ffap-menu-rescan'. The optional RESCAN argument (a prefix, interactively) forces a rebuild. Searches with 'ffap-menu-regexp'. |
| Open <u>Dired</u> (Directory Editor) | directory path to Cx C | f then Dired-mode is used. You observe a file directory tree with f | ffer in Dired mode, that looks like a Is -I output, which allows several operations. If you specify a can also use the following commands to open buffer in Dired mode. ile tree browsers, like NeoTree and Ztree, described below in this table. |
| Open a directory editor See also: <u>M Dired</u> <u>Completion/Input</u> | • 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'. ➤ See ∑ Completion/Input for completion modes available at the prompt. |
| 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-addr-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-addr-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 | C-c C-n | (pel-goto-next-url) | Move point forward to the end of the next URL located in the current buffer. |
| See also: <u>Navigation</u> Move to beginning of | <f6> C-n</f6> | (pel-goto-previous-url) | Move point backward to the beginning of the previous URL located in the current buffer. |
| previous URL in buffer See also: Navigation | <f11> C-p</f11> | (per gete protient arr) | |
| 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 commands can be used to write text selected 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 revert-without-query 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. Vou 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> |

| <u>Operation</u> | <u>Keystroke</u> | Function | <u>Note</u> | |
|---|--|--|---|--|
| 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 | |
| | | , u (d) | 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> f r t</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 | <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. • Start a new timer if Global Auto-Revert Mode is active or if Auto-Revert Mode is active in some buffer. • Restarting the timer ensures that Auto-Revert Mode will use an up-to-date value of 'auto-revert-interval' (which is normally 5 seconds by default). : pel-auto-revert-set-timer is a thin wrapper over auto-revert-set-timer that displays a warning if executed when the buffer is not already in auto-revert-mode. It also displays the value of auto-revert-interval when auto-revert-set-timer is executed. | |
| Saving Files | Use the following com | mands to save the content of a bu | Iffer to a filesystem file. | |
| | | | ave. Each of these actions are activated via an action-specific PEL user-option, and can be current buffer. The actions and their associated user-option and command are listed here: | |
| | Action Delete trailing space Update time stamp Update copyright no | e and lines on save pel-delete- on save pel-update | trailing-whitespace pel-toggle-delete-trailing-space-on-save cf11> M-W tropyright pel-toggle-update-time-stamp-on-save cf11> M-T pel-toggle-update-copyright-on-save cf11> M-C | |
| Save file to disk | • C-x C-s • %-s | (save-buffer &optional ARG) | 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. • On macOS in graphics mode only: ℜ-s brings a OS file-save dialog. • 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 | 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. | |
| 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-C</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. It is 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 | <f11> M-W</f11> | (pel-toggle-update-time- stamp-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). 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</f11> 1 This command is only available when the pel-delete-trailing-whitespace user-option s set to t. | |
| Inserting & Automatically Updating Copyrights | Two commands, sho the copyright notice code: | 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-update) | | |
| Insert copyright notice at | limit variable, normally | | ers. This variable is customizable. Insert a copyright by \$ORGANIZATION notice at cursor. | |
| point See also: <u>▶ Inserting Text</u> | | , | If the ORGANIZATION environment variable is not available, Emacs prompts for it. | |

| <u>Operation</u> | <u>Keystroke</u> | Function | <u>Note</u> |
|--|---|---|--|
| Update file's copyright notice | | (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 non-nil, INTERACTIVEP tells the function to behave as when it's called interactively. ⚠ 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) |
| Automatic File Time Stamp on file save | 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: " The format of the time stamp is controlled by several variables: 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. time-stamp-time-zone specifies the time zone selection: nil: Emacs local time t: Universal time vall: system wall clock time To be automatically updated, the time-stamp string must be placed within the first 8 lines of the file. To be automatically updated, the time-stamp string must be placed within the first 8 lines of the file. To insert a non-updatable time stamp, the PEL package provides a set of text insert commands which include inserting a time stamp. See the linserting Text table for the appropriate commands. | | |
| Update file time stamp | <f11> f t</f11> | (time-stamp) | Force update the time stamp string(s) in the current buffer. The time stamp is updated if the one of the following strings is found in the first 8 lines of the file: |
| See also: National Na | | | Time-stamp: <> Time-stamp: " If you want time stamps updated automatically, write the following inside your init.el file: (add-hook 'before-save-hook 'time-stamp) |
| Toggle time stamp automatic update | | (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> |
| Directory Tree Browsers | Emacs supports mechanisms to browse file directories. This includes: • Emacs built-in Moired directory editor, along with several extensions. You can have several different Direct 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 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 (neotree-toggle) Toggle show/hide the NeoTree window. In the NeoTree buffer the following keys are available:</f11> | | |
| | | U Go up a directory g Refresh A Maximize/Minimize the Neo H Toggle display hidden files. O Recursively open a directory | urrent item if it is a file, Fold/Unfold current item if it is a directory. Tree Window Controlled by neo-hidden-regexp-list user option. y ate a directory if filename ends with a '/' rectory. directory. rectory. |
| 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. |
| View Directory Tree with ZTree | The <u>ztree external package</u> provides a text-based tree-view of a directory with expansion/collapse. PEL ztree customization: • <f11> B <f2> opens the PEL customization group (select the tree subgroup) . See also: Customize. • We pel activates it when pel-use-ztree is set to t. • Modify one of the following PEL provided customization user options: • pel-ztree-dir-move-focus : set to t to move focus to new entry when <ret> is typed. • pel-ztree-dir-filter-list : add a list of regexp to ignore more file. Do not enter quote for string. For example, to ignore the .pyc files, enter ^.*pyc on a line. • pel-ztree-show-filtered-files : set to t to display filtered files until H is typed. Normally they are not shown until H is typed. • <f11> B <f3> prompts, select ztree to open the ztree customization group itself. 1. Execute M-x pel-init after settling and applying new values to activate the new values.</f3></f11></ret></f2></f11> | | |
| View directory as tree with ztree-dir | <f11> B Z</f11> | (ztree-dir PATH) | Open an interactive buffer with the directory tree of the PATH given. Opens the tree buffer in the current window. There can be several buffers with different ztree-dir trees. |

| <u>Operation</u> | <u>Keystroke</u> | Function | <u>Note</u> | |
|--|---|---|--|--|
| | | In the Ztree Dir buffer the followi > : narrow/display directory or d: Open Dired at point. H: toggle display of filtered fil x: Toggle expand/collapse of Use x with care! On la time. Investigate. | n current line < : widen/display parent directory es. Controlled by regexp in the ztree-dir-filter-list user option. | |
| Searching/Finding Files | | The following commands can be used to search for file by name or content. Ref: Video: .Emacs #6: searching and finding files. | | |
| Run grep via find See also: <u>▼ Grep</u> | • <f11> f g • <f11> g f</f11></f11> | (find-grep COMMAND-ARGS) | Run grep via find, with user-specified args COMMAND-ARGS. Collect output in a buffer. 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 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. The Search Path Search Pa | |
| Search for files with 'find' and open Dired buffer | <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. • Specify the arguments for the <u>find command</u> . For example, to perform a case insensitive search for all .h files: -iname "*\h" for all .h files. • Opens a Dired-mode buffer and show the files found in there. | |
| Search directory for files and open Dired buffer for those | <f11> 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 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-ls-option' specifies what to use in place of "-ls" as the final argument. | |
| Find Emacs Lisp files in directory tree | <f11> f 1</f11> | (find-lisp-find-dired DIR REGEXP) | Find Emacs Lisp files in DIR, matching 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? | | | |