




File & Directory Management with Treemacs

| Operation | Keystroke | Function | Note |
|---|---|---|---|
| Treemacs <ul style="list-style-type: none"> Manipulate directory trees associated as projects/workspaces Manipulate the directories and files See also: ↗ Dired ↗ File-mngt |  The treemacs external package provides a workspace/project oriented tree-based view with expansion/collapse and actions of directories and files.  PEL activates treemacs when the pel-use-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. PEL <f11> B <f3> key sequence gives access to the customization group. | | With PEL, open (or close) the treemacs buffer with the <f11> B T key sequence. <ul style="list-style-type: none"> In graphics mode the mouse provides access to most commands. In terminal mode use the mouse to quickly put point on a directory or file node. |
| Open/close treemacs | <f11> B T | (treemacs) | Initialise or toggle treemacs. <ul style="list-style-type: none"> 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. |
| Configure Treemacs window | The following commands control the size and minor modes available to extend the Treemacs window behaviour. <ul style="list-style-type: none"> These commands are available in the Treemacs window that was opened with treemacs command above. | | |
| Set Treemacs window width | w | (treemacs-set-width &optional ARG) | Select a new value for 'treemacs-width'. With a prefix ARG simply reset the width of the treemacs window. |
| Toggle file-watch mode on specified file | t a | (treemacs-filewatch-mode &optional ARG) | Minor mode to let treemacs auto-refresh itself on file system changes. <ul style="list-style-type: none"> Activating this mode enables treemacs to watch the files it is displaying (and only those) for changes and automatically refresh its view when it detects a change that it decides is relevant. A file change event is relevant for treemacs if a new file has been created or deleted or a file has been changed and 'treemacs-git-mode' is enabled. Events caused by files that are ignored as per 'treemacs-ignored-file-predicates' are counted as not relevant. The refresh is not called immediately after an event was received, treemacs instead waits 'treemacs-file-event-delay' ms to see if any more files have changed to avoid having to refresh multiple times over a short period of time. The watch mechanism only applies to directories opened *after* this mode has been activated. This means that to enable file watching in an already existing treemacs buffer it needs to be torn down and rebuilt by calling 'treemacs' or 'treemacs-projectile'. Turning off this mode is, on the other hand, instantaneous - it will immediately turn off all existing file watch processes and outstanding refresh actions. |
| Toggle treemacs follow-mode | t f | (treemacs-follow-mode &optional ARG) | Toggle treemacs-follow-mode. When enabled treemacs will keep track of and focus the currently selected buffer's file: <ul style="list-style-type: none"> This only applies if the file is within the treemacs root directory. This functionality can also be manually invoked with 'treemacs-find-file' |
| Toggle treemacs git mode | t g | (treemacs-git-mode &optional ARG) | Toggle 'treemacs-git-mode'. <ul style="list-style-type: none"> When enabled treemacs will check files' git status and highlight them accordingly. This git integration is available in 3 variants: simple, extended and deferred. <ul style="list-style-type: none"> The simple variant will start a git status process whose output is parsed in elisp. This version is simpler and slightly faster, but incomplete - it will highlight only files, not directories. The extended variant requires a non-trivial amount of parsing to be done, which is achieved with python (specifically python3). It is slightly slower, but complete - both files and directories will be highlighted according to their git status. The deferred variant is the same is extended, except the tasks of rendering nodes and highlighting them are separated. The former happens immediately, the latter after 'treemacs-deferred-git-apply-delay' seconds of idle time. This may be faster (if not in truth then at least in appearance) as the git process is given a much greater amount of time to finish. The downside is that the effect of nodes changing their colours may be somewhat jarring, though this issue is largely mitigated due to the use of a caching layer. All versions run asynchronously and are optimized for not doing more work than is necessary, so their performance cost should, for the most part, be the constant time needed to fork a subprocess. |
| Toggle showing dot files | t h | (treemacs-toggle-show-dotfiles) | Toggle the hiding and displaying of dotfiles. |
| Toggle/set treemacs fringe indicator | t v | (treemacs-fringe-indicator-mode &optional ARG) | Toggle 'treemacs-fringe-indicator-mode'. <ul style="list-style-type: none"> When enabled, a visual indicator in the fringe will be displayed to highlight the selected line in addition to 'hl-line-mode'. Useful if 'hl-line-mode' doesn't stand out enough with your colour theme. Can be called with one of two arguments: <ul style="list-style-type: none"> 'always' will always show the fringe indicator. 'only-when-focused' will only show the fringe indicator when the treemacs window is focused (only possible with Emacs 27+). For backward compatibility just enabling this mode without an explicit argument has the same effect as using 'always'. |
| Toggle fixed-width font in treemacs buffer | t w | (treemacs-toggle-fixed-width) | Toggle whether the local treemacs buffer should have a fixed width. <ul style="list-style-type: none"> See also 'treemacs-width.' Has no effect in terminal mode. |
| Treemacs Hydras | Inside the Treemacs buffer you can use the two hydras keys that simplify operation and provide visual list of available commands. | | |
| Open Treemacs Common Hydra | * ? | (treemacs-common-helpful-hydra) | Summon a helpful hydra to show you the treemacs keymap. <ul style="list-style-type: none"> This hydra will show the most commonly used keybinds for treemacs. For the more advanced (probably rarely used keybinds) see 'treemacs-advanced-helpful-hydra'. The keybinds shown in this hydra are not static, but reflect the actual keybindings currently in use (including evil mode). If the hydra is unable to find the key a command is bound to it will show a blank instead. |

Treemacs

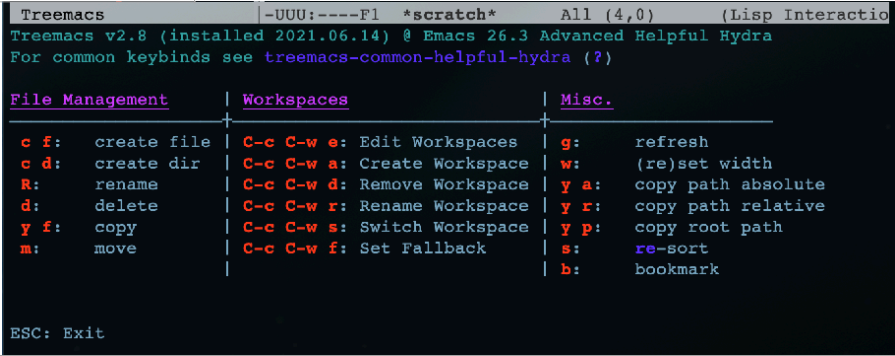
-UUU:----F1 *scratch* All (4,0) (Lisp Interaction WK LY Fly ? Anzu

Treemacs v2.8 (installed 2021.06.14) @ Emacs 26.3 Common Helpful Hydra




For advanced keybinds see treemacs-advanced-helpful-hydra (C-?)

| Navigation | Opening Nodes | Toggles | Projects |
|------------------------------|-----------------------------------|------------------------------|----------------------------------|
| n: next line | TAB: dwim TAB | t f: follow mode | C-c C-p a: add project |
| p: prev line | l: dwim RET | t a: filewatch mode | C-c C-p d: remove project |
| M-n: next neighbour | o o: open no split | t g: git mode | C-c C-p r: rename project |
| M-p: prev neighbour | o h: open horizontal | t h: show dotfiles | |
| u: goto parent | o v: open vertical | t w: resizability | |
| M-N: down next window | o a a: open ace | t v: fringe indicator | |
| M-P: up next window | o a h: open ace horizontal | | |
| M-H: root up | o a v: open ace vertical | | |
| M-L: root down | o r: open mru window | | |
| | o x: open externally | | |
| | H: close parent | | |

ESC: Exit

| Operation | Keystroke | Function | Note |
|---|---|---|--|
| Open Treemacs Advanced Hydra | ✱ C-? ✱ M-? | (treemacs-advanced-helpful-hydra) | <p>Summon a helpful hydra to show you the treemacs keypad.</p> <ul style="list-style-type: none"> This hydra will show the more advanced (rarely used) keybinds for treemacs. For the more commonly used keybinds see ‘treemacs-common-helpful-hydra’ above The keybinds shown in this hydra are not static, but reflect the actual keybindings currently in use (including evil mode). If the hydra is unable to find the key a command is bound to it will show a blank instead. <p>The C-? key binding is only available in graphics mode.</p> <p>In terminal mode invoke the command via the M-x key. PEL provides the M-? key.</p>  |
| Treemacs Workspace Management | Treemacs associates a set of directory managed and viewed as a <i>Treemacs workspace</i> . There is always one workspace. You can create more, and each one has a name. | | |
| Create a new workspace | C-c C-w a | (treemacs-create-workspace) | Create a new workspace. |
| Delete a workspace | C-c C-w d | (treemacs-remove-workspace) | Delete a workspace. |
| Edit Treemacs workspaces as Org file See also: • M-J Outline/Org-Mode | C-c C-w e | (treemacs-edit-workspaces) | Edit your treemacs workspaces and projects as an ‘org-mode’ file. |
| Set fallback workspace | C-c C-w f | (treemacs-set-fallback-workspace &optional ARG) | <p>Set the current workspace as the default fallback.</p> <ul style="list-style-type: none"> With a non-nil prefix ARG choose the fallback instead. The fallback workspace is the one treemacs will select when it is opened for the first time and the current file at the time is not part of any of treemacs’ workspaces. |
| Rename workspace | C-c C-w r | (treemacs-rename-workspace) | <p>Rename a workspace.</p> <ul style="list-style-type: none"> Prompt to select a workspace to rename and then for the new workspace name. |
| Select another workspace | C-c C-w s | (treemacs-switch-workspace ARG) | <p>Select a different workspace for treemacs. Prompts for the other workspace name.</p> <ul style="list-style-type: none"> With a prefix ARG clean up buffers after the switch. A single prefix argument will delete all file visiting buffers, 2 prefix arguments will clean up all open buffers (except for treemacs itself and the scratch and messages buffers). Without a prefix argument ‘treemacs-workspace-switch-cleanup’ will be followed instead. |
| Add project to current workspace | C-c C-p a | (treemacs-add-project-to-workspace PATH &optional NAME) | <p>Add a project at given PATH to the current workspace.</p> <ul style="list-style-type: none"> The PATH’s directory name will be used as a NAME for a project. The NAME can (or must) be entered manually with either a prefix arg or if a project with the auto-selected name already exists. |
| Removed project from workspace | C-c C-p d | (treemacs-remove-project-from-workspace &optional ARG) | <p>Remove the project at point from the current workspace.</p> <ul style="list-style-type: none"> With a prefix ARG select project to remove by name. |
| Renamed project at point | C-c C-p r | (treemacs-rename-project) | Give the project at point a new name. |
| Treemacs window view | | | |
| Collapse parent node | H | (treemacs-collapse-parent-node ARG) | <p>Close the parent of the node at point.</p> <ul style="list-style-type: none"> Prefix ARG will be passed on to the closing function (see ‘treemacs-toggle-node’.) |
| Collapse current node | • h • M-h | (treemacs-COLLAPSE-action &optional ARG) | <p>Run the appropriate COLLAPSE action for the current button.</p> <ul style="list-style-type: none"> In the default configuration this usually means to close the content of the currently selected node. A potential prefix ARG is passed on to the executed action, if possible. This function’s exact configuration is stored in ‘treemacs-COLLAPSE-actions-config’ |
| Collapse all projects nodes | • <backtab> • <S-tab> • C-c C-p c a | (treemacs-collapse-all-projects &optional ARG) | <p>Collapses all projects.</p> <ul style="list-style-type: none"> With a prefix ARG also forget about all the nodes opened in the projects. |
| Collapse current project | C-c C-p c c | (treemacs-collapse-project &optional ARG) | <p>Close the project at point.</p> <ul style="list-style-type: none"> With a prefix ARG also forget about all the nodes opened in the project. |
| Collapse all other projects | C-c C-p c o | (treemacs-collapse-other-projects &optional ARG) | <p>Collapses all projects except the project at point.</p> <ul style="list-style-type: none"> With a prefix ARG also forget about all the nodes opened in the projects. |
| Cleanup litter | C | (treemacs-cleanup-litter) | <p>(treemacs-cleanup-litter)</p> <ul style="list-style-type: none"> Collapse all nodes matching any of ‘treemacs-litter-directories’. This defaults to (“/node_modules” “.venv” “./cask”) |
| Scroll up | • SPC • C-v | (scroll-up-command &optional ARG) | <p>Scroll text of selected window upward ARG lines; or near full screen if no ARG.</p> <ul style="list-style-type: none"> If ‘scroll-error-top-bottom’ is non-nil and ‘scroll-up’ cannot scroll window further, move cursor to the bottom line. When point is already on that position, then signal an error. A near full screen is ‘next-screen-context-lines’ less than a full screen. Negative ARG means scroll downward. If ARG is the atom ‘-’, scroll downward by nearly full screen. |
| Scroll down | • S-SPC • DEL • M-v | (scroll-down-command &optional ARG) | <p>Scroll text of selected window down ARG lines; or near full screen if no ARG.</p> <ul style="list-style-type: none"> If ‘scroll-error-top-bottom’ is non-nil and ‘scroll-down’ cannot scroll window further, move cursor to the top line. When point is already on that position, then signal an error. A near full screen is ‘next-screen-context-lines’ less than a full screen. Negative ARG means scroll upward. If ARG is the atom ‘-’, scroll upward by nearly full screen. |
| Scroll other window forward | M-N | (treemacs-next-line-other-window &optional COUNT) | <p>Scroll forward COUNT lines in ‘next-window’.</p> <ul style="list-style-type: none"> That window is normally the window that holds the file last opened by treemacs. |
| Scroll other window backward | M-P | (treemacs-previous-line-other-window &optional COUNT) | <p>Scroll backward COUNT lines in ‘next-window’.</p> <ul style="list-style-type: none"> That window is normally the window that holds the file last opened by treemacs. |

| Operation | Keystroke | Function | Note |
|--|---|--|---|
| Navigation in Treemacs window | The following commands move point inside the Treemacs window. 👉 This is Emacs: you can also use any other ⌘ Navigation command. You can also perform search commands. See ⌘ Search/Replace . | | |
| • Move across projects | | | |
| To next project | C-j | (treemacs-next-project) | Move to the next project root node. |
| To previous project | C-k | (treemacs-previous-project) | Move to the next project root node. |
| To project down | <M-down> | (treemacs-move-project-down) | Switch position of the project at point and the one below it. |
| To project up | <M-up> | (treemacs-move-project-up) | Switch position of the project at point and the one above it. |
| • Move within a project | The <down> and <up> cursor keys also move up and down which have the same impact as n and p but are not implemented by Treemacs. | | |
| Move point to next line | <ul style="list-style-type: none">n<down> | (treemacs-next-line &optional COUNT) | Go to next line. <ul style="list-style-type: none">A COUNT argument, moves COUNT lines down. |
| Select next neighbour node | M-n | (treemacs-next-neighbour) | Select next node at the same depth as currently selected node, if possible. <ul style="list-style-type: none">If point is on an expanded directory with files, moves point to the neighbour of that directory, instead of the first file of that directory. |
| Move point to previous line | <ul style="list-style-type: none">p<up> | (treemacs-previous-line &optional COUNT) | Go to previous line. <ul style="list-style-type: none">A COUNT argument, moves COUNT lines up. |
| Select previous neighbour node | M-p | (treemacs-previous-neighbour) | Select previous node at the same depth as currently selected node, if possible. |
| Move point to parent node | u | (treemacs-goto-parent-node &optional ARG) | Select parent of selected node, if possible. <ul style="list-style-type: none">ARG is optional and only available so this function can be used as an action. No effect. |
| Manage Treemacs Window | | | |
| Refresh Treemacs buffer | g | (treemacs-refresh) | Refresh the project at point. Use it when directory content was changed from outside Treemacs |
| Kill Treemacs buffer | Q | (treemacs-kill-buffer) | Kill the treemacs buffer. |
| Operate on Treemacs Nodes | | | |
| • Operate on nodes | | | |
| Add Bookmark | b | (treemacs-add-bookmark) | Add the current node to Emacs' list of bookmarks. <ul style="list-style-type: none">For file and directory nodes their absolute path is saved. Tag nodes additionally also save the tag's position. A tag can only be bookmarked if the treemacs node is pointing to a valid buffer position. |
| Refresh project | r | (treemacs-refresh) | Refresh the project at point. |
| Sort | s | (treemacs-resort &optional ARG) | Select a new permanent value for 'treemacs-sorting' and refresh. <ul style="list-style-type: none">Prompt for a sorting method. Supports tab completion.With a single prefix ARG use the new sort value to "temporarily" resort the (closest) directory at point.With a double prefix ARG use the new sort value to "temporarily" resort the entire treemacs view.Temporary sorting will only stick around until the next refresh, either manual or automatic via 'treemacs-filewatch-mode'.Instead of calling this with a prefix arg you can also directly call 'treemacs-temp-resort-current-dir' and 'treemacs-temp-resort-root'. |
| Move Treemacs root one level upward | M-H | (treemacs-root-up &optional _) | Move treemacs' root one level upward. <ul style="list-style-type: none">Only works with a single project in the workspace. |
| Move Treemacs root into directory at point | M-L | (treemacs-root-down &optional _) | Move treemacs' root into the directory at point. <ul style="list-style-type: none">Only works with a single project in the workspace. |
| • Collapse/Visit/View | • Collapse/Expand Nodes Visit files and Directories in other windows | | |
| Return action: <ul style="list-style-type: none">Expand NodeCollapse Node... | <ul style="list-style-type: none">RET1M-1 | (treemacs-RET-action &optional ARG) | Run the appropriate RET action for the current button. <ul style="list-style-type: none">In the default configuration this usually means to open the content of the currently selected node. A potential prefix ARG is passed on to the executed action, if possible.This function's exact configuration is stored in 'treemacs-RET-actions-config'. |
| Tab action: <ul style="list-style-type: none">Expand NodeCollapse Node... | <tab> | (treemacs-TAB-action &optional ARG) | Run the appropriate TAB action for the current node. <ul style="list-style-type: none">In the default configuration this usually means to expand or close the content of the currently selected node. A potential prefix ARG is passed on to the executed action, if possible.This function's exact configuration is stored in 'treemacs-TAB-actions-config'. |
| Peek at node | P | (treemacs-peek) | Toggle peek mode. When on: moving to a node peeks at the content of the node at point. <ul style="list-style-type: none">This will display the file (or tag) at point in 'next-window' much like 'treemacs-visit-node-no-split' would. The difference that the file is not really (or rather permanently) opened - any command other than 'treemacs-peek', 'treemacs-next-line-other-window', 'treemacs-previous-line-other-window', 'treemacs-next-page-other-window' or 'treemacs-previous-page-other-window' will cause it to be closed again and the previously shown buffer to be restored. The buffer visiting the peeked file will also be killed again, unless it was already open before being used for peeking.👉 Useful mode to quickly familiarize yourself with a set of files by navigating through them. |
| Open file/dir/tag | o o | (treemacs-visit-node-no-split &optional ARG) | Open current file or tag within the window the file is already opened in. <ul style="list-style-type: none">If the file/tag is no visible opened in any window use 'next-window' instead.Stay in current window with a prefix argument ARG. |
| Open file/dir/tag in most recently used windows | o r | (treemacs-visit-node-in-most-recently-used-window &optional ARG) | Open current file or tag in window selected by 'get-mru-window'. <ul style="list-style-type: none">Stay in current window with a prefix argument ARG. |
| Open file/directory in other window selected by ace-window | o a a | (treemacs-visit-node-ace &optional ARG) | Open current file or tag in window selected by ace-window . <ul style="list-style-type: none">Stay in current window with a prefix argument ARG.See ⌘ Windows |
| Open file/dir/tag in horizontal split of current window | o h | (treemacs-visit-node-horizontal-split &optional ARG) | Open current file or tag by horizontally splitting 'next-window'. <ul style="list-style-type: none">Stay in current window with a prefix argument ARG. |
| Open file/directory in horizontal split of window selected by ace-window | o a h | (treemacs-visit-node-ace-horizontal-split &optional ARG) | Open current file by horizontally splitting window selected by ace-window . <ul style="list-style-type: none">Stay in current window with a prefix argument ARG.See ⌘ Windows |
| Open file/dir/tag in vertical split of current window | o v | (treemacs-visit-node-vertical-split &optional ARG) | Open current file or tag by vertically splitting 'next-window'. <ul style="list-style-type: none">Stay in current window with a prefix argument ARG. |
| Open file/directory in vertical split of window selected by ace-window | o a v | (treemacs-visit-node-ace-vertical-split &optional ARG) | Open current file by vertically splitting window selected by ace-window . <ul style="list-style-type: none">Stay in current window with a prefix argument ARG.See ⌘ Windows |

| Operation | Keystroke | Function | Note |
|---|-------------|--|---|
| Open file/dir with OS application    | o x | (treemacs-visit-node-in-external-application) | Open current file according to its mime type in an external application. Treemacs knows how to open files on linux, windows and macos. |
| • Operate on file/directories | | | |
| Rename | R | (treemacs-rename) | Rename the currently selected node. <ul style="list-style-type: none"> Buffers visiting the renamed file or visiting a file inside a renamed directory and windows showing them will be reloaded. The list of recent files will likewise be updated. |
| Delete | d | (treemacs-delete &optional ARG) | Delete node at point. <ul style="list-style-type: none"> A delete action must always be confirmed. Directories are deleted recursively. By default files are deleted by moving them to the trash. With a prefix ARG they will instead be wiped irreversibly. |
| Move | m | (treemacs-move-file) | Move file (or directory) at point. <ul style="list-style-type: none"> Destination may also be a filename, in which case the moved file will also be renamed. |
| Run shell command | ! | (treemacs-run-shell-command-for-current-node &optional ARG) | Run a shell command on the current node. <ul style="list-style-type: none"> Output will only be saved and displayed if prefix ARG is non-nil. Will use the location of the current node as working directory. If the current node is not a file/dir, then the next-closest file node will be used. If all nodes are non-files, or if there is no node at point, \$HOME will be set as the working directory. Every instance of the string '\$path' will be replaced with the (properly quoted) absolute path of the node (if it is present). |
| Run shell command in root of current project | M- ! | (treemacs-run-shell-command-in-project-root &optional ARG) | Run an asynchronous shell command in the root of the current project. <ul style="list-style-type: none"> Output will only be saved and displayed if prefix ARG is non-nil. Every instance of the string '\$path' will be replaced with the (properly quoted) absolute path of the project root. |
| • Operate on Files | | | |
| Copy file/dir | y f | (treemacs-copy-file) | Copy file (or directory) at point. Prompts for destination. Destination may also be a filename, in which case the copied file will also be renamed. |
| Copy absolute path | y a | (treemacs-copy-absolute-path-at-point) | Copy the absolute path of the node at point. |
| Copy absolute path of project root | y p | (treemacs-copy-project-path-at-point) | Copy the absolute path of the current treemacs root. |
| Copy relative path node in project root | y r | (treemacs-copy-relative-path-at-point) | Copy the path of the node at point relative to the project root. |
| Create a new directory | c d | (treemacs-create-dir) | Create a new directory. Prompts for directory name. <ul style="list-style-type: none"> Enter first the directory to create the new dir in, then the new dir's name. The pre-selection for what directory to create in is based on the "nearest" path to point - the containing directory for tags and files or the directory itself, using \$HOME when there is no path at or near point to grab. |
| Create a new file | c f | (treemacs-create-file) | Create a new file. <ul style="list-style-type: none"> Enter first the directory to create the new file in, then the new file's name. The pre-selection for what directory to create in is based on the "nearest" path to point - the containing directory for tags and files or the directory itself, using \$HOME when there is no path at or near point to grab. |