








Dired — Directory Editor

| Description | Keystroke | Function | Note |
|--|--|--|---|
| Dired | Dired is Emacs directory editor. You can open a Dired buffer by visiting a directory instead of a file. With it you can operate on the files of the directory and its sub-directories. | | |
| Entering Dired Mode | The following commands can be used to list files in a directory a=in a buffer that operates in Dired Mode. | | |
| Open a directory editor (See also: ⌘ File mngt) | <ul style="list-style-type: none">• C-x d• ⌘-D | <ul style="list-style-type: none">• (dired DIRNAME &optional SWITCHES)— — — — —— — —• (ido-dired) | Opens a Dired-mode buffer on the specified directory. Prompt for the directory name.  The PEL package use IDO when the pel-use-ido-mode customize variable is set to t and then the keys are bound to the (ido-dired) command. |
| Open (visit) a file/directory (See also: ⌘ File mngt) | C-x C-f | <ul style="list-style-type: none">• (find-file FILENAME &optional WILDCARDS)— — — — —— — —• (ido-find-file) | 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. This can be replaced by the ido-mode by the ido-find-file: it provides suggestions. When ido mode is used, you can also: <ul style="list-style-type: none">• Type C-x f to change to original find-file• Type C-j to accept the file/directory name verbatim without replacement or suggestion. Note: it is also possible to change the read-only state of a buffer with C-x C-q . So you can open a file with C-x C-f and then change the buffer to read-only mode.  The PEL package use IDO when the pel-use-ido-mode customize variable is set to t and then the keys are bound to the (ido-dired) command. |
| Dired Mode commands | A buffer in Dired Mode provides a large set of specialized single key commands. There's also a set of secondary/extended Dired Modes that can be activated via some of these commands. As an example on what can be done, the following describes the keys to use to show only the files that have a specific file extension: <ul style="list-style-type: none">• Mark all .el files: %f \.el\$ RET• Toggle the marking to mark all other files: t• Kill the lines that are now marked: k• To refresh the file list: g | | |
| Help - summary of available commands | ? | (dired-summary) | Summarize basic Dired commands on the echo line and show recent Dired errors. |
| Help - complete help | <ul style="list-style-type: none">• h• <f1> m | (describe-mode &optional BUFFER) | Display documentation of Dired major mode and available minor modes. <ul style="list-style-type: none">• A brief summary of the minor modes comes first, followed by the major mode description. This is followed by detailed descriptions of the minor modes, each on a separate page. |
| Navigate | | | |
| Move down one line | <ul style="list-style-type: none">• SPC• n• C-n | (dired-next-line ARG) | Down one line |
| Move up one line | <ul style="list-style-type: none">• S-SPC• p• C-p | (dired-previous-line ARG) | Up one line |
| Move to next marked file | <ul style="list-style-type: none">• M-}• * C-n | (dired-next-marked-file ARG &optional WRAP OPOINT) | Move to the ARGth next marked file. <ul style="list-style-type: none">• ARG is the numeric prefix argument and defaults to 1.• If WRAP is non-nil, which happens interactively, wrap around to the beginning of the buffer and search from there, if no marked file is found after this line.• Optional argument OPOINT specifies the buffer position to return to if no ARGth marked file is found; it defaults to the position where this command was invoked. |
| Move to previous marked file | <ul style="list-style-type: none">• M-{• * C-p | (dired-prev-marked-file ARG &optional WRAP) | Move to the ARGth previous marked file. <ul style="list-style-type: none">• ARG is the numeric prefix argument and defaults to 1.• If WRAP is non-nil, which happens interactively, wrap around to the end of the buffer and search backwards from there, if no ARGth marked file is found before this line. |
| Move to the next directory line | > | (dired-next-dirline ARG &optional OPOINT) | Goto ARGth next directory file line. |
| Move to previous directory line | < | (dired-prev-dirline ARG) | Goto ARGth previous directory file line. |
| Control Visibility | | | |
| Toggle sorting lines by date | s | (dired-sort-toggle-or-edit &optional ARG) | Toggle sorting by date, and refresh the Dired buffer. With a prefix argument, edit the current listing switches instead. |
| Toggle visibility of details info | (| (dired-hide-details-mode &optional ARG) | Toggle visibility of detailed information in current Dired buffer. <ul style="list-style-type: none">• When this minor mode is enabled, details such as file ownership and permissions are hidden from view.  User options: 'dired-hide-details-hide-symlink-targets', 'dired-hide-details-hide-information-lines'. |
| Hide/unhide current sub-directory, move to next one | \$ | (dired-hide-subdir ARG) | Hide or unhide the current subdirectory and move to next directory. <ul style="list-style-type: none">• Optional prefix arg is a repeat factor. |
| Hide/Show all sub-directories | M-\$ | (dired-hide-all &optional IGNORED) | Hide all subdirectories, leaving only their header lines. <ul style="list-style-type: none">• If there is already something hidden, make everything visible again.• Use M-x dired-hide-subdir to (un)hide a particular subdirectory. |
| Refresh | g | (revert-buffer &optional IGNORE-AUTO NOCONFIRM PRESERVE-MODES) | Refresh: read directory again. |
| Re-display from point | l | (dired-do-redisplay &optional ARG TEST-FOR-SUBDIR) | Relist the file at point or the marked files, or a subdirectory. <ul style="list-style-type: none">• Redisplay all marked (or next ARG) files.• If on a subdir line, redisplay that subdirectory. In that case, a prefix arg lets you edit the 'ls' switches used for the new listing. |
| Quit Dired Window | q | (quit-window &optional KILL WINDOW) | Quit WINDOW and bury its buffer. <ul style="list-style-type: none">• WINDOW must be a live window and defaults to the selected one.• With prefix argument KILL non-nil, kill the buffer instead of burying it. |
| Visit Files | | | |
| Visit file/directory on line | <ul style="list-style-type: none">• <RET>• e• f | (dired-find-file) | Visit current line's file or directory. For directory create new Dired buffer. |
| Visit parent directory | ^ | (dired-up-directory &optional OTHER-WINDOW) | Up one directory level (creates a new buffer) |
| Visit file/directory on line in current buffer | a | (dired-find-alternate-file) | Access current line's file or directory. Kill old Dired buffer. |
| Dired-subtree | i | (dired-maybe-insert-subdir DIRNAME &optional SWITCHES NO-ERROR-IF-NOT-DIR-P) | Insert a subdirectory listing in this buffer. Use on subdirectory line. |

| Description | Keystroke | Function | Note |
|--|--|--|--|
| Toggle al marks | t | (dired-toggle-marks) | Toggle marks: marked files become unmarked, and vice versa. <ul style="list-style-type: none"> Files marked with other flags (such as ‘D’) are not affected. ‘.’ and ‘..’ are never toggled. As always, hidden subdirs are not affected. |
| Change mark type | | | |
| Change mark type | * c | (dired-change-marks &optional OLD NEW) | Change all OLD marks to NEW marks. <ul style="list-style-type: none"> OLD and NEW are both characters used to mark files. |
| Kill/Delete | | | |
| Kill all marked lines (not files) | k | (dired-do-kill-lines &optional ARG FMT) | Kill all marked lines (not the files). <ul style="list-style-type: none"> With a prefix argument, kill that many lines starting with the current line. (A negative argument kills backward.) If you use this command with a prefix argument to kill the line for a file that is a directory, which you have inserted in the Dired buffer as a subdirectory, then it deletes that subdirectory from the buffer as well. To kill an entire subdirectory (without killing its line in the parent directory), go to its directory header line and use this command with a prefix argument (the value does not matter). |
| Delete files marked for deletion | x | (dired-do-flagged-delete &optional NOMESSAGE) | Delete (eXpunge) the files flagged ‘D’. |
| Delete marked files/directories | D | (dired-do-delete &optional ARG) | Delete all marked (or next ARG) files.  dired-recursive-deletes’ controls whether deletion of non-empty directories is allowed. |
| Create | | | |
| Create directory | + | (dired-create-directory DIRECTORY) | Create a new directory. <ul style="list-style-type: none"> Parent directories of DIRECTORY are created as needed. If DIRECTORY already exists, signal an error. |
| Copy | | | |
| Copy file/directory | C | (dired-do-copy &optional ARG) | Copy all marked (or next ARG) files, or copy the current file. When operating on just the current file, prompt for the new name. |
| Copy files which match regexp | %C | (dired-do-copy-regexp REGEXP NEWNAME &optional ARG WHOLE-NAME) | Copy selected files whose names match REGEXP to NEWNAME. <ul style="list-style-type: none"> With non-zero prefix argument ARG, the command operates on the next ARG files. Otherwise, it operates on all the marked files, or the current file if none are marked. As each match is found, the user must type a character saying what to do with it. For directions, type C–h at that time. NEWNAME may contain \<n> or \& as in ‘query-replace-regexp’. REGEXP defaults to the last regexp used. With a zero prefix arg, renaming by regexp affects the absolute file name. Normally, only the non-directory part of the file name is used and changed. |
| Rename | | | |
| Rename file/move file into other directory | R | (dired-do-rename &optional ARG) | Rename a file or move the marked files to another directory. |
| Rename files selected by regexp | <ul style="list-style-type: none"> %R %r | (dired-do-rename-regexp REGEXP NEWNAME &optional ARG WHOLE-NAME) | Rename selected files whose names match REGEXP to NEWNAME. <ul style="list-style-type: none"> With non-zero prefix argument ARG, the command operates on the next ARG files. Otherwise, it operates on all the marked files, or the current file if none are marked. As each match is found, the user must type a character saying what to do with it. For directions, type C–h at that time. NEWNAME may contain \<n> or \& as in ‘query-replace-regexp’. REGEXP defaults to the last regexp used. With a zero prefix arg, renaming by regexp affects the absolute file name. Normally, only the non-directory part of the file name is used and hanged. |
| Rename all marked files to lowercase names | %l | (dired-downcase &optional ARG) | Rename all marked (or next ARG) files to lower case. |
| Rename all marked files to uppercase names | %u | (dired-upcase &optional ARG) | Rename all marked (or next ARG) files to upper case. |
| Diff | | | |
| Diff file | = | (dired-diff FILE &optional SWITCHES) | Compare file at point with FILE using ‘diff’. <ul style="list-style-type: none"> Prompt for FILE, but : <ul style="list-style-type: none"> If the mark is active in Transient Mark mode, use the file at the mark as the default for FILE. (That’s the mark set by C-SPC, not by Dired’s M-x dired-mark command.) If the file at point has a backup file, use that as the default FILE. If the file at point is a backup file, use its original, if that exists and can be found. FILE is the first argument given to the ‘diff’ function (the “old” version). The file at point is the second argument given to ‘diff’. With prefix arg, prompt for second argument SWITCHES, which is the string of command switches used as the third argument of ‘diff’.  Customizations of ‘backup-directory-alist’ and ‘make-backup-file-name-function’ change where this function searches for the backup file, and affect its ability to find the original of a backup file. |
| Grep & Replace | | | |
| Grep for text in marked file(s) | A | (dired-do-find-regexp REGEXP) | Find all matches for REGEXP in all marked files. <ul style="list-style-type: none"> For any marked directory, all of its files are searched recursively. <ul style="list-style-type: none"> However, files matching ‘grep-find-ignored-files’ and subdirectories matching ‘grep-find-ignored-directories’ are skipped in the marked directories.  REGEXP should use constructs supported by your local ‘grep’ command. |
| Replace text found in marked file(s) | Q | (dired-do-find-regexp-and-replace FROM TO) | Replace matches of FROM with TO, in all marked files. <ul style="list-style-type: none"> For any marked directory, matches in all of its files are replaced, recursively. However, files matching ‘grep-find-ignored-files’ and subdirectories matching ‘grep-find-ignored-directories’ are skipped in the marked directories.  REGEXP should use constructs supported by your local ‘grep’ command. |
| Byte Compile & Load | | | |
| Byte compile marked Emacs Lisp files | B | (dired-do-byte-compile &optional ARG) | Byte compile marked (or next ARG) Emacs Lisp files. |
| Load Emacs Lisp files | L | (dired-do-load &optional ARG) | Load the marked (or next ARG) Emacs Lisp files. |
| Compress/Uncompress | | | |
| Compress/Uncompress file(s) | z | (dired-do-compress &optional ARG) | Compress or uncompress marked (or next ARG) files. <ul style="list-style-type: none"> If invoked on a directory, compress all of the files in the directory and all of its subdirectories, recursively, into a .tar.gz archive. If invoked on a .tar.gz or a .tgz or a .zip or a .7z archive, uncompress and unpack all the files in the archive. |
| Shell Commands | | | |

| Description | Keystroke | Function | Note |
|--|---|--|---|
| Execute shell command on marked files | <ul style="list-style-type: none"> ! x | (dired-do-shell-command COMMAND &optional ARG FILE-LIST) | Execute shell command in the directory. Run a shell command COMMAND on the marked files. <ul style="list-style-type: none"> If no files are marked or a numeric prefix arg is given, the next ARG files are used. Just C-u means the current file. The prompt mentions the file(s) or the marker, as appropriate. If there is a ‘*’ in COMMAND, surrounded by whitespace, this runs COMMAND just once with the entire file list substituted there. If there is no ‘*’, but there is a ‘?’ in COMMAND, surrounded by whitespace, or a ‘?’ this runs COMMAND on each file individually with the file name substituted for ‘?’ or ‘?’. Otherwise, this runs COMMAND on each file individually with the file name added at the end of COMMAND (separated by a space). When COMMAND runs, its working directory is the top-level directory of the Dired buffer, so output files usually are created there instead of in a subdir. |
| Asynchronously execute shell command on marked files | & | (dired-do-async-shell-command COMMAND &optional ARG FILE-LIST) | Like ‘dired-do-shell-command’, but adds ‘&’ at the end of COMMAND to execute it asynchronously. <ul style="list-style-type: none"> When operating on multiple files, asynchronous commands are executed in the background on each file in parallel. In shell syntax this means separating the individual commands with ‘&’. However, when COMMAND ends in ‘;’ or ‘;&’ then commands are executed in the background on each file sequentially waiting for each command to terminate before running the next command. In shell syntax this means separating the individual commands with ‘;’. The output appears in the buffer “Async Shell Command”. |
| Execute a shell command inside the current Dired directory | M-! | (dired-smart-shell-command COMMAND &optional OUTPUT-BUFFER ERROR-BUFFER) | Like function ‘shell-command’, but in the current Virtual Dired directory. <ul style="list-style-type: none"> Does not use the marked files; just uses the directory identified by Dired as the current directory for the command. For example, to execute make in the directory identified by the Dired buffer. That way you don’t need to open a shell. The disadvantage is that you’ll have to wait for the command to finish to see the output inside the “Shell Command Output” buffer. |
| Modify file attributes | | | |
| Change the file(s) group (chgrp) | G | (dired-do-chgrp &optional ARG) | Change the group of the marked (or next ARG) files. <ul style="list-style-type: none"> Type M-n to pull the file attributes of the file at point into the minibuffer. |
| Change the file(s) mode (chmod) | M | (dired-do-chmod &optional ARG) | Change the mode of the marked (or next ARG) files. <ul style="list-style-type: none"> Symbolic modes like ‘g+w’ are allowed. Type M-n to pull the file attributes of the file at point into the minibuffer. |
| Change the file(s) ownership (chown) | O | (dired-do-chown &optional ARG) | Change the owner of the marked (or next ARG) files. <ul style="list-style-type: none"> Type M-n to pull the file attributes of the file at point into the minibuffer. |
| Touch the marked file(s) | T | (dired-do-touch &optional ARG) | Change the timestamp of the marked (or next ARG) files using ‘touch’. <ul style="list-style-type: none"> Type M-n to pull the file attributes of the file at point into the minibuffer. 👉 Use this to set the timestamp of all marked files to timestamp of the current file! |
| File Hard/Soft Links | | | |
| Create hard link for current or all marked files. | H | (dired-do-hardlink &optional ARG) | Add names (hard links) for current file or all marked (or next ARG) files. <ul style="list-style-type: none"> When operating on just the current file, prompts and you specify the new name. When operating on multiple or marked files, you specify a directory and new hard links are made in that directory with the same names that the files currently have. The default suggested for the target directory depends on the value of ‘dired-dwim-target’. |
| Hardlink files selected by regexp | %H | (dired-do-hardlink-regexp REGEXP NEWNAME &optional ARG WHOLE-NAME) | Hardlink selected files whose names match REGEXP to NEWNAME. <ul style="list-style-type: none"> With non-zero prefix argument ARG, the command operates on the next ARG files. Otherwise, it operates on all the marked files, or the current file if none are marked. As each match is found, the user must type a character saying what to do with it. For directions, type C-h at that time. NEWNAME may contain \<n> or \& as in ‘query-replace-regexp’. REGEXP defaults to the last regexp used. With a zero prefix arg, renaming by regexp affects the absolute file name. Normally, only the non-directory part of the file name is used and changed. |
| Create symbolic links for current or all marked files | S | (dired-do-symlink &optional ARG) | Make symbolic links to current file or all marked (or next ARG) files. <ul style="list-style-type: none"> When operating on just the current file, you specify the new name. When operating on multiple or marked files, you specify a directory and new symbolic links are made in that directory with the same names that the files currently have. The default suggested for the target directory depends on the value of ‘dired-dwim-target’, which see. For relative symlinks, use M-x dired-do-relymlink. |
| Create relative symbolic links for current or all marked files | Y | (dired-do-relymlink &optional ARG) | Relative symlink all marked (or next ARG) files into a directory. <ul style="list-style-type: none"> Otherwise make a relative symbolic link to the current file. This creates relative symbolic links like: <pre>foo -> ../bar/foo</pre> not absolute ones like: <pre>foo -> /ugly/file/name/that/may/change/any/day/bar/foo</pre> |
| Symlink files selected by regexp | %S | (dired-do-symlink-regexp REGEXP NEWNAME &optional ARG WHOLE-NAME) | Symlink selected files whose names match REGEXP to NEWNAME. See function ‘dired-do-rename-regexp’ for more info. |
| Miscellaneous | | | |
| Print the file(s) | P | (dired-do-print &optional ARG) | Print the marked (or next ARG) files.  Uses the shell command coming from variables ‘lpr-command’ and ‘lpr-switches’ as default. |
| Visit current file as mailbox | V | (dired-do-run-mail) | Visit the current file as a mailbox, using VM or RMAIL. <ul style="list-style-type: none"> Prompt for confirmation first; if the user says yes, call ‘dired-vm’ if ‘dired-bind-vm’ is non-nil, ‘dired-rmail’ otherwise. |
| WDired — Writable Dired | When the Wdired mode is on, the file names, symlinks and file permissions (if enabled via customization) can be modified by simply editing their names as they show in the buffer. | | |
| Toggle read-only status of buffer | <ul style="list-style-type: none"> C-x C-q <f11> b r | (dired-toggle-read-only) | Toggle the WDired mode on/off. |
| | | | |
| Using dired-narrow | The external dired-narrow package provides a way to limit the number of files shown in the dired buffer by using expressions typed following one of the following commands.  The following commands require the external dired-narrow package to be installed.  This PEL bindings below are mode sensitive and are only available when the <i>pel-use-dired-narrow</i> user option is t.  To widen the list back, refresh the Dired buffer: press g. | | |
| Narrow the file list to specific filter | <f12> s | (dired-narrow) | Narrow a dired buffer to the files matching a string. (not a regexp) <ul style="list-style-type: none"> If the string contains spaces, then each word is matched against the file name separately. To succeed, all of them have to match but the order does not matter. For example "foo bar" matches filename “bar-and-foo.el”. |

| Description | Keystroke | Function | Note |
|--|-----------|-----------------------|---|
| Narrow the file list to specific regexp filter | <f12> r | (dired-narrow-regexp) | Narrow a dired buffer to the files matching a regular expression. • For example, to list all .el files, use: .e1\$ |
| Narrow the file list to specific fuzzy filter | <f12> f | (dired-narrow-fuzzy) | Narrow a dired buffer to the files matching a fuzzy string. A fuzzy string is constructed from the filter string by inserting "." between each letter. This is then matched as regular expression against the file name. |

Dired — References

| Description & link | Notes |
|---|--|
| Dired @ wikipedia | An overview. |
| Dired in Emacs Manual | Dired table of Content in the emacs manual. |
| Mike Sperber's dired page | Page of Dired current maintainer |
| Emacs: techniques to narrow Dired - Youtube video | A quick video on how to use straight dired-mode to list a sub-set of all files. It also describes the dired-narrow package though. |
| Integrating OS X and Emacs Dired | Jason Blevins describes how to launch macOS registered application for a specific file in Dired mode. |
| | |
| | |