Diff & Merge

Operation	Keystroke	Function	Note
Comparing and Merging Files See also: ∑ **Customize**	Emacs has two built-in packages that support comparing files: diff and ediff. • The diff package is the simplest of the two. The second, ediff is more powerful and also support file merge. 📦 The **ztree external package** provides a tree-view directory diff. 🔣 PEL activates it when **pel-use-ztree** is set to **t**.		
Open this PDF file. See also: ∑ **Help/Info**	`<f11> d <f1>`	(**pel-help-pdf** &optional OPEN-WEB-PAGE)	Open the ∑ **Diff & Merge** local PDF. If the prefix argument (like `C-u` or `M--`) is used, then it opens the remote GitHub hosted raw PDF instead. If the **pel-flip-help-pdf-arg** user-option is set it's the other way around.
Customize PEL support for diff	`<f11> d <f2>`	(**pel-customize-pel** &optional OTHER-WINDOW)	Customize PEL support for diff: ztree. • If OTHER-WINDOW is non-nil (use `C-u`), display in other window.
Customize Emacs support for diff & ediff	`<f11> d <f3>`	(**pel-customize-library** &optional OTHER-WINDOW)	Customize Emacs support for diff, ediff, ztree. • If OTHER-WINDOW is non-nil (use `C-u`), display in other window.
Customize Emacs support for ediff	`<f11> d e <f3>`	(**pel-customize-library** &optional OTHER-WINDOW)	Customize Emacs ediff. • If OTHER-WINDOW is non-nil (use `C-u`), display in other window.
Diff	The diff mode provides a set of commands to compare files and buffers using the simple diff-mode.		
Compare 2 files	`<f11> d f`	(**diff** OLD NEW &optional SWITCHES NO-ASYNC)	Find and display the differences between OLD and NEW files. • Prompt for NEW, then OLD files.
Compare file with its backup	`<f11> d k`	(**diff-backup** FILE &optional SWITCHES)	Diff this file with its backup file or vice versa. • Uses the latest backup, if there are several numerical backups. • If this file is a backup, diff it with its original. • The backup file is the first file given to 'diff'. • With prefix arg, prompt for diff switches.
Compare buffer and associated file	`<f11> d b`	(**diff-buffer-with-file** &optional BUFFER)	View the differences between BUFFER and its associated file.
Compare current and other window	`<f11> d w`	(**compare-windows** IGNORE-WHITESPACE)	Compare text in current window with text in another window.
	• The **compare-windows-get-window-function** user option defines how to get another window. By default it is set to compare-windows-get-recent-window. • Compares the text starting at point in each window, moving over text in each one as far as they match. • This command pushes the mark in each window at the prior location of point in that window. • If both windows display the same buffer, the mark is pushed twice in that buffer: first in the other window, then in the selected window. 👆 Use this for simple comparison between 2 windows just to see if they have the same content. • Split your frame in 2 windows, loaded with the buffer of the files you want to compare. • Place point at the top of each buffer and issue the command. • Point will be moved to the first difference in both window. If there is no difference point will be moved at the end of each window.		
Ediff	Ediff sessions have several commands, shown in the Ediff Quick Help buffer. Type **?** to toggle Ediff Quick Help from 1 line to multiple that shows all available commands. For more info see Ediff Quick Help Commands. 👆 While showing 2 or 3 buffer/files in windows Ediff provides the v/V keys for scrolling up/down. PEL scroll sync commands can also be used to provide single line scroll synced between the windows. See the `<f11> |` commands in ∑ **Scrolling** table.		
Display Ediff Manual	`<f11> d e ?`	(**ediff-documentation** &optional NODE)	Display Ediff's manual. • With optional NODE, goes to that node.
Display registry of active Ediff sessions	`<f11> d e R`	• (**eregistry**) • (**ediff-show-registry**)	Display registry of all active Ediff sessions.
Ediff file against previous revision	`<f11> d r`	(**pel-ediff-revisions**)	Run ediff-revision on the file in the current window. • Prompts for revisions, default to current copy and last commit.
Ediff 2 files	`<f11> d 2`	(**pel-ediff-2files** &optional N)	Run ediff-files on the files of current and the other window. • Select the current file and the other file without prompting. • If a numeric argument is specified, and if N is in [2,8] range, select the other window identified by the direction corresponding to the cursor in a numeric keypad: 8 := 'up 4 := 'left 5 := 'current 6 := 'right 2 := 'down
Compare buffer with its file on disk	`<f11> d e b f`	(**ediff-current-file**)	Compare the buffer with its file on disk. This function can be used as a safe version of revert-buffer.
Compare 2 buffers	`<f11> d e b b`	(**ediff-buffers** BUFFER-A BUFFER-B &optional STARTUP-HOOKS JOB-NAME)	Compare 2 buffers. • Prompts for buffer A and buffer B
Compare 3 buffers	`<f11> d e b 3`	(**ediff-buffers3** BUFFER-A BUFFER-B BUFFER-C &optional STARTUP-HOOKS JOB-NAME)	Compare 3 buffers. • Prompts for buffer A, buffer B and buffer C.
Compare file with its backup ∑ Autosave/backup	`<f11> d e f k`	(**ediff-backup** FILE)	Compare a file with its backup. If there are several numerical backups, use the latest. • If the file is itself a backup, then compare it with its original.
Compare 2 files	`<f11> d e f f`	• (**ediff** FILE-A FILE-B &optional STARTUP-HOOKS) • (**ediff-files** FILE-A FILE-B &optional STARTUP-HOOKS)	Compare 2 files. • Prompts for file A and B. • PEL provide a shortcut function **pel-ediff-2files** mapped to `<f11> d 2` which does not prompt. See above.
Compare 3 files	`<f11> d e f 3`	• (**ediff3** FILE-A FILE-B FILE-C &optional STARTUP-HOOKS) • (**ediff-files3** FILE-A FILE-B FILE-C &optional STARTUP-HOOKS)	Compare 3 files. • Prompts for file A, B and C.
Compare revision of buffer with file revision	`<f11> d e f r`	(**ediff-revision** &optional FILE STARTUP-HOOKS)	Compare versions of the current buffer, if the buffer is visiting a file under version control. • Prompts for the file name and each of its revisions. • PEL provide a shortcut function **pel-ediff-revision** mapped to `<f11> d r`.
Compare versions of files in a given directory	`<f11> d e d r`	• (**edir-revisions** DIR1 REGEXP) • (**ediff-directory-revisions** DIR1 REGEXP)	Compare versions of files in a given directory. • Ediff selects only the files that are under version control. • Prompts for directory and regexp to identify files: if empty: selects all files.
Compare text in 2 windows word-by-word	`<f11> d e w w`	(**ediff-windows-wordwise** DUMB-MODE &optional WIND-A WIND-B STARTUP-HOOKS)	Compare text visible in 2 windows word-by-word. • Uses current and other (next) window.
Compare text in 2 windows line-by-line	`<f11> d e w l`	(**ediff-windows-linewise** DUMB-MODE &optional WIND-A WIND-B STARTUP-HOOKS)	Compare text visible in 2 windows line-by-line. • Uses current and other (next) window.
Compare 2 regions word-by-word	`<f11> d e r w`	(**ediff-regions-wordwise** BUFFER-A BUFFER-B &optional STARTUP-HOOKS)	Compare text visible in 2 regions word-by-word. • Prompts for the 2 buffers and regions.
Compare 2 regions line-by-line	`<f11> d e r l`	(**ediff-regions-linewise** BUFFER-A BUFFER-B &optional STARTUP-HOOKS)	Compare text visible in 2 regions line-by-line. • Prompts for the 2 buffers and regions.

Operation	Keystroke	Function	Note
Patch file(s) and compare	`<f11> d e p f`	• (**epatch** &optional ARG PATCH-BUF) • (**ediff-patch-file** &optional ARG PATCH-BUF)	Query for a file name, and then run Ediff by patching that file. • If optional PATCH-BUF is given, use the patch in that buffer and don't ask the user. • If prefix argument ARG, then: if even argument, assume that the patch is in a buffer. If odd -- assume it is in a file.
Patch a buffer then compare	`<f11> d e p b`	• (**epatch-buffer** &optional ARG PATCH-BUF) • (**ediff-patch-buffer** &optional ARG PATCH-BUF)	Run Ediff by patching the buffer specified at prompt. • Without the optional prefix ARG, asks if the patch is in some buffer and prompts for the buffer or a file, depending on the answer. • With ARG=1, assumes the patch is in a file and prompts for the file. • With ARG=2, assumes the patch is in a buffer and prompts for the buffer. • PATCH-BUF is an optional argument, which specifies the buffer that contains the patch. If not given, the user is prompted according to the prefix argument.
Compare Directories		The built-in Ediff can compare 2 or 3 directories. 📦The **ztree external package** provides a tree-view directory diff. 🔲 PEL activates it when pel-use-ztree is set to **t**.	
Compare common files in 2 directories	`<f11> d e d d`	• (**edirs** DIR1 DIR2 REGEXP) • (**ediff-directories** DIR1 DIR2 REGEXP)	Compare files common to two directories. • Prompts for directory A and B and a regexp to identify files. If empty: select all files.
Compare common files in 3 directories	`<f11> d e d 3`	• (**edirs3** DIR1 DIR2 DIR3 REGEXP) • (**ediff-directories3** DIR1 DIR2 DIR3 REGEXP)	Compare files common to three directories. • Prompts for directory A, B and C and a regexp to identify files: if empty: selects all files.
Compare 2 directories with ztree-diff	`<f11> d z`	(**ztree-diff** DIR1 DIR2)	Open an interactive buffer with the directory tree of the path given, and highlight of file differences between the directories. • Argument DIR1 left directory. Interactively: prompts. • Argument DIR2 right directory. Interactively: prompts. 📦 Requires the ztree external package. 🔲 PEL activates it if **pel-use-ztree** is **t**.

• **Ediff 2 files** ☞		In the Ztree Diff buffer the following keys are available:
		• **<SPACE>** : Open/close directory
		• **<RET>** : Open/close directory | **Ediff 2 files** /open orphan file
		• **<TAB>** : switch between panels
		• **C** : copy current file/directory to the directory shown in the other side (prompts)
		• **D** : delete current file/directory (prompts)
		• **h** : toggle display of identical files/directories.
		• **H** : toggle display of filtered files.
		• **r** : rescan/refresh current file/directory
		• **v** : visit current file in view-mode .
		    • In view-mode the following keys are available (see ∑ **Buffers** for a complete list):
		      • q: quit view-mode and return to Ztree Diff
		      • e: leave view-mode, edit/visit the file normally.
		• **x** : Toggle expand/collapse of all nodes of the subtree.
		    ⚠️ Use **x** with care! On large directory trees it may take a long time. I have see Emacs hang when typing **x** again during that time.
		• **<f5>** : force full rescan, re-write buffer (useful when changing window size).	

Merge			
Merge 2 files	`<f11> d e m f`	• (**ediff-merge** FILE-A FILE-B &optional STARTUP-HOOKS MERGE-BUFFER-FILE) • (**ediff-merge-files** FILE-A FILE-B &optional STARTUP-HOOKS MERGE-BUFFER-FILE)	Merge two files without ancestor. • Prompt for FILE-A and FILE-B, the names of the files to be merged.
Merge 2 files with ancestor	`<f11> d e m F`	• (**ediff-merge-with-ancestor** FILE-A FILE-B FILE-ANCESTOR &optional STARTUP-HOOKS MERGE-BUFFER-FILE) • (ediff-merge-files-with-ancestor FILE-A FILE-B FILE-ANCESTOR &optional STARTUP-HOOKS MERGE-BUFFER-FILE)	Merge two files with ancestor. • Prompt for FILE-A and FILE-B, the names of the files to be merged, and FILE-ANCESTOR, the name of the ancestor file.
Merge 2 buffers	`<f11> d e m b`	(**ediff-merge-buffers** BUFFER-A BUFFER-B &optional STARTUP-HOOKS JOB-NAME MERGE-BUFFER-FILE)	Merge buffers without ancestor. • Prompt for BUFFER-A and BUFFER-B, the buffers to be merged.
Merge 2 buffers with ancestor	`<f11> d e m B`	(**ediff-merge-buffers-with-ancestor** BUFFER-A BUFFER-B BUFFER-ANCESTOR &optional STARTUP-HOOKS JOB-NAME MERGE-BUFFER-FILE)	Merge buffers with ancestor. • Prompts for BUFFER-A and BUFFER-B, the buffers to be merged, and BUFFER-ANCESTOR, their ancestor.
Merge versions of files in a directory	`<f11> d e m d`	• (**edir-merge-revisions** DIR1 REGEXP &optional MERGE-AUTOSTORE-DIR) • (**ediff-merge-directory-revisions** DIR1 REGEXP &optional MERGE-AUTOSTORE-DIR)	Merge versions of files in a given directory. • Ediff selects only the files that are under version control. • Prompts for directory and regexp to identify files: if empty: selects all files.
Merge versions of files in a directory using other versions as their ancestors	`<f11> d e m D`	• (**edir-merge-revisions-with-ancestor** DIR1 REGEXP &optional MERGE-AUTOSTORE-DIR) • (**ediff-merge-directory-revisions-with-ancestor** DIR1 REGEXP &optional • MERGE-AUTOSTORE-DIR)	Merge versions of files in a given directory using other versions as ancestors. • Ediff selects only the files that are under version control. • Prompts for directory and regexp to identify files: if empty: selects all files.
Merge file commons to 2 directories	`<f11> d e m c`	• (**edirs-merge** DIR1 DIR2 REGEXP &optional MERGE-AUTOSTORE-DIR) • (**ediff-merge-directories** DIR1 DIR2 REGEXP &optional MERGE-AUTOSTORE-DIR)	Merge files common to two directories. • Run Ediff on a pair of directories, DIR1 and DIR2, merging files that have the same name in both. • The third argument, REGEXP, is nil or a regular expression; only file names that match the regexp are considered. • MERGE-AUTOSTORE-DIR is the directory in which to store merged files.
Merge file commons to 2 directories with ancestors	`<f11> d e m C`	• (**edirs-merge-with-ancestor** DIR1 DIR2 ANCESTOR-DIR REGEXP &optional MERGE-AUTOSTORE-DIR) • (**ediff-merge-directories-with-ancestor** DIR1 DIR2 ANCESTOR-DIR REGEXP &optional MERGE-AUTOSTORE-DIR)	Merge files in directories DIR1 and DIR2 using files in ANCESTOR-DIR as ancestors. • Ediff merges files that have identical names in DIR1, DIR2. • If a pair of files in DIR1 and DIR2 doesn't have an ancestor in ANCESTOR-DIR, Ediff will merge without ancestor. • The fourth argument, REGEXP, is nil or a regular expression; only file names that match the regexp are considered.
Merge 2 versions of visited file	`<f11> d e m r`	(**ediff-merge-revisions** &optional FILE STARTUP-HOOKS MERGE-BUFFER-FILE)	Merge two versions of the file visited by the current buffer.
Merge 2 versions of visited file with ancestor	`<f11> d e m R`	(**ediff-merge-revisions-with-ancestor** &optional FILE STARTUP-HOOKS MERGE-BUFFER-FILE)	Merge two versions of the file visited by the current buffer with ancestor.
Specialized Ediff		Some packages provide specialization of Ediff-based comparisons.	
ParInfer EDiff **Diff current code before/.after ParInfer modifications** 📜 - Emacs Lisp	• `<f12> a D` • `<f11> SPC l a D`	(**parinfer-diff**)	Diff current code and the code after applying Indent Mode in Ediff. Use this to browse and apply the changes. ☞📦 Requires the parinfer package. ☞🔲 PEL activates this when the **pel-use-parinfer** user option is set to **t**.

DIff & Merge — References

emacs vdiff github project	
Using Emacs as a merge program @ Mercurial Wiki	
GNU Emacs Ediff Manual	Describes built-in Ediff: file diff of 2 or 3 files, 3-way merge, diff with backup, directory level diffs, version control aware…
vdiff.el	Github home of vdiff.el, a package that implements a diff tool similar to vimdiff.