









## Diff & Merge

Operation	Keystroke	Function	Note
<b>Comparing and Merging Files</b> See also: <a href="#">🔗 Customize</a>	Emacs has two built-in packages that support comparing files: diff and ediff. <ul style="list-style-type: none"> <li>The diff package is the simplest of the two. The second, ediff is more powerful and also support file merge.</li> </ul>  The <a href="#">ztree external package</a> provides a tree-view directory diff.  PEL activates it when <b>pel-use-ztree</b> is set to <b>t</b> .		
<b>Open this PDF file.</b> See also: <a href="#">🔗 Help/Info</a>	<b>&lt;f11&gt; d &lt;f1&gt;</b>	<b>(pel-help-pdf</b> &optional OPEN-WEB-PAGE)	Open the <a href="#">🔗 Diff &amp; Merge</a> local PDF. If the prefix argument (like <b>C-u</b> or <b>M--</b> ) is used, then it opens the remote GitHub hosted raw PDF instead. If the <b>pel-flip-help-pdf-arg</b> user-option is set it's the other way around.
<b>Customize PEL support for diff</b>	<b>&lt;f11&gt; d &lt;f2&gt;</b>	<b>(pel-customize-pel</b> &optional OTHER-WINDOW)	Customize PEL support for diff: ztree. <ul style="list-style-type: none"> <li>If OTHER-WINDOW is non-nil (use <b>C-u</b>) , display in other window.</li> </ul>
<b>Customize Emacs support for diff &amp; ediff</b>	<b>&lt;f11&gt; d &lt;f3&gt;</b>	<b>(pel-customize-library</b> &optional OTHER-WINDOW)	Customize Emacs support for diff, ediff, ztree. <ul style="list-style-type: none"> <li>If OTHER-WINDOW is non-nil (use <b>C-u</b>) , display in other window.</li> </ul>
<b>Customize Emacs support for ediff</b>	<b>&lt;f11&gt; d e &lt;f3&gt;</b>	<b>(pel-customize-library</b> &optional OTHER-WINDOW)	Customize Emacs ediff. <ul style="list-style-type: none"> <li>If OTHER-WINDOW is non-nil (use <b>C-u</b>) , display in other window.</li> </ul>
<b>Diff</b>	The diff mode provides a set of commands to compare files and buffers using the simple <a href="#">diff-mode</a> .		
<b>Compare 2 files</b>	<b>&lt;f11&gt; d f</b>	<b>(diff OLD NEW</b> &optional SWITCHES NO-ASYNC)	Find and display the differences between OLD and NEW files. <ul style="list-style-type: none"> <li>Prompt for NEW, then OLD files.</li> </ul>
<b>Compare file with its backup</b>	<b>&lt;f11&gt; d k</b>	<b>(diff-backup FILE</b> &optional SWITCHES)	Diff this file with its backup file or vice versa. <ul style="list-style-type: none"> <li>Uses the latest backup, if there are several numerical backups.</li> <li>If this file is a backup, diff it with its original.</li> <li>The backup file is the first file given to 'diff'.</li> <li>With prefix arg, prompt for diff switches.</li> </ul>
<b>Compare buffer and associated file</b>	<b>&lt;f11&gt; d b</b>	<b>(diff-buffer-with-file</b> &optional BUFFER)	View the differences between BUFFER and its associated file.
<b>Compare current and other window</b>	<b>&lt;f11&gt; d w</b>	<b>(compare-windows IGNORE-WHITESPACE)</b>	Compare text in current window with text in another window. <ul style="list-style-type: none"> <li>The <b>compare-windows-get-window-function</b> user option defines how to get another window. By default it is set to compare-windows-get-recent-window.</li> <li>Compares the text starting at point in each window, moving over text in each one as far as they match.</li> <li>This command pushes the mark in each window at the prior location of point in that window.</li> <li>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.</li> <li>👉 Use this for simple comparison between 2 windows just to see if they have the same content.</li> <li>Split your frame in 2 windows, loaded with the buffer of the files you want to compare.</li> <li>Place point at the top of each buffer and issue the command.</li> <li>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.</li> </ul>
<b>Ediff</b>  See also: <a href="#">🔗 Scrolling</a>	<b>Ediff sessions</b> have several commands, shown in the Ediff Quick Help buffer. <ul style="list-style-type: none"> <li>Type <b>?</b> to toggle Ediff Quick Help from 1 line to multiple that shows all available commands.</li> <li>For more info about a command, place point on the command character and type <b>RET</b>: Emacs will open the <a href="#">Ediff Quick Help Commands</a> Info buffer.</li> </ul> 👉 While showing 2 or 3 buffer/files in windows Ediff provides the <b>▼/▼</b> keys for scrolling up/down. PEL scroll sync commands can also be used to provide single line scroll synced between the windows. See the <a href="#">&lt;f11&gt;   commands in table</a> .		
<b>Display Ediff Manual</b>	<b>&lt;f11&gt; d e ?</b>	<b>(ediff-documentation</b> &optional NODE)	Display Ediff's manual. <ul style="list-style-type: none"> <li>With optional NODE, goes to that node.</li> </ul>
<b>Display registry of active Ediff sessions</b>	<b>&lt;f11&gt; d e R</b>	<ul style="list-style-type: none"> <li><b>(eregistry)</b></li> <li><b>(ediff-show-registry)</b></li> </ul>	Display registry of all active Ediff sessions.
<b>Ediff file against previous revision</b>	<b>&lt;f11&gt; d r</b>	<b>(pel-ediff-revisions)</b>	Run ediff-revision on the file in the current window. <ul style="list-style-type: none"> <li>Prompts for revisions, default to current copy and last commit.</li> </ul>
<b>Ediff 2 files</b>	<b>&lt;f11&gt; d 2</b>	<b>(pel-ediff-2files</b> &optional N)	Run ediff-files on the files of current and the other window. <ul style="list-style-type: none"> <li>Select the current file and the other file without prompting.</li> <li>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:                             <div>                                 8 := 'up                                  4 := 'left 5 := 'current 6 := 'right                                  2 := 'down                             </div> </li> </ul>
<b>Compare buffer with its file on disk</b>	<b>&lt;f11&gt; d e b f</b>	<b>(ediff-current-file)</b>	Compare the buffer with its file on disk. This function can be used as a safe version of revert-buffer.
<b>Compare 2 buffers</b>	<b>&lt;f11&gt; d e b b</b>	<b>(ediff-buffers BUFFER-A BUFFER-B</b> &optional STARTUP-HOOKS JOB-NAME)	Compare 2 buffers. <ul style="list-style-type: none"> <li>Prompts for buffer A and buffer B</li> </ul>
<b>Compare 3 buffers</b>	<b>&lt;f11&gt; d e b 3</b>	<b>(ediff-buffers3 BUFFER-A BUFFER-B BUFFER-C</b> &optional STARTUP-HOOKS JOB-NAME)	Compare 3 buffers. <ul style="list-style-type: none"> <li>Prompts for buffer A, buffer B and buffer C.</li> </ul>
<b>Compare file with its backup</b> <a href="#">🔗 Autosave/backup</a>	<b>&lt;f11&gt; d e f k</b>	<b>(ediff-backup FILE)</b>	Compare a file with its backup. If there are several numerical backups, use the latest. <ul style="list-style-type: none"> <li>If the file is itself a backup, then compare it with its original.</li> </ul>
<b>Compare 2 files</b>	<b>&lt;f11&gt; d e f f</b>	<ul style="list-style-type: none"> <li><b>(ediff FILE-A FILE-B</b> &amp;optional STARTUP-HOOKS)</li> <li><b>(ediff-files FILE-A FILE-B</b> &amp;optional STARTUP-HOOKS)</li> </ul>	Compare 2 files. <ul style="list-style-type: none"> <li>Prompts for file A and B.</li> <li>PEL provide a shortcut function <b>pel-ediff-2files</b> mapped to <b>&lt;f11&gt; d 2</b> which does not prompt. See above.</li> </ul>
<b>Compare 3 files</b>	<b>&lt;f11&gt; d e f 3</b>	<ul style="list-style-type: none"> <li><b>(ediff3 FILE-A FILE-B FILE-C</b> &amp;optional STARTUP-HOOKS)</li> <li><b>(ediff-files3 FILE-A FILE-B FILE-C</b> &amp;optional STARTUP-HOOKS)</li> </ul>	Compare 3 files. <ul style="list-style-type: none"> <li>Prompts for file A, B and C.</li> </ul>
<b>Compare revision of buffer with file revision</b>	<b>&lt;f11&gt; d e f r</b>	<b>(ediff-revision</b> &optional FILE STARTUP-HOOKS)	Compare versions of the current buffer, if the buffer is visiting a file under version control. <ul style="list-style-type: none"> <li>Prompts for the file name and each of its revisions.</li> <li>PEL provide a shortcut function <b>pel-ediff-revision</b> mapped to <b>&lt;f11&gt; d r</b>.</li> </ul>
<b>Compare versions of files in a given directory</b>	<b>&lt;f11&gt; d e d r</b>	<ul style="list-style-type: none"> <li><b>(edir-revisions DIR1 REGEXP)</b></li> <li><b>(ediff-directory-revisions DIR1 REGEXP)</b></li> </ul>	Compare versions of files in a given directory. <ul style="list-style-type: none"> <li>Ediff selects only the files that are under version control.</li> <li>Prompts for directory and regexp to identify files: if empty: selects all files.</li> </ul>
<b>Compare text in 2 windows word-by-word</b>	<b>&lt;f11&gt; d e w w</b>	<b>(ediff-windows-wordwise DUMB-MODE</b> &optional WIND-A WIND-B STARTUP-HOOKS)	Compare text visible in 2 windows word-by-word. <ul style="list-style-type: none"> <li>Uses current and other (next) window.</li> </ul>
<b>Compare text in 2 windows line-by-line</b>	<b>&lt;f11&gt; d e w l</b>	<b>(ediff-windows-linewise DUMB-MODE</b> &optional WIND-A WIND-B STARTUP-HOOKS)	Compare text visible in 2 windows line-by-line. <ul style="list-style-type: none"> <li>Uses current and other (next) window.</li> </ul>
<b>Compare 2 regions word-by-word</b>	<b>&lt;f11&gt; d e r w</b>	<b>(ediff-regions-wordwise BUFFER-A BUFFER-B</b> &optional STARTUP-HOOKS)	Compare text visible in 2 regions word-by-word. <ul style="list-style-type: none"> <li>Prompts for the 2 buffers and regions.</li> </ul>

Operation	Keystroke	Function	Note
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. <ul style="list-style-type: none"> <li>Prompts for the 2 buffers and regions.</li> </ul>
Patch file(s) and compare	<f11> d e p f	<ul style="list-style-type: none"> <li>(epatch &amp;optional ARG PATCH-BUF)</li> <li>(ediff-patch-file &amp;optional ARG PATCH-BUF)</li> </ul>	Query for a file name, and then run Ediff by patching that file. <ul style="list-style-type: none"> <li>If optional PATCH-BUF is given, use the patch in that buffer and don't ask the user.</li> <li>If prefix argument ARG, then: if even argument, assume that the patch is in a buffer. If odd -- assume it is in a file.</li> </ul>
Patch a buffer then compare	<f11> d e p b	<ul style="list-style-type: none"> <li>(epatch-buffer &amp;optional ARG PATCH-BUF)</li> <li>(ediff-patch-buffer &amp;optional ARG PATCH-BUF)</li> </ul>	Run Ediff by patching the buffer specified at prompt. <ul style="list-style-type: none"> <li>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.</li> <li>With ARG=1, assumes the patch is in a file and prompts for the file.</li> <li>With ARG=2, assumes the patch is in a buffer and prompts for the buffer.</li> <li>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.</li> </ul>
Compare Directories	The built-in Ediff can compare 2 or 3 directories.  The <a href="#">ztree external package</a> provides a tree-view directory diff.  PEL activates it when <a href="#">pel-use-ztree</a> is set to <a href="#">t</a> .		
Compare common files in 2 directories	<f11> d e d d	<ul style="list-style-type: none"> <li>(edirs DIR1 DIR2 REGEXP)</li> <li>(ediff-directories DIR1 DIR2 REGEXP)</li> </ul>	Compare files common to two directories. <ul style="list-style-type: none"> <li>Prompts for directory A and B and a regexp to identify files. If empty: select all files.</li> </ul>
Compare common files in 3 directories	<f11> d e d 3	<ul style="list-style-type: none"> <li>(edirs3 DIR1 DIR2 DIR3 REGEXP)</li> <li>(ediff-directories3 DIR1 DIR2 DIR3 REGEXP)</li> </ul>	Compare files common to three directories. <ul style="list-style-type: none"> <li>Prompts for directory A, B and C and a regexp to identify files: if empty: selects all files.</li> </ul>
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. <ul style="list-style-type: none"> <li>Argument DIR1 left directory. Interactively: prompts.</li> <li>Argument DIR2 right directory. Interactively: prompts.</li> </ul>  Requires the <a href="#">ztree external package</a> .  PEL activates it if <a href="#">pel-use-ztree</a> is <a href="#">t</a> .
<ul style="list-style-type: none"> <li>Ediff 2 files </li> </ul>	In the Ztree Diff buffer the following keys are available: <ul style="list-style-type: none"> <li>&lt;SPACE&gt; : Open/close directory</li> <li>&lt;RET&gt; : Open/close directory   <b>Ediff 2 files</b> /open orphan file</li> <li>&lt;TAB&gt; : switch between panels</li> <li>C : copy current file/directory to the directory shown in the other side (prompts)</li> <li>D : delete current file/directory (prompts)</li> <li>h : toggle display of identical files/directories.</li> <li>H : toggle display of filtered files.</li> <li>r : rescan/refresh current file/directory</li> <li>v : visit current file in <i>view-mode</i> . <ul style="list-style-type: none"> <li>In view-mode the following keys are available (see <a href="#">Z Buffers</a> for a complete list):</li> <li>q: quit view-mode and return to Ztree Diff</li> <li>e: leave view-mode, edit/visit the file normally.</li> </ul> </li> <li>x : Toggle expand/collapse of all nodes of the subtree. <ul style="list-style-type: none"> <li> Use <b>x</b> with care! On large directory trees it may take a long time. I have see Emacs hang when typing <b>x</b> again during that time.</li> </ul> </li> <li>&lt;f5&gt; : force full rescan, re-write buffer (useful when changing window size).</li> </ul>		
Merge			
Merge 2 files	<f11> d e m f	<ul style="list-style-type: none"> <li>(ediff-merge FILE-A FILE-B &amp;optional STARTUP-HOOKS MERGE-BUFFER-FILE)</li> <li>(ediff-merge-files FILE-A FILE-B &amp;optional STARTUP-HOOKS MERGE-BUFFER-FILE)</li> </ul>	Merge two files without ancestor. <ul style="list-style-type: none"> <li>Prompt for FILE-A and FILE-B, the names of the files to be merged.</li> </ul>
Merge 2 files with ancestor	<f11> d e m F	<ul style="list-style-type: none"> <li>(ediff-merge-with-ancestor FILE-A FILE-B FILE-ANCESTOR &amp;optional STARTUP-HOOKS MERGE-BUFFER-FILE)</li> <li>(ediff-merge-files-with-ancestor FILE-A FILE-B FILE-ANCESTOR &amp;optional STARTUP-HOOKS MERGE-BUFFER-FILE)</li> </ul>	Merge two files with ancestor. <ul style="list-style-type: none"> <li>Prompt for FILE-A and FILE-B, the names of the files to be merged, and FILE-ANCESTOR, the name of the ancestor file.</li> </ul>
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. <ul style="list-style-type: none"> <li>Prompt for BUFFER-A and BUFFER-B, the buffers to be merged.</li> </ul>
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. <ul style="list-style-type: none"> <li>Prompts for BUFFER-A and BUFFER-B, the buffers to be merged, and BUFFER-ANCESTOR, their ancestor.</li> </ul>
Merge versions of files in a directory	<f11> d e m d	<ul style="list-style-type: none"> <li>(edir-merge-revisions DIR1 REGEXP &amp;optional MERGE-AUTOSTORE-DIR)</li> <li>(ediff-merge-directory-revisions DIR1 REGEXP &amp;optional MERGE-AUTOSTORE-DIR)</li> </ul>	Merge versions of files in a given directory. <ul style="list-style-type: none"> <li>Ediff selects only the files that are under version control.</li> <li>Prompts for directory and regexp to identify files: if empty: selects all files.</li> </ul>
Merge versions of files in a directory using other versions as their ancestors	<f11> d e m D	<ul style="list-style-type: none"> <li>(edir-merge-revisions-with-ancestor DIR1 REGEXP &amp;optional MERGE-AUTOSTORE-DIR)</li> <li>(ediff-merge-directory-revisions-with-ancestor DIR1 REGEXP &amp;optional MERGE-AUTOSTORE-DIR)</li> </ul>	Merge versions of files in a given directory using other versions as ancestors. <ul style="list-style-type: none"> <li>Ediff selects only the files that are under version control.</li> <li>Prompts for directory and regexp to identify files: if empty: selects all files.</li> </ul>
Merge file commons to 2 directories	<f11> d e m c	<ul style="list-style-type: none"> <li>(edirs-merge DIR1 DIR2 REGEXP &amp;optional MERGE-AUTOSTORE-DIR)</li> <li>(ediff-merge-directories DIR1 DIR2 REGEXP &amp;optional MERGE-AUTOSTORE-DIR)</li> </ul>	Merge files common to two directories. <ul style="list-style-type: none"> <li>Run Ediff on a pair of directories, DIR1 and DIR2, merging files that have the same name in both.</li> <li>The third argument, REGEXP, is nil or a regular expression; only file names that match the regexp are considered.</li> <li>MERGE-AUTOSTORE-DIR is the directory in which to store merged files.</li> </ul>
Merge file commons to 2 directories with ancestors	<f11> d e m C	<ul style="list-style-type: none"> <li>(edirs-merge-with-ancestor DIR1 DIR2 ANCESTOR-DIR REGEXP &amp;optional MERGE-AUTOSTORE-DIR)</li> <li>(ediff-merge-directories-with-ancestor DIR1 DIR2 ANCESTOR-DIR REGEXP &amp;optional MERGE-AUTOSTORE-DIR)</li> </ul>	Merge files in directories DIR1 and DIR2 using files in ANCESTOR-DIR as ancestors. <ul style="list-style-type: none"> <li>Ediff merges files that have identical names in DIR1, DIR2.</li> <li>If a pair of files in DIR1 and DIR2 doesn't have an ancestor in ANCESTOR-DIR, Ediff will merge without ancestor.</li> <li>The fourth argument, REGEXP, is nil or a regular expression; only file names that match the regexp are considered.</li> </ul>
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.

Operation	Keystroke	Function	Note
Specialized Ediff	Some packages provide specialization of Ediff-based comparisons.		
ParInfer EDiff Diff current code before/.after ParInfer modifications x39l - Emacs Lisp	<ul style="list-style-type: none"> <li>• &lt;f12&gt; a D</li> <li>• &lt;f11&gt; SPC l</li> <li>  a D</li> </ul>	(parinfer-diff)	Diff current code and the code after applying Indent Mode in Ediff. Use this to browse and apply the changes. Requires the <a href="#">parinfer</a> package. ⚠️ This is an obsolete package. PEL activates this when the <b>pel-use-parinfer</b> user option is set to <b>t</b> .

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.