

Ediff Mode

Because [Ediff](#) is a very useful *Emacs* mode (in particular for VC) i wrote a shell script to start it from the command line.

```

1. #!/bin/bash
2. #
3. # Emacs EDIFF shell script.
4. # NOTE: This script was tested with bash, dash (Debian 12) and zsh
   (macOS 26+).
5. #
6. # Usage: ediff.sh [--client] LOCAL REMOTE [MERGED [BASE]]
7. #
8. # Copyright (C) 2013-2026 Ralf Hoppe <ralf@rho62.de>
9. #
10.
11. abspath ()
12. {
13.     [[ $1 = /* ]] && echo "$1" || echo "$PWD/${1#./}"
14. }
15.
16.
17. # if this is really a MERGE
18. MERGE=1
19.
20. if [ $# -gt 0 ] && [ "$1" = "--client" ] ; then
21.     LOCAL=$(abspath "$2")
22.     REMOTE=$(abspath "$3")
23.     MERGED=$(abspath "$4")
24.     BASE=$(abspath "$5")
25.     EMACS_CMD="emacsclient --create-frame"
26.
27.     # NOTE: Empty list `vc-handled-backends' is a workaround for Emacs
   24
28.     #      bug #18788, regarding vc-git mode-line.
29.     ELISP_COMMON=\
30. "(setq rho/vc-handled-backends-backup vc-handled-backends      \
31.   rho/sh-ediff-buf-local (get-file-buffer \"${LOCAL}\")      \
32.   rho/sh-ediff-buf-remote (get-file-buffer \"${REMOTE}\")    \
33.   rho/sh-ediff-frame (selected-frame)                        \
34.   cscope-initial-directory nil                               \
35.   vc-handled-backends ())                                    \
36. (defun rho/sh-ediff-quit-hook ()                               \
37.   (setq vc-handled-backends rho/vc-handled-backends-backup) \
38.   (ignore-errors                                             \
39.     (unless rho/sh-ediff-buf-local                            \
40.       (kill-buffer (get-file-buffer \"${LOCAL}\"))          \
41.     (unless rho/sh-ediff-buf-remote

```

```

42.         (kill-buffer (get-file-buffer \ "${REMOTE}\")) \
43.     (remove-hook 'ediff-quit-hook 'rho/sh-ediff-quit-hook) \
44.     (when (frame-live-p rho/sh-ediff-frame) \
45.         (delete-frame rho/sh-ediff-frame)) \
46. (add-hook 'ediff-quit-hook 'rho/sh-ediff-quit-hook t)"
47.
48.     if [ -z "$(lsuf -u $USER -a -c '/[Ee]macs/' 2>/dev/null | grep
'/server')" ] ; then
49.         emacs --rho-minimal &
50.         sleep 2
51.     fi
52.
53.     sleep 1
54.     shift
55. else
56.     LOCAL=$(abspath "$1")
57.     REMOTE=$(abspath "$2")
58.     MERGED=$(abspath "$3")
59.     BASE=$(abspath "$4")
60.     EMACS_CMD="emacs --rho-minimal"
61.     ELISP_COMMON=\
62. "(setq ediff-quit-hook 'kill-emacs \
63.     vc-handled-backends ())"
64. fi
65.
66. if [ $# -lt 2 ] ; then
67.     echo "Usage: $(basename $0) [--client] LOCAL REMOTE [MERGED
[BASE]]"
68.     echo "          (see also GIT-DIFFTOOL(1) and GIT-MERGETOOL(1))"
69.     exit 1
70. fi
71.
72. echo -e "$# files:\nLOCAL=$LOCAL\nREMOTE=$REMOTE"
73.
74. if [ $# -eq 4 ] ; then
75.     echo -e "MERGED=$MERGED\nBASE=$BASE"
76.     $EMACS_CMD --eval \
77. "(progn $ELISP_COMMON
78.     (setq-default ediff-show-clashes-only t)
79.     (ediff-merge-files-with-ancestor \ "${LOCAL}\ " \ "${REMOTE}\ "
\ "${BASE}\ " nil \ "${MERGED}\ "))" > /dev/null
80. elif [ $# -eq 2 -o "${REMOTE}" = "${MERGED}" ] ; then
81.     $EMACS_CMD --eval \
82. "(progn $ELISP_COMMON
83.     (ediff-files \ "${LOCAL}\ " \ "${REMOTE}\ "))" > /dev/null
84.     MERGE=0
85. else
86.     echo "MERGED=$MERGED"
87.     $EMACS_CMD --eval \
88. "(progn $ELISP_COMMON
89.     (setq-default ediff-show-clashes-only t)

```

```
90.      (ediff-merge-files \ "${LOCAL}\ " \ "${REMOTE}\ " nil
        \ "${MERGED}\ ") " > /dev/null
91.  fi
92.
93.  if [ $MERGE -ne 0 ]; then
94.      EGREPHITS="$(egrep -c '^(<<<<<<|>>>>>>)' $MERGED)"
95.
96.      if [ $EGREPHITS -ne 0 ]; then
97.          TMPFILE="$(mktemp --tmpdir $(basename $MERGED).XXXXXXXXXX)"
98.          cp -p "$MERGED" "$TMPFILE"
99.          echo "Conflicts -> see file $TMPFILE"
100.         exit 1
101.     fi
102. fi
103.
104. exit 0
```

From:

<https://wiki.rho62.de/> - rho62 Wiki

Permanent link:

<https://wiki.rho62.de/doku.php?id=development:emacs:ediff>

Last update: 2026/03/29 20:02

