

## CONTENTS

NAME .....	1
SYNOPSIS .....	1
DESCRIPTION .....	1
OPTIONS .....	2
INTERACTIVE COMMANDS .....	4
CONFIGURATION FILE ~/.vddiffrc .....	10
VDDIFF MODES .....	13
One-column diff mode .....	14
Two-column diff mode .....	14
One-column browse mode (bmode) .....	14
Two-column browse mode (fmode) .....	14
MOUSE SUPPORT .....	14
FILES .....	14
EXAMPLES .....	14

## NAME

**vddiff** — text terminal directory diff tool

## SYNOPSIS

```
vddiff [-u [RC file]] [-BbCcDeefgIiklMmnoqRrVwXy] [-F file name pattern]
[-G file content pattern] [-t diff_tool] [-v view_tool] [directory_1 [directory_2]]
```

## DESCRIPTION

**vddiff** stands for “Vim Directory DIFF”. It is a simple text terminal directory diff tool using **vim**(1) for the actual diff operation. (If **vim** is not installed, “**diff file\_1 file\_2 | less**” is used instead. Other diff tools can be configured as well.) A file browser displays the names found in the directories given on the command line. Only plain files and symbolic links are compared. Unless option **-r** is used, directories are not compared before they are entered.

Compressed files are automatically decompressed into a temporary directory before the diff tool is started. These files are detected by their file name extension, the file contents are not checked. Currently the following file name extensions are supported: bz2, gz, tar, tbz, tgz, and zip.

If **vddiff** is started with only one or no argument it behaves like a simple single directory view file browser (a mode called **bmode** whereas the normal mode is called **diff mode**). By use of marks files can be compared in bmode, but comparing directories is not possible. If less than two arguments are used in two-column mode kind of a file manager mode (called **fmode**) is started. This mode does not support global marks but allows comparison of directories. All **vddiff** modes are explained in section **VDDIFF MODES**.

Symbols are printed in two columns before the names which have the following meaning:

First column (or middle bar in two-column mode): Difference type

- ‘ ’ Directory, equal files or different file types
- ‘!’ Different files or links (or directories with option **-r**)
- ‘<’ Files found in first directory only
- ‘>’ Files found in second directory only
- ‘=’ Files have same i-node

- ‘-’ Error
- ‘X’ Two-column mode: Different file type

Second column (or first column in two-column mode and **bmode**): File type

- ‘/’ Directory
- ‘ ’ Plain file
- ‘@’ Symbolic link
- ‘c’ Character device
- ‘b’ Block device
- ‘=’ Socket
- ‘|’ FIFO
- ‘?’ Unknown file type
- ‘!’ Different file type

## OPTIONS

- B** Force two-column file browser mode even if two arguments are given. Normally diff mode is started with two arguments. Even if it is possible to change to browse mode with key ‘#’, it may take a long time until the (not needed) diff is done. So **-B** avoids diff mode at all.
- b** Disable color.
- C** Don’t compare file contents, only test for file existence. If differences regarding the existing of files are of interest in a directory with many huge files, it may take very long to compare their contents. **-C** does any tests but always assumes that two existing files are equal.  
  
Alternatively it is also possible to type ‘%’ to ignore file contents for the further directory diff. (‘%’ should pressed only short and one time. The comparison of the current file continues which may take a while, but further files are not compared.)
- c** Show only directories which exist in both trees and really different files.
- d** Use “diff \$1 \$2 | less” as diff tool. (If a specific diff tool (other than vim -d) should be used in general, it is recommended to configure it with the RC file option **difftool**, see section **CONFIGURATION FILE ~/.vddiffrc** below.)
- E** Use extended regular expressions.
- e** Use basic regular expressions.
- F pattern** Display only file names which match *pattern*, which is normally a case ignoring basic regular expression. With **-E** extended regular expressions are enabled, with **-I** case-sensitive matching is used. The pattern match is only applied to non-directories. To display only directories which contain matching files add option **-r** (command “**vddiff -rF pattern directory**” works in bmode somehow similar to “**find directory -name pattern**”). Key ‘E’ toggles display of all files and directories or matching files only. The pattern can be removed with the “:nofind” command.
- f** Normally directories are displayed on top. With this option files are displayed first.
- G pattern** Display only files which contain *pattern* (which is normally a case ignoring basic regular expression). With **-E** extended regular expressions are enabled, with **-I** case-sensitive matching is used. The pattern match is only applied to regular files. To display only directories which contain matching files add option **-r** (command “**vddiff -rG**

*pattern directory*” works in bmode somehow similar to “**grep -r pattern directory**”). Key ‘E’ toggles display of all files and directories or matching files only. Option **-G** can be combined with option **-F**. For example

```
vddiff -rF '\.[1-9]$\ ' -G '/usr'
```

finds all manpage source files which contain absolut path names. Option **-G** can be used multiple times, in this case **all patterns** are required to match. This differs from `grep(1)`, where at least one pattern needs to match. But this can also be expressed as

```
-EG '<pattern1>|<pattern1>'.
```

For example

```
vddiff -rF '\.[1-9]$\ ' -IG '^Dd' -EG '/(usr|etc|var)'
```

finds all manpage source files in **-mdoc** format which contain absolut path names. Options **-E**, **-e**, **-I**, and **-i** can be used again before each **-G** option to specify the kind of regular expression and the case sensitivity for the *pattern*. Further patterns can be added with the “**:grep pattern**” command, all patterns can be removed with the “**:nogrep**” command.

- g** Use **gvim** as diff and view tool.
- I** Use case-sensitive pattern match.
- i** Use case-insensitive pattern match.
- k** Use **tkdiff** as diff tool.
- l** Follow symbolic links.
- M** Normally flags **-W** and **-X** are only applied on single files operations. **-M** allows to use them on multiple file too.
- m** Normally directories are displayed on top. This is disabled with this option.
- n** This option suppresses the display of equal files.
- o** Hide files which are on one side only.
- q** Debug option: Print differing files and exit (similar to “**diff -q**”, but output is unsorted).
- R** Read-only mode: All file change operations (change name, permissions, owner, group; copy, move, delete) and function keys are disabled. To enable write mode, input “**:e**” or “**:edit**”. To only enable function keys, input “**:set fkeys**”.
- r** Recursively scan directories to detect differences in subdirectories. This allows to mark directories which contain differences. It increases the start time (due to disk I/O) since the full file tree is compared at begin. To only show different directories this option needs to be combined with **-c**. Pressing key ‘c’ enables to view all files in this mode.
- t diff\_tool** Specify diff tool on the command line. The filenames to compare are appended to the given string. To include them into the string the symbolic names “\$1” and “\$2” can be used (in any order), where “\$1” refers to the first and “\$2” to the second file. Note that the shell may require quoting as in

```
-t "diff \$1 \$2 | vim -R -"
```

- u** [*filename*]  
Skip reading the initialization file at start-up. If **-u** is used, it must be the very first option. Other used options need to begin with **-** again. If an optional *filename* is supplied, this file is read instead of the default initialization file `~/.vddiffrc`. *filename* needs to be separated with white space from **-u**.
- v**  
Print version and exit.
- v** *view\_tool*  
Specify view tool on the command line. The filenames is appended to the given string. To include it into the string the symbolic name “\$1” can be embedded which is expanded to the filename.
- w**  
Don’t ask for confirmation on file system operations.
- x**  
Don’t ask for confirmation on execute.
- y**  
Start in two-column mode. This is currently only supported if two arguments are given.

## INTERACTIVE COMMANDS

- ‘Q’**  
Quit **vddiff**.
- ‘h’** or **‘?’**  
Display help. Help mode is left with **‘q’**. Keys **⟨DOWN⟩**, **⟨UP⟩**, **⟨PAGE-DOWN⟩**, and **⟨PAGE-UP⟩** are used for scrolling in help mode.
- ⟨CTRL-l⟩**  
Refresh display. This may be necessary after another application had output text into the curses controlled display.
- ⟨TAB⟩**  
In **fmode**: Toggle column.
- ⟨UP⟩**, **‘k’** or **‘-’**  
Move cursor line up.
- ⟨DOWN⟩**, **‘j’** or **‘+’**  
Move cursor line down.
- ⟨LEFT⟩**  
Leave directory (one directory up).
- ⟨RIGHT⟩**, **⟨ENTER⟩**, or double click  
View file, enter directory or start diff tool. Compressed files and directories are unpacked before the view or diff tool is started.  
  
If a file is marked (with **‘m’**), only **⟨RIGHT⟩** and double click can be used to normally view files or enter directories. **⟨ENTER⟩** starts a diff between the marked file and the selected file or directory.
- ⟨PAGE-UP⟩** or **⟨BACKSPACE⟩**  
Scroll one screen up.
- ⟨PAGE-DOWN⟩** or **⟨SPACE⟩**  
Scroll one screen down.
- ⟨HOME⟩** or **‘1G’**  
Go to first file.
- ⟨END⟩** or **‘G’**  
Go to last file.
- ‘|’⟨LEFT⟩**  
In two-column mode: Enlarge right column by 10 characters.

' '⟨RIGHT⟩	In two-column mode: Enlarge left column by 10 characters.
" ="	In two-column mode: Make column widths equal.
⟨CTRL-w⟩	Toggle two-column mode.
'/'	Search file in list by typing the begin of the filename. Searching is normally done case-insensitive. Set option <b>noic</b> to change this. Search mode is left when a file is selected (with ⟨RIGHT⟩ or ⟨ENTER⟩).
" / /"	Search with a basic regular expression for a filename. This can be configured with options <b>noic</b> (don't ignore case), <b>magic</b> (use extended regular expressions), and <b>nows</b> (don't wrap around when search hits top or bottom of the file list). Regex search mode is not left until 'r' is pressed.  Previously entered search patterns are saved in a history, which can be accessed with the ⟨UP⟩ and ⟨DOWN⟩ keys.
"Sd"	Sort files with directories on top.
"Sm"	Sort files by name only (ignoring file type).
"SS"	Sort files by size with directories on top, smallest file first.
"St"	Sort files by modification time only, oldest first (ignoring file type).
'H'	Put cursor to top line.
'M'	Put cursor on middle line.
'L'	Put cursor on bottom line.
"z⟨ENTER⟩"	Put selected file to top.
"Z."	Center selected file.
"Z-"	Put selected file to bottom.
⟨CTRL-e⟩	Scroll one line down.
⟨CTRL-y⟩	Scroll one line up.
⟨CTRL-d⟩	Scroll half screen down.
⟨CTRL-u⟩	Scroll half screen up.
'!' or 'n'	Toggle display of equal files.
'c'	Toggle display of all files or only directories which exist in both trees and really different files.
'&'	Toggle display of files which are on one side only.
'F'	Toggle following symbolic links.
'E'	Toggle file name ( <b>-F</b> ) or file content ( <b>-G</b> ) filter.
'p'	Show current relative work directory.
'a'	Show command line directory arguments.
'f'	Show full path.
[n]"<<"	Copy from second to first tree. If <i>n</i> as a number between 2 and 9 (inclusive) is given, <i>n</i> files starting from the current selection are copied.

If symbolic links in source and target are followed or not is toggled with ‘F’. Following links is signaled with letter F on the right side of the status line.

When links are not followed the target file or directory is removed before the copy operation starts. This sets owner and group of the target file to the effective user and group of the **vddiff** process. To keep e.g. owner and group setting switch to follow links (with ‘F’). In this case files are only deleted when they are not writeable.

[n]“>>”	Copy from first to second tree.
“' <<”	Copy all files between the cursor and the local mark (inclusive) from second to first tree. After any ‘'’ command the cursor is set to the mark position if it had been below the mark.
“' >>”	Copy all files between the cursor and the local mark (inclusive) from first to second tree.
[n]“dd”	Delete file or directory, which must be present in one tree only. (Does not follow symbolic links.) <b>vddiff</b> does not warn if a directory to delete is not empty. If <i>n</i> as a number between 2 and 9 (inclusive) is given, <i>n</i> files starting from the current selection are deleted.
[n]“dl”	Delete file or directory in first tree. (Does not follow symbolic links.)
[n]“dr”	Delete file or directory in second tree. (Does not follow symbolic links.)
“' dd”	Delete all files between the cursor and the local mark (inclusive), which must be present in one tree only. (Does not follow symbolic links.)
“' dl”	Delete all files between the cursor and the local mark (inclusive) in first tree. (Does not follow symbolic links.)
“' dr”	Delete all files between the cursor and the local mark (inclusive) in second tree. (Does not follow symbolic links.)
[n]‘T’	<b>fmode</b> only: Move file or directory to the other file tree.
[n]“Tl”	Move file or directory to left file tree.
[n]“Tr”	Move file or directory to right file tree.
“' T”	<b>fmode</b> only: Move all files between the cursor and the local mark (inclusive) to the other file tree.
“' Tl”	Move all files between the cursor and the local mark (inclusive) to left file tree.
“' Tr”	Move all files between the cursor and the local mark (inclusive) to right file tree.
[n]‘@’	<b>fmode</b> only: Create symbolic link in other column to selected file in active column.
[n]“@l”	Create symbolic link in left file tree to file in right file tree.
[n]“@r”	Create symbolic link in right file tree to file in left file tree.
“' @”	<b>fmode</b> only: Create symlinks in other column to all files between the cursor and the local mark (inclusive) in active column.
“' @l”	Create symlinks in left file tree to all files between the cursor and the local mark (inclusive) in right file tree.
“' @r”	Create symlinks in right file tree to all files between the cursor and the local mark (inclusive) in left file tree.

"en"	Rename file, which must be present in one tree only.
"eln"	Rename file in first tree.
"ern"	Rename file in second tree.
[n]"ep"	Change permissions of file, which must be present in one tree only. If following symbolic links is not enabled the command is ignored for symbolic links. If decimal digit <i>n</i> is given, the mode of <i>n</i> files starting from the current selection is set.
[n]"elp"	Change permissions of file in first tree.
[n]"erp"	Change permissions of file in second tree.
"'ep"	Change permissions of all files between the cursor and the local mark (inclusive), which must be present in one tree only.
"'elp"	Change permissions of all files between the cursor and the local mark (inclusive) in first tree.
"'erp"	Change permissions of all files between the cursor and the local mark (inclusive) in second tree.
[n]"eu"	Change owner of file, which must be present in one tree only. If following symbolic links is not enabled the command is ignored for symbolic links. Previously entered user names are saved in a history, which can be accessed with the <UP> and <DOWN> keys. If decimal digit <i>n</i> is given, the owner of <i>n</i> files starting from the current selection is set.
[n]"elu"	Change owner of file in first tree.
[n]"eru"	Change owner of file in second tree.
"'eu"	Change owner of all files between the cursor and the local mark (inclusive) which must be present in one tree only.
"'elu"	Change owner of all files between the cursor and the local mark (inclusive) in first tree.
"'eru"	Change owner of all files between the cursor and the local mark (inclusive), in second tree.
[n]"eg"	Change group of file, which must be present in one tree only. If following symbolic links is not enabled the command is ignored for symbolic links. Previously entered group names are saved in a history, which can be accessed with the <UP> and <DOWN> keys. If decimal digit <i>n</i> is given, the group of <i>n</i> files starting from the current selection is set.
[n]"elg"	Change group of file in first tree.
[n]"erg"	Change group of file in second tree.
"'eg"	Change group of all files between the cursor and the local mark (inclusive) which must be present in one tree only.
"'elg"	Change group of all files between the cursor and the local mark (inclusive) in first tree.
"'erg"	Change group of all files between the cursor and the local mark (inclusive), in second tree.
'p'	Create directory ( <b>bmode</b> and <b>fmode</b> only).
"p1"	Create directory in left tree.

- “Pr” Create directory in right tree.
- ‘.’ Repeat last file system or function key command. For commands using marks and the ‘.’ key the number of files is saved. This number is applied on a ‘.’ command, starting from the cursor position (the mark is not relevant then).
- ‘m’ Mark file or directory. This can be used to compare files or directories which had been renamed or compressed in one file tree.
- Marks are defined globally and can be used to compare files from different file hierarchy depths. They can only be undefined with the ‘r’ key (or redefined with the ‘m’ key). As long as the current directory is not left, the marked file is highlighted with either a bold font or a blue background. (In **fm**ode marks are cleared if the directory is left or if any external command is executed.)
- If a mark is set, only **<RIGHT>** and double click can be used to normally view files or change to other directories. **<ENTER>** starts a diff between the marked and the selected file or directory.
- ‘r’ Remove mark, edit line, or regex search.
- ‘b’ Test for binary difference between selected and marked file. Compressed files are unpacked but compressed archive files are compared directly.
- ‘y’ Copy file path(s) to edit line. If a ‘\$’ command is entered later, this file path can be used to build a `sh(1)` command.
- ‘Y’ Copy file paths to edit line in reverse order.
- ‘\$’ Enter shell command. If paths had been copied to the edit line before using the ‘y’ or ‘Y’ command, the shell command can be prepended by pressing **<HOME>** and then entering the command. Predefined strings can be inserted by pressing a function key. The work directory is the directory where **vddiff** had been started. (In **bm**ode and **fm**ode the work directory is always the current view directory.) Each entered command is saved in a history. The keys **<UP>** and **<DOWN>** fetch other history entries. The shell to be used (default “sh”) can be configured with the **sh** option.
- [*n* | ‘!’]**<F1>** – **<F12>**
- Define string which can be inserted later with this function key when entering a `sh` command using ‘\$’. This string is usually the name of a UNIX tool. Regularly used strings can be set using the RC file **key** command.
- If the string begins with a ‘\$’ followed by at least one space (‘ ’) it is treated as shell command itself, which is applied to a selected file(s). If that function key is pressed later, a dialog opens to ask if the command should be executed or the function key should be redefined. The filename is appended to the saved string. To embed it, “\$1” and “\$2” can be used, as in “\$ nroff \$1 | less”.
- If *n* as a number between 2 and 9 (inclusive) is given, the command is applied to *n* files starting from the current selection. If ‘!’ is prepended instead, the command is applied to all files between the cursor and the local mark (inclusive).
- If the string starts with ‘!’ instead of ‘\$’, **<ENTER>** must be pressed after running the command. This allows to check the command’s output.
- For strings starting with ‘\$’ or ‘!’ the user has to confirm the command execution. If the string starts with ‘#’ instead the command is started immediately.



To save a command for a function key which does not take the selected filename as argument, add a space followed by an octothorpe (“ #”) at the end of the string.

If no diff tool which supports merging is available, files can be merged manually by defining two function keys for editing the left and right side file with “\$ vi \$1” and “\$ vi \$2”.

'l'	List strings which had been defined for a function key.
'u'	Update file list.
's'	Open shell in current directory. The shell to open can be set with the <b>shell</b> option, else it is read from the user's entry in <code>/etc/passwd</code> .
"sl"	Open shell in left directory tree.
"sr"	Open shell in right directory tree.
'o'	Open file (instead of diff tool).
"ol"	Open left file or directory. Plain files are opened with <code>less(1)</code> by default. If no diff tool which supports merging is available, files can be merged manually opening them with “ol” or “or” and then pressing ‘v’ in <code>less</code> which opens <code>vi</code> .
"or"	Open right file or directory.
'v'	View raw file contents.
"vl"	View raw left file contents.
"vr"	View raw right file contents.
':'	Enter command:
<code>cd</code>	bmode and fmode only: Change to home directory.
<code>cd path</code>	bmode and fmode only: Change to directory <i>path</i> . <i>path</i> may be absolute or relative. ‘~’ and environment variables are not (yet) expanded.
<code>e[dit]</code>	Allow file change operations and function keys.
<code>find pattern</code>	Display only filenames which match <i>pattern</i> .
<code>nofind</code>	Remove filename pattern.
<code>grep pattern</code>	Display only files which contain <i>pattern</i> .
<code>nogrep</code>	Remove file content pattern.
<code>q, qa</code>	Quit <code>vddiff</code> .
<code>set all</code>	Display the current setting of the changable options.
<code>set fkeys</code>	Enable function keys.
<code>set nofkeys</code>	Disable function keys.
<code>set ic</code>	Set case-insensitive match.

<code>set noic</code>	Set case-sensitive match.
<code>set magic</code>	Use extended regular expressions.
<code>set nomagic</code>	Use basic regular expressions.
<code>set recursive</code>	Use recursive diff, find and grep operations.
<code>set norecursive</code>	Use directory local diff, find and grep.
<code>set ws</code>	File name searches wrap around top and bottom.
<code>set nows</code>	File name searches don't wrap around top and bottom.
<code>vie[w]</code>	Read-only mode: Disable file change operations and function keys.
	Previously entered commands are saved in a history, which can be accessed with the <b>&lt;UP&gt;</b> and <b>&lt;DOWN&gt;</b> keys.
<code>'#'</code>	Toggle between diff mode and two-column browse mode ( <b>fmode</b> ).
<code>'='</code>	In <b>fmode</b> : Set path of active column to path of the other column.
<code>'%'</code>	Toggle ignoring file contents for further diff operations. This is useful if only differences regarding the file existence are of interest. Any time this function is disabled, the currently displayed directories are compared again. <code>'%'</code> can be pressed during this directory compare (short and only one time) to abort it. Note that an active file diff is not interrupted. This may take some time in case of a huge file.
<code>'w'</code>	Toggle to always wait for <b>&lt;ENTER&gt;</b> after running an external tool. This can be useful for debugging purposes.

## CONFIGURATION FILE `~/.vddiffrc`

Permanent non-default options can be set in the file `~/.vddiffrc`, which has a very simple (*not* .exrc like) syntax. The elements in this file may be separated with spaces, tabs or line breaks. Line breaks are not required, everything can be written into one long line. Also spaces or tabs are not required (outside quoted strings), when every element is on it's own line. Everything following a `#` (outside quoted strings) to the end of the line is a comment.

`twocolumn` Start in two-column mode. This is currently only supported if two arguments are given.

`difftool string`

Configure *string* as diff tool. If *string* contains spaces it needs to be quoted with `"` at begin and end. The two filenames are appended to this string. If the filenames need to be before a pipe symbol (`|`), the symbols `"$1"` and `"$2"` can be embedded into *string* (in any order). These symbols are expanded to the respective filename.

Other possible diff tools are `colordiff(1)`, `mgdiff(1)`, or `tkxcd(1)`.

`difftool bg string`

Start *string* as a background process (don't block **vdiff** while executing *string*). **bg** is ignored if one of the files to be compared is compressed. Since they are decompressed into a temporary directory which is removed after starting the diff tool, the file may be removed before the tool reads it.

`viewtool string`

Configure *string* as view tool. If *string* contains spaces it needs to be quoted with

‘”’ at begin and end. The filename is appended to this string. If the filename need to be before a pipe symbol (‘|’) the symbol “\$1” can be embedded into *string*, which is expanded to the filename.

**viewtool** *bg string*

Start *string* as a background process. **bg** is ignored if the file to be viewed is compressed. Since it is decompressed into a temporary directory which is removed after starting the view tool, the file may be removed before the tool reads it. For example

```
viewtool bg "xterm -e less"
```

displays every file to view in a separate window while not blocking the file browser.

**ext** *extension string*

Configure view tool *string* for filenames ending with “*extension*”. *extension* is compared case-insensitive. If *extension* is an integer number it needs to be enclosed in double quotes (‘”’) as in

```
ext "1" "tbl $1 | neqn | nroff -mandoc | less"
```

because the parser expects a string (and not a number) after the keyword **ext**.

**ext** *extension bg string*

Start *string* as a background process.

**alias** *name string*

If a pipe of tools (like the **nroff** pipe above) or a command with options (like e.g. “**mplayer -idx --**” should be applied to many file types, an alias *name* can be defined for such a *string*. The **ext** command can then use this alias *name* instead the full command *string*, for example

```
alias mandoc "tbl $1 | neqn | nroff -mandoc | less"
alias mplayer "mplayer -idx --"
```

```
ext "1" mandoc
ext "2" mandoc
ext "3" mandoc
```

```
ext mts mplayer
ext ts mplayer
```

It is also possible to use an alias for defining another alias:

```
alias mplayer "mplayer -idx --"
alias audio mplayer
```

```
ext mpg mplayer
ext mp3 audio
```

**skipext** *extension*

When checking a filename extension skip *extension* at the end of the filename. E.g.

```
skipext old
```

would remove “.old” from the end of each filename before checking the extension.

**fkey** *number string*

Define *string* which can be inserted later with this function key when entering a **sh** command using ‘\$’. This string is usually the name of a UNIX tool. *number* must be

between 1 and 12 (inclusive).

If *string* begins with a '\$' followed by at least one space (the string needs to be enclosed in double quotes in this case), it defines a shell command to be executed for the selected file when pressing the function key.

If *string* starts with '!' instead of '\$', **<ENTER>** must be pressed after running the command. This allows to check the command's output.

To save a command for a function key which does not take the selected filename as argument, add a space followed by an octothorpe (" #") at the end of the string.

mono	Disable colors.
followlinks	Follow symbolic links.
noequal	Display differences only, hide equal files.
real_diff	Show only directories which exist in both trees and really different files.
recursive	Recursively scan file tree to detect and mark directories which contain differences. This increases the start time. To show only directories with differences additionally set option <i>real_diff</i> . To view all files when in this mode key 'c' can be used.
noic	Searching for a filename with '/' or "/" is normally done case-insensitive. Case-sensitive search can be configured with this option.
magic	For searching with "/" normally basic regular expressions are used. Use of extended regular expressions is configured with this option.
nows	Searching for a filename with "/" normally wraps around top and bottom of the file list. This behaviour is disabled with this option.
include <i>string</i>	Include RC file <i>string</i> . If <i>string</i> does not start with '/', \$HOME is prepended. '~' and environment variables are <i>not</i> expanded. (The include command does only work if vddiff had been build with a lex version which supports buffer switching.)
filesfirst	Display directories at the end instead on top.
mixed	Display files and directories mixed.
dir_color <i>integer</i>	Set color for directories. Default is 3 (yellow).
diff_color <i>integer</i>	Set color for different files. Default is 1 (red).
link_color <i>integer</i>	Set color for symbolic links. Default is 5 (magenta).
left_color <i>integer</i>	Set color for files found in first directory only. Default is 6 (cyan).
right_color <i>integer</i>	Set color for files found in second directory only. Default is 2 (green).
unknown_color <i>integer</i>	

Set color for unknown file types. Default is 4 (blue).

`normal_color integer`

Set color for normal text. Default is 7 (white).

`bg_color integer`

Set background color. Default is 0 (black). E.g.

```
bg_color 4 mark_color 7 0
```

changes the background color to blue (and the normally blue mark color to black).

`cursor_color foreground background`

Set the cursor color. *foreground* and *background* must be given as integers. Default is 0 7 (black on white).

`mark_color foreground background`

Set color for marks. *foreground* and *background* must be given as integers. Default is 7 4 (white on blue).

`error_color foreground background`

Set color for diff errors. *foreground* and *background* must be given as integers. Default is 7 1 (white on red).

`histsize integer`

Set history size to *integer*. Default is 100. A size less than 2 disables the history (at least the current command line and one previous entry need to be saved to use this feature).

`scale`

Show file size in human-readable format.

`shell string`

Set shell for the ‘s’, “sl” and “sr” command. Default is the user’s login shell. *string* may be a full path or just the shell name. In the latter case \$PATH is searched to determine the full path.

`sh string`

Set shell for the ‘\$’ and function key commands, .gz and .bz2 uncompressing and any use of the shell special characters |, &, ;, <, >, (, ), `, \, ", ', [, #, and ~. Default is “sh”. *string* may be a full path or just the shell name. In the latter case \$PATH is searched to determine the full path.

`readonly`

Read-only mode: All file change operations (change name, permissions, owner, group; copy, move, delete) and function keys are disabled. Function keys get enabled if a **fkey** keyword is read after **readonly**. To avoid this, **readonly** can be placed at the end of the RC file.

## VDDIFF MODES

**vdiff** has four modes: A diff mode and a browser mode, both have an one-column and a two-column submode. The following table shows the transition between these modes: Key **<CTRL-w>** toggles between one-column and two-column mode and key ‘#’ toggles between diff mode and browse mode:

```

                                [1-column]      [2-column]

[Diff mode]                    <---- ^W ---->
                                ^               ^
                                |               |
                                `#'            `#'
                                |               |
                                v               v
[Browse mode]                  <---- ^W ---->

```

While the diff modes access files with their path name, the browse modes always set the current working directory to the active file list column. This is relevant if shell commands are applied which create files. Marks are cleared on any mode change except switching between one-column and two-column diff mode.

### One-column diff mode

Diff mode is entered when two arguments are used. Key **<CTRL-w>** switches to two-column diff mode, key **#** switches to two-column browse mode.

### Two-column diff mode

Two-column diff mode is equal to diff mode except for a different data presentation. Key **<CTRL-w>** switches to single-column diff mode, key **#** switches to two-column browse mode.

### One-column browse mode (bmode)

Bmode is a simply file browser mode. It allows file comparison by use or marks, but directory comparison is not possible. Key **<CTRL-w>** switches to two-column browse mode.

### Two-column browse mode (fmode)

Fmode is a simple two-column file manager mode. It allows comparison of files and directories by use of local marks. This mode does not have global marks. If a directory is left, an existing mark in this directory is deleted. Generally in any mode at most one mark is supported. If a mark is defined, an existing mark (in either column) is deleted.

The symbolic variables **“\$1”** and **“\$2”** exchange when the cursor is in the right column. That means **“\$1”** is the file from the cursor column and **“\$2”** is the marked file in the other column.

Key **<CTRL-w>** switches to single-column browse mode, key **#** or a compare of a marked directory (or archive) with a directory (or archive) switches to two-column diff mode.

## MOUSE SUPPORT

For mouse support currently the ncurses interface is used which means that ncurses is required if one wants to use the mouse. Furthermore at least ncurses version 6 is currently required for using the mouse scroll wheel.

If the middle bar can be moved with the mouse to resize the columns in two-column mode is terminal dependend. See section **PORTABILITY** in mouse(3ncurses). For example, to make this function best work in xterm, set **“TERM=xterm-1002”** (else usually a further mouse click is required to finalize the resize).

## FILES

```
~/vddiffrc
```

Read on start-up to set non-default options.

## EXAMPLES

To display only differing files and subdirectories which contain differing files in two-column mode enter

```
vddiff -cry directory_1 directory_2
```

where **-c** hides equal files, files which exist in one directory only and directories which don't contain differing files. Pressing 'c' toggles showing all files. **-r** enables recursive mode. This option increases the start time, if huge directories are compared. If this takes too long, '%' can be pressed to disable further file content comparison. **-y** enables two-column mode.

To only hide equal files enter

```
vddiff -nry directory_1 directory_2
```

Pressing 'n' toggles showing all files.

If a recursive diff is not required,

```
vddiff -y directory_1 directory_2
```

(optionally with option **-c** or **-n**) is much faster, since a diff is only done, when a directory is entered. The only disadvantage is, that also directories may be shown, which contain no differences (since the tool had not visited them yet).