

# GNU wdiff, version 0.5

---

A word difference finder  
Edition 0.5, November 1994

François Pinard

---

Copyright © 1992, 1994 Free Software Foundation, Inc.

Permission is granted to make and distribute verbatim copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the entire resulting derived work is distributed under the terms of a permission notice identical to this one.

Permission is granted to copy and distribute translations of this manual into another language, under the above conditions for modified versions, except that this permission notice may be stated in a translation approved by the Foundation.

# 1 Overview

The program `wdiff` is a front end to `diff` for comparing files on a word per word basis. A word is anything between whitespace. This is useful for comparing two texts in which a few words have been changed and for which paragraphs have been refilled. It works by creating two temporary files, one word per line, and then executes `diff` on these files. It collects the `diff` output and uses it to produce a nicer display of word differences between the original files.

Ideally, `wdiff` should avoid calling `diff` and do all the work internally, allowing it to be faster and more polished. However, I loathe replicating the `diff` algorithm and development effort, instead of improving `diff` itself. It would be more sensible to integrate `wdiff` into `diff` than the other way around. I did it this way only because I had a sudden and urgent need for it, and it would have taken too much time to integrate it correctly into GNU `diff`. Your advice or opinions about this are welcome.

`wdiff` was written by François Pinard. Please report bugs to ‘`bug-gnu-utils@prep.ai.mit.edu`’. Include the version number, which you can find by running ‘`wdiff --version`’. Include in your message sufficient input to reproduce the problem and also, the output you expected.

## 2 Invoking `wdiff`

The format for running the `wdiff` program is:

```
wdiff option ... old_file new_file
```

`wdiff` compares files *old\_file* and *new\_file* and produces an annotated copy of *new\_file* on standard output. The empty string or the string ‘-’ denotes standard input, but standard input cannot be used twice in the same invocation. The complete path of a file should be given, a directory name is not accepted. `wdiff` will exit with a status of 0 if no differences were found, a status of 1 if any differences were found, or a status of 2 for any error.

In this documentation, *deleted text* refers to text in *old\_file* which is not in *new\_file*, while *inserted text* refers to text on *new\_file* which is not in *old\_file*.

`wdiff` supports the following command line options:

‘--help’

‘-h’            Print an informative help message describing the options.

‘--version’

‘-v’            Print the version number of `wdiff` on the standard error output.

‘--no-deleted’

‘-1’            Avoid producing deleted words on the output. If neither ‘-1’ or ‘-2’ is selected, the original right margin may be exceeded for some lines.

‘--no-inserted’

‘-2’            Avoid producing inserted words on the output. When this flag is given, the whitespace in the output is taken from *old\_file* instead of *new\_file*. If neither ‘-1’ or ‘-2’ is selected, the original right margin may be exceeded for some lines.

‘--no-common’

‘-3’            Avoid producing common words on the output. When this option is not selected, common words and whitespace are taken from *new\_file*, unless option ‘-2’ is given, in which case common words and whitespace are rather taken from *old\_file*. When selected, differences are separated from one another by lines of dashes. Moreover, if this option is selected at the same time as ‘-1’ or ‘-2’, then none of the output will have any emphasis, i.e. no bold or underlining. Finally, if this option is not selected, but both ‘-1’ and ‘-2’ are, then sections of common words between differences are segregated by lines of dashes.

**'--ignore-case'**

**'-c'** Do not consider case difference while comparing words. Each lower case letter is seen as identical to its upper case equivalent for the purpose of deciding if two words are the same.

**'--statistics'**

**'-s'** On completion, for each file, the total number of words, the number of common words between the files, the number of words deleted or inserted and the number of words that have changed is output. (A changed word is one that has been replaced or is part of a replacement.) Except for the total number of words, all of the numbers are followed by a percentage relative to the total number of words in the file.

**'--auto-pager'**

**'-a'** Some initiatives which were previously automatically taken in previous versions of `wdiff` are now put under the control of this option. By using it, a pager is interposed whenever the `wdiff` output is directed to the user's terminal. Without this option, no pager will be called, the user is then responsible for explicitly piping `wdiff` output into a pager, if required.

The pager is selected by the value of the `PAGER` environment variable when `wdiff` is run. If `PAGER` is not defined at run time, then a default pager, selected at installation time, will be used instead. A defined but empty value of `PAGER` means no pager at all.

When a pager is interposed through the use of this option, one of the options `'-l'` or `'-t'` is also selected, depending on whether the string `'less'` appears in the pager's name or not.

It is often useful to define `'wdiff'` as an alias for `'wdiff -a'`. However, this *hides* the normal `wdiff` behaviour. The default behaviour may be restored simply by piping the output from `wdiff` through `cat`. This dissociates the output from the user's terminal.

**'--printer'**

**'-p'** Use over-striking to emphasize parts of the output. Each character of the deleted text is underlined by writing an underscore `'_'` first, then a backspace and then the letter to be underlined. Each character of the inserted text is emboldened by writing it twice, with a backspace in between. This option is not selected by default.

**'--less-mode'**

**'-l'** Use over-striking to emphasize parts of output. This option works as option `-p`, but also over-strikes whitespace associated with inserted text. `less` shows such whitespace using reverse video. This option is not selected by default. However, it is automatically turned on whenever `wdiff` launches the pager `less`. See option `'-a'`.

This option is commonly used in conjunction with `less`:

```
wdiff -l old_file new_file | less
```

‘--terminal’

‘-t’ Force the production of `termcap` strings for emphasising parts of output, even if the standard output is not associated with a terminal. The ‘`TERM`’ environment variable must contain the name of a valid `termcap` entry. If the terminal description permits, underlining is used for marking deleted text, while bold or reverse video is used for marking inserted text. This option is not selected by default. However, it is automatically turned on whenever `wdiff` launches a pager, and it is known that the pager is *not less*. See option ‘-a’.

This option is commonly used when `wdiff` output is not redirected, but sent directly to the user terminal, as in:

```
wdiff -t old_file new_file
```

A common kludge uses `wdiff` together with the pager `more`, as in:

```
wdiff -t old_file new_file | more
```

However, some versions of `more` use `termcap` emphasis for their own purposes, so strange interactions are possible.

‘--start-delete *argument*’

‘-w *argument*’

Use *argument* as the *start delete* string. This string will be output prior to any sequence of deleted text, to mark where it starts. By default, no start delete string is used unless there is no other means of distinguishing where such text starts; in this case the default start delete string is ‘[-’.

‘--end-delete *argument*’

‘-x *argument*’

Use *argument* as the *end delete* string. This string will be output after any sequence of deleted text, to mark where it ends. By default, no end delete string is used unless there is no other means of distinguishing where such text ends; in this case the default end delete string is ‘-]’.

‘--start-insert *argument*’

‘-y *argument*’

Use *argument* as the *start insert* string. This string will be output prior to any sequence of inserted text, to mark where it starts. By default, no start insert string is used unless there is no other means of distinguishing where such text starts; in this case the default start insert string is ‘{+’.

‘--end-insert *argument*’

‘-z *argument*’

Use *argument* as the *end insert* string. This string will be output after any sequence of inserted text, to mark where it ends. By default, no end insert string is used unless there is no other means of distinguishing where such text ends; in this case the default end insert string is ‘+}’.

`--avoid-wraps`

`-n`      Avoid spanning the end of line while showing deleted or inserted text. Any single fragment of deleted or inserted text spanning many lines will be considered as being made up of many smaller fragments not containing a newline. So deleted text, for example, will have an end delete string at the end of each line, just before the new line, and a start delete string at the beginning of the next line. A long paragraph of inserted text will have each line bracketed between start insert and end insert strings. This behaviour is not selected by default.

Note that options `-p`, `-t`, and `-[wxyz]` are not mutually exclusive. If you use a combination of them, you will merely accumulate the effect of each. Option `-l` is a variant of option `-p`.

### 3 Actual examples of `wdiff` usage

This section presents a few examples of usage, most of them have been contributed by `wdiff` users.

- Change bars example.

This example comes from a discussion with Joe Wells, ‘`jbw@cs.bu.edu`’.

The following command produces a copy of *new\_file*, shifted right one space to accommodate change bars since the last revision, ignoring those changes coming only from paragraph refilling. Any line with new or changed text will get a ‘|’ in column 1. However, deleted text is not shown nor marked.

```
wdiff -1n old_file new_file |
sed -e 's/^/ /;/{+/s/^ /|/;s/{+//g;s/{+}//g'
```

Here is how it works. Word differences are found, paying attention only to additions, as requested by option ‘-1’. For bigger changes which span line boundaries, the insert bracket strings are repeated on each output line, as requested by option ‘-n’. This output is then reformatted with a `sed` script which shifts the text right two columns, turns the initial space into a bar only if there is some new text on that line, then removes all insert bracket strings.

- LaTeX example.

This example has been provided by Steve Fisk, ‘`fisk@polar.bowdoin.edu`’.

The following uses LaTeX to put deleted text in boxes, and new text in double boxes:

```
wdiff -w "\fbox{" -x "}" -y "\fbox{\fbox{" -z "}}" ...
```

works nicely.

- troff example.

This example comes from Paul Fox, ‘`pgf@cayman.com`’.

Using `wdiff`, with some `troff`-specific delimiters gives *much* better output. The delimiters I used:

```
wdiff -w'\s-5' -x'\s0' -y'\fB' -z'\fP' ...
```

This makes the pointsize of deletions 5 points smaller than normal, and emboldens insertions. Fantastic!

I experimented with:

```
wdiff -w'\fI' -x'\fP' -y'\fB' -z'\fP'
```

since that’s more like the defaults you use for terminals/printers, but since I actually use italics for emphasis in my documents, I thought the point size thing was clearer.

I tried it on code, and it works surprisingly well there, too...

Marty Leisner ‘`leisner@eso.mc.xerox.com`’ says:



In the previous example, you had smaller text being taken out and bold face inserted. I had smaller text being taken out and larger text being inserted, I'm using bold face for other things, so this is more clear.

```
wdiff -w '\s-3' -x'\s0' -y'\s+3' -z'\s0'
```