



## ***Linux Ubuntu 22.4.5 Manual Pages on command 'grepdiff.1'***

**\$ man grepdiff.1**

GREPDIFF(1) Man pages GREPDIFF(1)

### NAME

grepdiff - show files modified by a diff containing a regex

### SYNOPSIS

```
grepdiff [[-n] | [--line-number]] [[-N] | [--number-files]] [[-p n] |
  [--strip-match=n]] [--strip=n] [--addprefix=PREFIX]
  [--addoldprefix=PREFIX] [--addnewprefix=PREFIX] [[-s] | [--status]]
  [[-i PATTERN] | [--include=PATTERN]] [[-I FILE] |
  [--include-from-file=FILE]] [[-x PATTERN] | [--exclude=PATTERN]]
  [[-X FILE] | [--exclude-from-file=FILE]] [[-# RANGE] | [--hunks=RANGE]]
  [--lines=RANGE] [[-FRANGE] | [--files=RANGE]] [--annotate]
  [--as-numbered-lines=WHEN] [--format=FORMAT] [--remove-timestamps] [[-v] |
  [--verbose]] [[-z] | [--decompress]] [[-E] | [--extended-regexp]] [[-H] |
  [--with-filename]] [[-h] | [--no-filename]] [--output-matching=WHAT]
  {[REGEX] | [-f FILE]} [file...]
```

```
grepdiff [--help] | [--version] | [--list] | [--filter ...]
```

### DESCRIPTION

For each file modified by a patch, if the patch hunk contains the REGEX then the

file's name is printed.

The regular expression is treated as POSIX Basic Regular Expression syntax, unless the -E option is given in which case POSIX Extended Regular Expression syntax is used.

For example, to see the patches in my.patch which contain the regular expression ?pf\_gfp\_mask?, use:

```
grepdiff pf_gfp_mask my.patch | \  
xargs -rn1 filterdiff my.patch -i
```

You can use both unified and context format diffs with this program.

## OPTIONS

-n, --line-number

Display the line number that each patch begins at. If verbose output is requested, each matching hunk is listed as well.

For a description of the output format see `lsdiff(1)`.

-N, --number-files

File numbers are listed, beginning at 1, before each filename.

-p n, --strip-match=n

When matching, ignore the first n components of the pathname.

--strip=n

Remove the first n components of the pathname before displaying it.

--addprefix=PREFIX

Prefix the pathname with PREFIX before displaying it. This will override any individual settings specified with the --addoldprefix or --addnewprefix

options.

--addoldprefix=PREFIX

Prefix pathnames for old or original files in the output by PREFIX.

--addnewprefix=PREFIX

Prefix pathnames for updated or new files in the output by PREFIX.

-s

Show file additions, modifications and removals. A file addition is indicated by a ?+?, a removal by a ?-?, and a modification by a ?!?.

-i PATTERN, --include=PATTERN

Include only files matching PATTERN.

-I FILE, --include-from-file=FILE

Include only files matching any pattern listed in FILE, one pattern per line. All other lines in the input are suppressed.

-x PATTERN --exclude=PATTERN

Exclude files matching PATTERN.

-X FILE, --exclude-from-file=FILE

Exclude files matching any pattern listed in FILE, one pattern per line. All other lines in the input are displayed.

-# RANGE, --hunks=RANGE

Only include hunks within the specified RANGE. Hunks are numbered from 1, and the range is a comma-separated list of numbers or ?first-last? spans, optionally preceeded by a modifier 'x' which inverts the entire range; either the first or the last in the span may be omitted to indicate no limit in that direction.

`--lines=RANGE`

Only list hunks that contain lines from the original file that lie within the specified RANGE. Lines are numbered from 1, and the range is a comma-separated list of numbers or `?first-last?` spans, optionally preceded by a modifier `'x'` which inverts the entire range; either the first or the last in the span may be omitted to indicate no limit in that direction.

`-F=RANGE, --files=RANGE`

Only list files indicated by the specified RANGE. Files are numbered from 1 in the order they appear in the patch input, and the range is a comma-separated list of numbers or `?first-last?` spans, optionally preceded by a modifier `'x'` which inverts the entire range; either the first or the last in the span may be omitted to indicate no limit in that direction.

`--annotate`

Annotate each hunk with the filename and hunk number.

`--as-numbered-lines=before|after`

Instead of a patch fragment, display the lines of the selected hunks with the line number of the file before (or after) the patch is applied, followed by a TAB character and a colon, at the beginning of each line. Each hunk except the first will have a line consisting of `?...?` before it.

`--format=unified|context`

Use specified output format.

`--remove-timestamps`

Do not include file timestamps in the output.

`-z, --decompress`

Decompress files with extensions `.gz` and `.bz2`.

`-E, --extended-regexp`

Use POSIX Extended Regular Expression syntax.

`-H, --with-filename`

Print the name of the patch file containing each match.

`-h, --no-filename`

Suppress the name of the patch file containing each match.

`-f FILE, --file=FILE`

Read regular expressions from FILE, one per line.

`--output-matching=hunk|file`

Display the matching hunk-level or file-level diffs.

`--help`

Display a short usage message.

`--version`

Display the version number of grepdiff.

`--filter`

Behave like filterdiff(1) instead.

`--list`

Behave like lsdiff(1) instead.

## SEE ALSO

filterdiff(1), lsdiff(1)

## AUTHOR

Tim Waugh <twauth@redhat.com>

Package maintainer

