Linux Ubuntu 22.4.5 Manual Pages on command 'filterdiff.1'

$ man filterdiff.1

FILTERDIFF(1) Man pages FILTERDIFF(1)

NAME

filterdiff - extract or exclude diffs from a diff file

SYNOPSIS

filterdiff [[-i PATTERN] | [--include=PATTERN]] [[-I FILE] |

[--include-from-file=FILE]] [[-p n] | [--strip-match=n]] [--strip=n]

[--addprefix=PREFIX] [--addoldprefix=PREFIX] [--addnewprefix=PREFIX]

[[-x PATTERN] | [--exclude=PATTERN]] [[-X FILE] |

[--exclude-from-file=FILE]] [[-v] | [--verbose]] [--clean] [[-z] |

[--decompress]] [[-# RANGE] | [--hunks=RANGE]] [--lines=RANGE]

[[-FRANGE] | [--files=RANGE]] [--annotate] [--format=FORMAT]

[--as-numbered-lines=WHEN] [--remove-timestamps] [file...]

filterdiff {[--help] | [--version] | [--list] | [--grep ...]}

DESCRIPTION

You can use filterdiff to obtain a patch that applies to files matching the shell wildcard PATTERN from a larger collection of patches. For example, to see the patches in patch-2.4.3.gz that apply to all files called lp.c:

filterdiff -z -i '*/lp.c' patch-2.4.3.gz

If neither -i nor -x options are given, -i '*' is assumed. This way filterdiff can be used to clean up an existing diff file, removing redundant lines from the beginning (eg. the text from the mail body) or between the chunks (eg. in CVS diffs). To extract pure patch data, use a command like this:

filterdiff message-with-diff-in-the-body > patch

Note that the interpretation of the shell wildcard pattern does not count slash characters or periods as special (in other words, no flags are given to fnmatch). This is so that ?*/basename?-type patterns can be given without limiting the number of pathname components.

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

OPTIONS

-i PATTERN, --include=PATTERN

Include only files matching PATTERN. All other lines in the input are suppressed.

-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. All other lines in the input are displayed.

-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.

-p n, --strip-match=n

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

-# 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, optionially 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 include 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, optionially 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.

-F=RANGE, --files=RANGE

Only include 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, optionially 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.

--annotate

Annotate each hunk with the filename and hunk number.

--format=unified|context

Use specified output format.

--strip=n

Remove the first n components of pathnames in the output.

--addprefix=PREFIX

Prefix pathnames in the output by PREFIX. 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.

--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.

--remove-timestamps

Do not include file timestamps in the output.

-v, --verbose

Always show non-diff lines in the output. By default, non-diff lines are only shown when excluding a filename pattern.

--clean

Always remove all non-diff lines from the output. Even when excluding a filename pattern.

-z, --decompress

Decompress files with extensions .gz and .bz2.

--help

Display a short usage message.

--version

Display the version number of filterdiff.

--list

Behave like lsdiff(1) instead.

--grep

Behave like grepdiff(1) instead.

EXAMPLES

To see all patch hunks that affect the first five lines of a C file:

```
filterdiff -i '*.c' --lines=-5 < patch
```

To see the first hunk of each file patch, use:

```
filterdiff -#1 patchfile
```

To see patches modifying a ChangeLog file in a subdirectory, use:

```
filterdiff -p1 Changelog
```

To see the complete patches for each patch that modifies line 1 of the original file, use:

```
filterdiff --lines=1 patchfile | lsdiff | \
  xargs -rn1 filterdiff patchfile -i
```

To see all but the first hunk of a particular patch, you might use:

```
filterdiff -p1 -i file.c -#2- foo-patch
```

If you have a very specific list of hunks in a patch that you want to see, list them:

```
filterdiff -#1,2,5-8,10,12,27-
```

To see the lines of the files that would be patched as they will appear after the patch is applied, use:

```
filterdiff --as-numbered-lines=after patch.file
```

You can see the same context before the patch is applied with:

```
filterdiff --as-numbered-lines=before
patch.file
```

Filterdiff can also be used to convert between unified and context format diffs:

```
filterdiff -v --format=unified context.diff
```

SEE ALSO

lsdiff(1), grepdiff(1)

AUTHOR

Tim Waugh <twaugh@redhat.com>

Package maintainer