Using latexdiff with git

115

96

I'd like to be able to use `latexdiff` with git. At the moment I have an alias in my `.gitconfig` file for a word-based diff:

```
[alias]
 wdiff = diff --color-words
```

but is it possible to make an alias that will use `latexdiff` then compile the output and open the created dvi/pdf file?

Probably the best approach to this is using `git difftool`, but has anyone already made an appropriate script?

revision-control latexdiff git

share improve this question

edited Feb 10 '12 at 11:55
Paul Hiemstra
873 7 8

asked Aug 8 '10 at 4:15
Simon
1,642 4 18 25

+1 Interesting idea, I should try this. – David Z Aug 8 '10 at 20:34

add a comment

8 Answers

active oldest **votes**

50

The most complete and general solution today is probably git-latexdiff, here:

https://gitlab.com/git-latexdiff/git-latexdiff

The script internally checks out the full tree for the specified revisions, and calls latexdiff with the --flatten option (or can call latexpand), hence this works if the document is split into multiple .tex files.

The script contains many special-cases to make it work for as many cases as possible (it received contributions from more than 10 contributors, most of them being "make it work for my use-case too").

Note: I'm the main author.

share improve this answer

edited Jan 13 at 15:05

answered Aug 21 '13 at 11:18
Matthieu Moy
722 6 13

4 This answer deserves to be much higher up. git-latexdiff is a brilliant tool that does everything and requires only a `make install` to install and then you can simply `git latexdiff HEAD~1` – Azrael3000 Feb 21 '15 at 16:56

I did not try the alternatives, but this one worked smoothly right away. Nice work! – BenC Oct 19

'15 at 15:36

add a comment

69

Do you mean something like the following?

~/bin/git-latexdiff

```
#!/bin/bash
TMPDIR=$(mktemp -d /tmp/git-latexdiff.XXXXXX)
latexdiff "$1" "$2" > $TMPDIR/diff.tex
pdflatex -interaction nonstopmode -output-directory $TMPDIR $TMPDIR/diff.tex
evince $TMPDIR/diff.pdf
rm -rf $TMPDIR
```

~/.gitconfig

```
[difftool.latex]
        cmd = git-latexdiff \"$LOCAL\" \"$REMOTE\"
[difftool]
        prompt = false
[alias]
        ldiff = difftool -t latex
```

You can then use the diff by running `git ldiff HEAD~1` , for example.

share improve this answer

edited May 13 '16 at 17:53
Arun Debray
4,935 2 15 38

answered Aug 9 '10 at 19:30
amonakov
929 9 7

1 Brilliant! Thanks, it works perfectly. The only changes I made was to use batchmode instead of nonstopmode to suppress some of the output (can look at the log file if need be) and to replace the last line with trap $(rm -rf $TMPDIR) EXIT – Simon Aug 10 '10 at 2:44

4 Minor point: the script should trap signals to make sure `$TMPDIR` is deleted in case it receives a signal like SIGINT. – Christian Lindig Feb 23 '12 at 12:49

Any easy way to copy across auxiliary files, like bibtex files? – naught101 Mar 19 '13 at 4:44

4 Am I right in my understanding that this only really works for documents made from a single .tex file? – drfrogsplat Aug 29 '14 at 4:18

@drfrogsplat please see an answer below for the git-latexdiff that should also work for multiple files. – Azrael3000 Feb 21 '15 at 16:57

show 1 more comment

Since this question is frequently visited: Here is a blog-posting on a really great bash script here. The maintainer says:

28 Latexbatchdiff

> A bash script that utilizes the brilliant perl script “latexdiff” in a latex project that is maintained in git.
>
> It does support multiple pdf files, and it automatically creates the pdf.

You can get a 'diff' between every commit and your current version, easily. It is very helpful. Some classes seem to brake the code of the soul package, which is used to highlight the changes. I found it helpful to add to my preamble this code to prevent any difficulties:

```
\providecommand{\DIFadd}[1]{{\protect\color{blue}#1}} %DIF PREAMBLE
\providecommand{\DIFdel}[1]{{\protect\color{red}\protect\scriptsize{#1}}}
```

What's been added, is printed in blue, what has been deleted, in small red type.

share improve this answer

edited Nov 11 '16 at 19:53
Stefan Pinnow
10.7k 5 21 56

answered Feb 10 '12 at 11:35
Keks Dose
15.8k 3 43 74

Nice to see that so many people write scripts to this kind of job in all kinds of languages :) – Paul Hiemstra Feb 10 '12 at 11:48

Apparently the project was moved there. – BenC Oct 19 '15 at 9:40

1 lol the blog post references this thread. Meta. – lindhe Jan 3 '16 at 17:21

add a comment

There is another option which has become available since the answers here were written. I think that it will be preferred by some as it does not require any additional packages.

21 Beginning in version 1.0.1, `latexdiff` has come with a version of `latexdiff-vc` which supports `git`. Simply running

```
   latexdiff-vc [ latexdiff-options ] [ latexdiff-vc-options ] -r [rev1] [-r rev2] file1.te
```

will run `latexdiff` on the specified versions from its best guess of the version control system in use. The `--git` option specifies that it should assume git. This can also be specified by running `latexdiff-git`, which has `git` as the default VCS.

The option `--pdf` will run `pdflatex` (and `bibtex` if necessary) on the output.

See the `latexdiff` manual for more information.

share improve this answer

edited Oct 19 '15 at 18:57
Community ♦

answered May 10 '15 at 3:32
Nathan Musoke

1

321 2 9

add a comment

15

I recently created a rather nice Python script which works both on Mercurial and git repositories. It also supports bibtex and diffing against files that are not yet in the repository. See this blog post for more details. The code can be downloaded from bitbucket:

```
hg clone https://bitbucket.org/paulhiemstra/scm-latexdiff
```

or download a tar ball. Some examples of the usage of the script on the command line:

```
# for hg
scm-latexdiff 4:spam.tex
scm-latexdiff 4:spam.tex 6:spam.tex
# for git
scm-latexdiff 87213:spam.tex
scm-latexdiff 87213:spam.tex 97123:spam.tex
# You can also diff against non-commited (local) files
scm-latexdiff local:spam.tex
scm-latexdiff 2:spam.tex local:spam.tex
```

The script should work on all platforms that support Python. If you have any suggestions, or bug reports, you can submit an new issue in the issue tracker.

EDIT: A new version of the program now also supports multi-file latex documents. Especially for reports and books this can be important. Note that you need a recent version of latexdiff (2007) to be able to work with multi-file documents.

To install the program, clone the repository or download a tar ball and go to the directory where you extracted/cloned the tool and type (note that you probably need to be root for this to work):

```
python setup.py install
```

To install to a non-standard directory tree (e.g. in your home directory) use --prefix:

```
python setup.py install --prefix=/home/spam/
```

Do remember to add `/home/spam/lib/python2.x/site-packages/` to your `PYTHONPATH` environment variable.

share improve this answer

edited Feb 23 '12 at 10:46

answered Feb 9 '12 at 22:20

Paul Hiemstra
873 7 8

add a comment

This does not work in a docummemt with included files. I have a very inelegant python script (see below). Any ideas to improve it?

7

~/.gitconfig

```
ldiff = !latexdiff-git.py
```

latexdiff-git.py

```
#!/usr/bin/env python
# A git integration for latex diff
# Using git diff will be faster but more work.

import os
from os.path import join, split
import sys
import shutil
import logging
import tempfile
from optparse import OptionParser

if __name__ == '__main__':
    parser = OptionParser()
    parser.add_option("-o", "--output", dest="output", default='diff', help="name of the
    parser.add_option("-t", "--temp", dest="tmp_path", default=None, help="name of the te
    #parser.add_option("-1", "--one", dest="", default=None, help="Uses pdflatex in the o

    (options, args) = parser.parse_args()

    if not options.tmp_path:
        options.tmp_path = tempfile.mkdtemp()

    pdf = '%s.pdf' % options.output
    texout = '%s.tex' % options.output
    basedir = os.getcwd()
    dir1 = join(options.tmp_path, '1')
    dir2 = join(options.tmp_path, '2')

    if len(args) == 2:
        commit1 = args[0]
```

share improve this answer

edited May 13 '16 at 17:54
Arun Debray
4,935 2 15 38

answered Jan 2 '11 at 0:13
Hernan
152 2 7

1 You could talk a look at my Python solution I show in my post. It tries to split all functionality up into separate functions to create more easy to read and maintainable code. – Paul Hiemstra Feb 10 '12 at 11:57

add a comment

rcs-latexdiff tool can do that. It can make a latexdiff of different versions of a same file within a RCS repository. It also manages included files.

4 The basic usage is:

```
$ rcs-latexdiff [OPTIONS] filename old_commit new_commit
```

So for example,

```
$ rcs-latexdiff paper.tex HEAD~1 HEAD
```

creates an output file *diff.tex* that is the latexdiff of the paper.tex file for the two last revisions. Then, it's up to you to compile the output file using your favorite compiler.

So, in your case, if you want to get a pdf highlighting changes between submitted version of a paper and its camera-ready version, you could do:

```
$ rcs-latexdiff report.tex branch_submitted_version branch_camera_ready_version
$ rubber -d diff.tex
```

These commands create a diff.tex (highlighting differences between two branches) and rubber generates a pdf.

The ultimate goal of this tool is to support all the different RCS software. At the moment, it only supports SVN and Git. You can find install instructions on Github page. Feel free to use, reports bugs and contribute ;)

share improve this answer

edited Jan 20 '14 at 9:51
amn
248 1 8

answered Jul 23 '12 at 13:35
driquet
141 2

1 Thanks for telling us about your tool. Although it is definitely relevant, I noticed that you posted this answer *verbatim* in reply to another question, which is poor form and a mild form of advertising, which can come off as spam. Please tailor your replies so they address the specific question. – Ryan Reich Jul 23 '12 at 16:20

Done. This post try to answer the git related question and propose a way to generate a diff between two commits. I also edited the other post, based on SVN. Thanks for your comment ;) – driquet Jul 24 '12 at 8:01

Isn't this basically equivalent to the ten-line solution proposed by amonakov? The case which is difficult to handle with the current tools is multiple tex sources (think `include` / `input`). In my opinion you should focus on it if you are writing a tool for this task. – Federico Poloni Jul 24 '12 at 11:22

rcs-latexdiff manages included files. It looks recursively for includes and inputs, get the included content and replace the initial LaTeX instruction by the content. And it also works for other RCS than Git (only SVN at the moment ; Mercurial is coming). The idea was to use only one tool for all RCS software. At my office, people use either SVN, Git or Mercurial, so it's easier to explain once how to make a latexdiff for different revisions. – driquet Jul 24 '12 at 12:46

add a comment

use this in case you are on windows, no installment, just a simple `bat` script It works perfectly on windows10, miktex2.9:

1 https://github.com/redreamality/git-latexdiff

share improve this answer

answered Jan 30 at 3:18