

Contents

1	Word-based diffs for org-mode	1
1.1	Generating diffs	1
1.2	Generating the latexdiff	2
1.3	A "convenient" function	2

1 Word-based diffs for org-mode

When we write manuscripts we often want or need to see how one version differs from another. `latexdiff` is pretty good for this, and today we consider how to use it when your org-file is in git. It turns out we can use git to retrieve the old versions for comparison.

```
1 git hist
```

```
* 4ff3b06 Wed Jan 6 13:32:32 2016 | initial commit (HEAD -> master) [John Kitchin]
```

You can see above, at the time I ran the command there was one commit. As we develop the manuscript, we periodically export the result to see how it looks. At some point, all the changes should be committed *including the tex file*. We will use the different versions of the org and tex files to generate latexdiffs.

1.1 Generating diffs

Supposing you have a commit that represents some interesting point in the past, e.g. a submitted version, or the last edits made by a collaborator, here is how we generate a diff of the changes between the current version and that version.

First, check out your history. We will compare the current version to the commit `6f9a77a`.

```
1 git hist
```

```
* 6f9a77a Wed Jan 6 13:37:03 2016 | successful build. we will compare to this version
* 26fdf5c Wed Jan 6 13:34:20 2016 | new files added, tex and pdf [John Kitchin]
* 4ff3b06 Wed Jan 6 13:32:32 2016 | initial commit [John Kitchin]
```

```
1 git diff --color-words --word-diff=plain 6f9a77a -- latex-diff-org.org
```

1.2 Generating the latexdiff

Run this in a bash shell (it doesn't work for me in org-mode I suppose because it is using sh, not bash). I find latexdiff somewhat on the fragile side, for example you cannot make changes in section headings or the latexdiff result won't compile. So we ignore those commands below. Somewhat unfortunate, but livable I suppose.

```
1 latexdiff --exclude-textcmd="section,subsection" <(git show 6f9a77a:latex-diff-org.tex) latex-diff-org.tex > diff.tex
```

It works like this, however.

```
1 (shell-command "latexdiff --exclude-textcmd=\"section,subsection\" <(git show 6f9a77a:latex-diff-org.tex) latex-diff-org.tex > diff.tex")
```

0

Now compile the diff to get the pdf:

```
1 pdflatex -shell-escape diff.tex
2 pdflatex -shell-escape diff.tex
```

~~Here is the diff:~~

1.3 A "convenient" function

So, we might as well get ourselves a convenient function for that. Helm is my goto selector, so here we grab some candidates based on the commits.

```
1 (mapcar (lambda (s)
2         (let ((commit (nth 1 (split-string s)))
3               msg)
4             (string-match "|.*\\DIFdelbegin \\DIFdel{" s)
5             (cons (concat commit " " (match-string 0 s)) commit)))
6         (split-string (shell-command-to-string "git hist") "\\n"))
```

```
1 ((("6f9a77a | successful build. we will compare to this version later. (HEAD -> master) [John Kitchin]" . "6f9a77a")
2   ("26fdf5c | new files added, tex and pdf [John Kitchin]" . "26fdf5c")
3   ("4ff3b06 | initial commit [John Kitchin]" . "4ff3b06"))
```

```

1 (defun latexdiff (commit)
2   (let ((fname (concat (file-name-base (buffer-file-name)) ".tex")))
3     (shell-command (format
4                       "latexdiff --exclude-textcmd=\"section,subsection\" <(git show %s:%s) %s > diff.tex"
5                       commit fname fname))
6     (shell-command "pdflatex -shell-escape diff.tex")
7     (shell-command "pdflatex -shell-escape diff.tex")
8     (org-open-file "diff.pdf")))
9
10 (helm :sources '((name . "commits")
11                  (candidates . ,(mapcar (lambda (s)
12                                           (let ((commit (nth 1 (split-string s)))
13                                             msg)
14                                             (string-match "|.*$\\DIFdelend \\DIFaddbegin \\DIFadd{$" s)
15                                             (cons (concat commit " " (match-string 0 s)) commit)))
16                  (split-string (shell-command-to-string "git hist") "\n")))))

```

```

1 (("6f9a77a | successful build. we will compare to this version later. (HEAD -> master) [John Kitchin]" . "6f9a77a")
2  ("26fdf5c | new files added, tex and pdf [John Kitchin]" . "26fdf5c")
3  ("4ff3b06 | initial commit [John Kitchin]" . "4ff3b06"))

```

Here is the new function that calls helm to select a commit, and then runs our latexdiff function.

```

1 (defun latexdiff (commit)
2   "Export to LaTeX and do a latexdiff from the version in COMMIT."
3   (save-buffer)
4   (let* ((fname (org-latex-export-to-latex))
5          (cmd (format
6                "latexdiff --exclude-textcmd=\"section,subsection\" <(git show %s:%s) %s > diff.tex"
7                commit fname fname)))
8     (message cmd)
9
10    (shell-command cmd)
11    (shell-command "pdflatex -shell-escape -interaction nonstopmode diff.tex")
12    (shell-command "pdflatex -shell-escape -interaction nonstopmode diff.tex")
13    (org-open-file "diff.pdf")))
14
15 (helm :sources '((name . "commits")
16                  (candidates . ,(mapcar (lambda (s)
17                                           (let ((commit (nth 1 (split-string s)))
18                                             msg)
19                                             (string-match "|.*$\\DIFaddend " s)
20                                             (cons (concat commit " " (match-string 0 s)) commit)))
21                  (split-string (shell-command-to-string "git hist") "\n")))))
22          (action . (lambda (commit)
23                      (latexdiff commit))))))

```

~~git hist~~

It works, but it is probably limited to tex files in the root git directory.