

Git / Mercurial

Distributed Revision Control Systems

Architecture

- No Server/Client paradigm
- Complete history resides in one subdirectory (.git / .hg)
- Each checkout contains the complete history
- Advanced merging
 - ➔ Fast local transactions, network independence

Why to use Revision Control

- Light-heartedly work on code – you can always go back
- Ever deleted a file that you're “never going to need again”?
- Work on different machines
 - ➔ Personal use: Everything code, LaTeX (papers, proposals)

How to start

- Git or Mercurial? Virtually the same. You can use either, conversion tools (complete history) are available.
- Beginner? Use Git (helpful command line comments)
- Social coding:
 - <http://www.github.com/> (free public repositories)
 - <http://gitorious.org/> (free **private** repositories)
 - <http://www.bitbucket.org/> (free **private** repositories)
- Download and install:
 - Git: <http://git-scm.com/>
 - Mercurial: <http://mercurial.selenic.com/>

Create a Repository

- For empty and existing repositories:

```
mkdir project && cd project  
git/hg init
```

- Adding existing files:

```
git/hg add <files>
```

Committing

- Commits are local, thus fast
- Git only commits files, also existing files, previously add-ed. Commits all changes with the `-a` switch. Mercurial commits all changes unless specific files are specified.

```
git commit -a -m "Message"
```

```
hg commit -m "Message"
```


Working

- See status

```
git/hg status  
git/hg log  
git/hg diff  
...
```

- Add, move and remove files

```
hg/git add/mv/rm
```

- Ignore certain files: Write names as shell-patterns into:

```
.gitignore / .hgignore
```

Much much more

- Get cheat-sheets:
 - Git: <http://zrusin.blogspot.com/2007/09/git-cheat-sheet.html>
 - Mercurial: <http://ivy.fr/mercurial/ref/v1.0/>

Publish your repository

- **Don't** use your working copy, create a clone to pull from

```
git clone --bare project /share/git/project.git  
hg clone project /share/hg/project.hg
```

- Many serving possibilities (HTTP, HTTPS, Samba, NFS)
- For yourself: Just use SSH, no additional setup required

Fetching remote repositories

- Initially: Clone the repo from remote

```
git clone ssh://you@machine.ch/share/git/project.git  
hg clone ssh://you@machine.ch/share/hg/project.hg
```

- In the future: Pull and merge changes

```
git pull (or:  
git fetch && git merge  
hg pull && hg merge
```


Share your commits

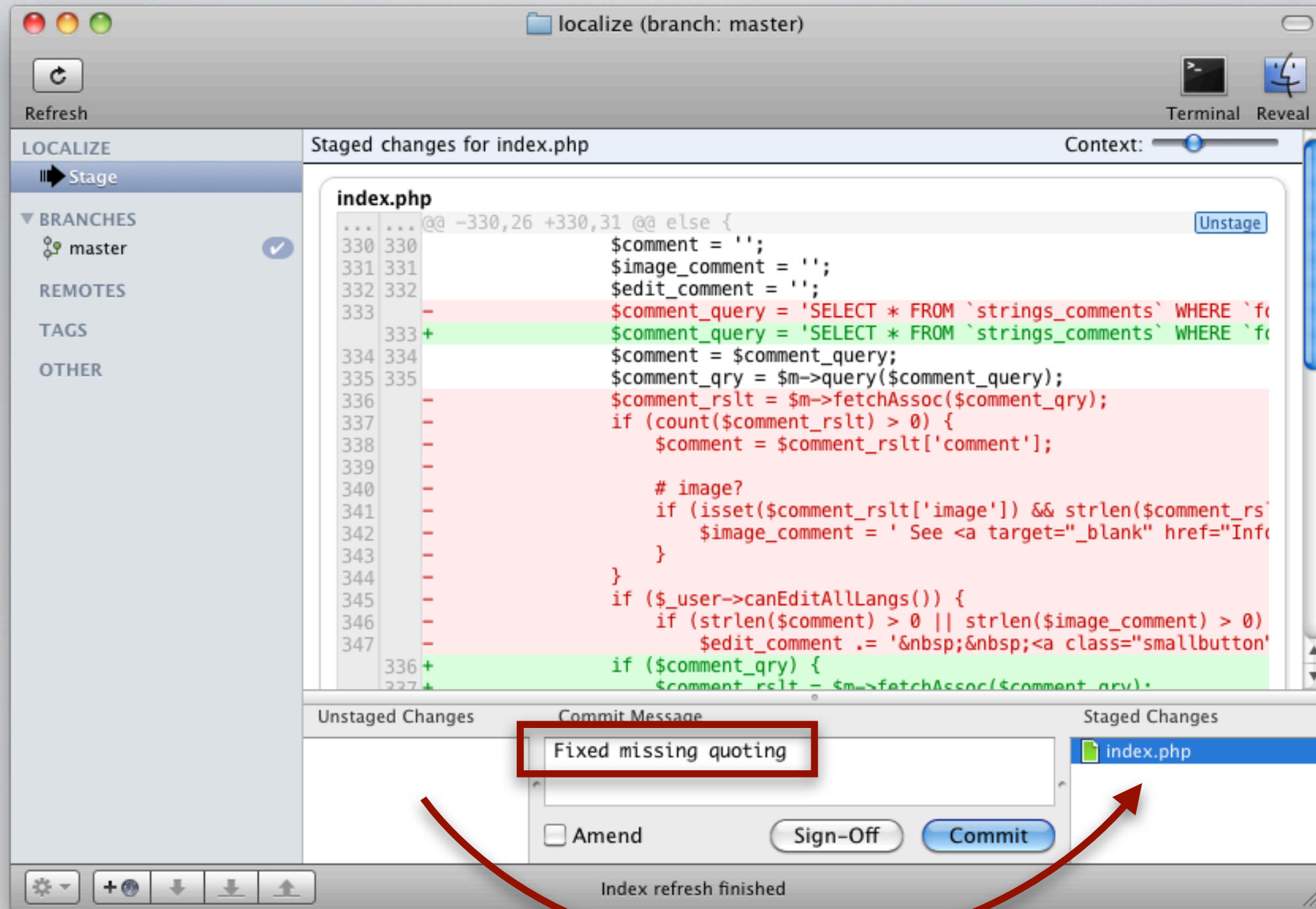
- After your commits are done, you push them. Both systems push a cloned repository to where it was cloned from

`git/hg push`

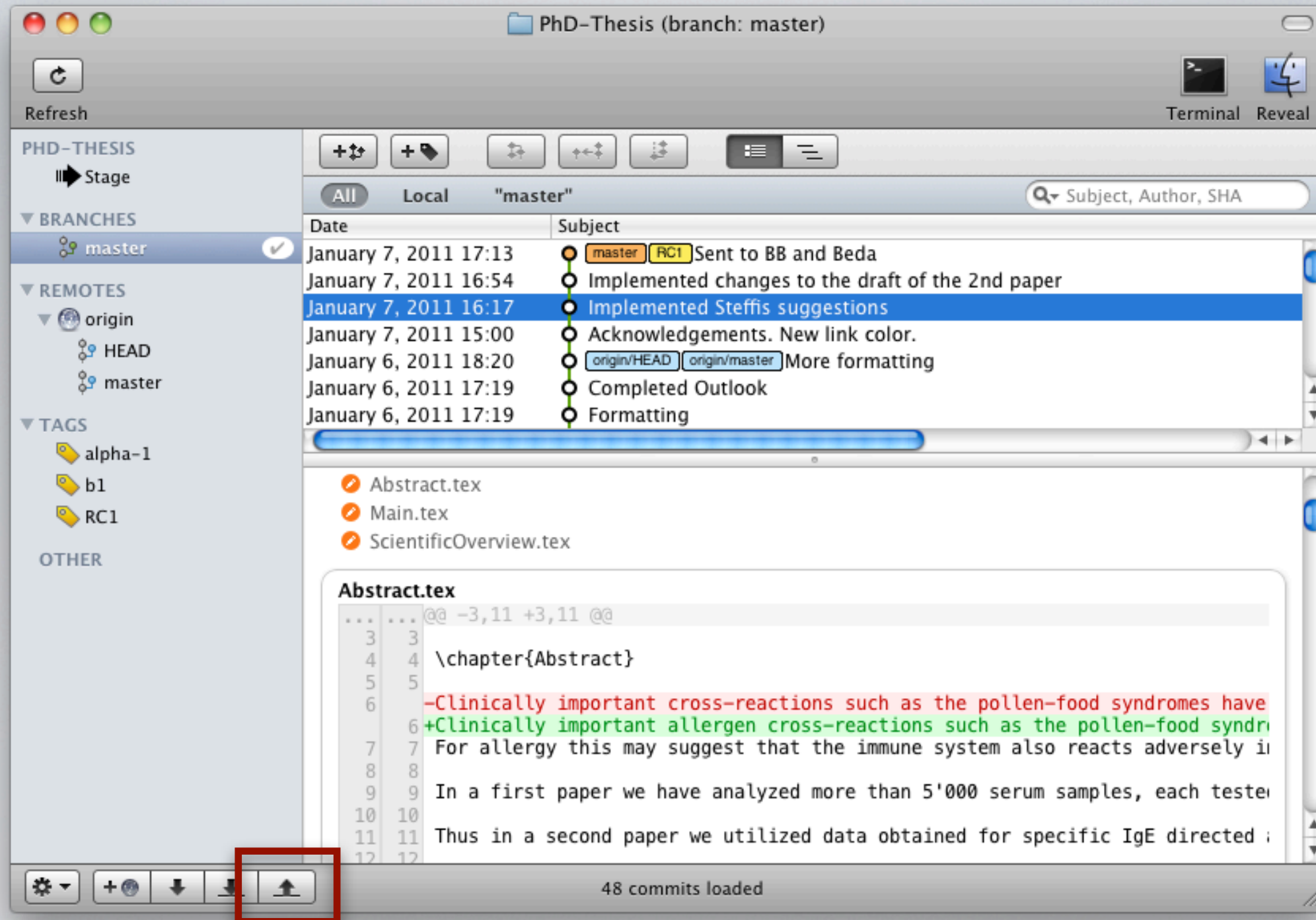
- But you can push to different locations

Graphical Interfaces

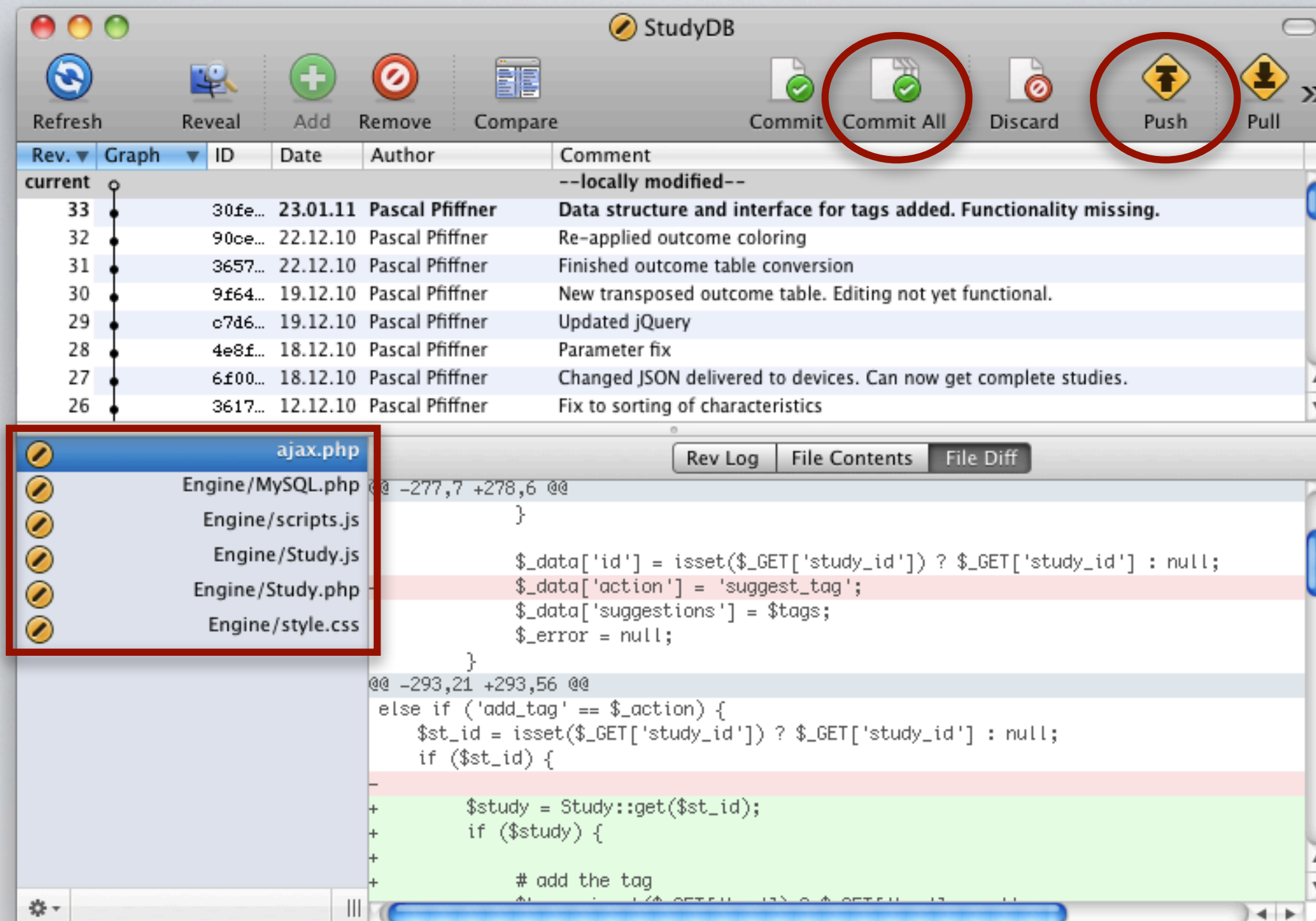
- Simplify many of the daily tasks
- No additional setup
- Best: Visually check your changes before committing
- My choice:
 - Git: GitX: <http://gitx.frim.nl/>
 - Mercurial: Murky: <https://bitbucket.org/snej/murky>



GitX: Stage commit



GitX: View history, push



Murky: Stage commit

Further Reading

- Check out advanced possibilities like branching, cherry-picking and blaming
- Documentation:
 - Git: <http://git-scm.com/documentation>
 - Mercurial: <http://hgbook.red-bean.com/>