

BensonMarguliesGitWorkflow

Here is my workflow for combining my preference for git as a day-to-day VCS with the official Lucene Subversion repository.

make a fork

I have a public github fork of the official github mirror. The mirror is at `git@github.com:apache/lucene-solr.git`, my fork is at `git@github.com:benson-basis/lucene-solr.git`.

make a clone

I have a clone. I cloned `git@github.com:apache/lucene-solr.git`, and then I did `git remote add fork git@github.com:benson-basis/lucene-solr.git`. To have access to both branches, I created a remote tracking branch called `fork-trunk`.

create a JIRA and describe the task

setup in git to work on the task

```
git checkout fork-trunk
git pull origin trunk
git checkout -b lucene-xxxx-some-description
```

Do the work

Create a github pull request from lucene-xxxx-some-description back to the mirror

Make sure that the JIRA # is in the PR title; this will cause the asf bot to pick up the PR and post an informative comment onto the JIRA.

Await review and act on review

Merge and Commit

I've found several scripts that purport to convert the output of `git diff` (or the `'patch'` URL from github) into the input of `svn patch`. You can find one at <https://gist.github.com/g-nix/7153630>. In my experience, `svn patch` can be very cranky, and has been known to decide, for example, that all of the changes are Conflicts. If you merge down on git you can more or less safely just copy the modified files to svn. Watch out for moves/renames.

Be sure to follow the instructions to include the sentence:

This closes #PP.

in the SVN commit log message where 'PP' is the number of the pull request. This causes an ASF bot to close out the PR.