UsingGit

Apache Nutch uses the Git version control system. There are two options

- https://gitbox.apache.org/repos/asf?p=nutch.git
- https://github.com/apache/nutch/

both are automatically mirrored, see the documentation on https://gitbox.apache.org/.

Checking out a copy of Nutch and modifying it

To check out a copy of Nutch, perform the following command:

```
git clone https://gitbox.apache.org/repos/asf/nutch.git master
```

Alternatively, using github:

```
git clone https://github.com/apache/nutch.git master
```

This will check out the trunk (master) copy of the code.

Once you have the code, you can modify a file, or two, then add the files for staging/commit, and then commit them. Once done, you can also decide to push the files up to the master repository if you have write access and are a member of the PMC and/or a committer. If you don't have write access to the repository, you can follow this guide for issuing pull requests that committers/PMC members can merge.

Here are the steps to achieve the above if you have commit access, assuming you have 2 files, `file1` and `file2` in the repository local copy that you modified.

1. `git add file1 file2`
2. `git commit -m "message describing change."`
3. `git push -u origin master`

Suggested User Contribution Workflow

Users contribute patches to Nutch using JIRA and/or Github. Let's take the use case for contributing via JIRA, then we'll tackle the Github one.

JIRA contribution

Grab the user's patch file, and then apply it locally. Before doing so, create the feature branch by following step 1 from the Developer workflow section. Then, assuming the patch file is named according to the conventions from step 6 in the Developer workflow section, perform the following steps (make sure you are on the NUTCH-xxx feature branch):

1. `git apply < NUTCH-xxx.<your last name>.<yyMMdd>.patch.txt`
2. Steps 3-4 from Developer workflow guide.
3. Final 2 steps from Developer workflow guide (aka the merge of feature branch step and then the push to origin trunk)

Github Pull-Requests

We accept pull requests on github. Please read the instructions in the pull-request template for more details.

If they are contributing using Github they are submitting a Pull Request for review you can easily merge their pull request into your local feature branch using Git. Assuming the Github user is "user01" and the branch they have created is "fix-nutch-stuff" (you can find this information in the Nutch Pull request, for example in this pull request #84, the username is `sujen1412` and branch name is `NUTCH-2157`), you can merge it into your local feature branch like so (again, make sure you are on the local feature branch for your issue, NUTCH-xxx first by typing `git checkout NUTCH-xxx` if you aren't already):

1. `git pull https://github.com/user01/nutch.git fix-nutch-stuff` (will find the remote github branch and merge locally)
2. test and make sure PR works, etc. Fix conflicts, etc.
3. Final 2 steps from Developer workflow guide (aka the merge of feature branch step and then the push to origin trunk)

Suggested Developer Workflow

As a longtime SVN user, it took me a while to figure out Git and realize its great support for use cases. Here is the one I most commonly use in Git that I think fits the Nutch workflow as a PMC member/committer quite well. Imagine that you are working on an issue, call it NUTCH-xxx. Either you have filed it yourself, or someone has filed it. My suggestion is to create a feature/working branch for that issue, named exactly after the JIRA issue, e.g., create a branch NUTCH-xxx. Commit your changes and modifications there. Then, after the changes are applied, tested, and verified, then you can switch back to the master branch, and push your local copy to the remote. If you'd like a review before pushing (e.g., attaching a patch you have two options). You can create a patch file by diffing the branches, and then attach that to the JIRA issue. This is common and already in use using Subversion in Nutch in years past. Let's take this use case to start. Inside of your nutch clone, do the following:

1. `git branch NUTCH-xxx`
2. modify files, change them, test, etc.
3. `git add <changed files>` (to see them, try `git status` or simply to stage **all** changes, `git add *`)
4. `git commit -m "Fixes for NUTCH-xxx contributed by <your first name> <your last name> <your email>"`

5. `git checkout trunk`
6. `git diff trunk..NUTCH-xxx > NUTCH-xxx.<your last name>.<yyMMdd>.patch.txt`
7. Attach the patch created in 6 to JIRA.

If you are **not** looking for a review or you have already had a review and are ready to commit the changes, push them:

1. (if you haven't already merge, the branch into master) `git checkout trunk && git merge NUTCH-xxx`
2. `git push -u origin trunk`

Migrating from an existing SVN checkout of Nutch (trunk) to Git

If you need to migrate from an SVN checkout of Nutch to Git, follow these instructions below.

1. `svn status` (ensure no local changes) 2. `mv .svn .svn.old` (or simply `find . -name "*.svn" -exec rm -rf {} \;`) 3. `git init` 4. `git remote add origin` https://gitbox.apache.org/repos/asf/nutch.git 5. `git checkout -b merge-branch` 6. `git fetch --all` 7. `git reset --hard origin/master` 8. `git checkout master`

And on my Nutch 2.x checkout the last two steps were changed to:

1. `git reset --hard origin/2.x` 2. `git checkout 2.x`