Workflow For Committers

Goal

- Keep a serialized, linear history of the master branch in the ASF repo.
- Each commit would have a dedicate JIRA ticket, unless the change is trivial.

Fork the ASF repo

1. `git clone https://git-wip-us.apache.org/repos/asf/incubator-quickstep.git quickstep`
2. `cd quickstep`
3. `git submodule update --init`
4. `cd third_party`
5. `./download_and_patch_prerequisites.sh`

NOTE

Always use "git rebase". Never use "git pull" or "git merge" as that changes the master history, and will produce unrelated commit messages for future PRs.

Work on a new feature

1. Create a JIRA ticket at https://issues.apache.org/jira/browse/QUICKSTEP. Let's assume the JIRA ticket is identified as: `QUICKSTEP-101`.
2. `git checkout master`
3. `git checkout -b quickstep-101`
4. `git add some-changed-or-new-files`
5. Then test your changes doing at least the following:
   - Check if your code confirms to the code guidelines by running the following command from the root of the quickstep source directory: `python third_party/cpplint/lint_everything.py`
   - Check for CMake validation errors by running the following command from the root of the quickstep source directory: `python validate_cmakelists.py`
   - Check for cyclic dependencies by running the following command from the root of the quickstep source directory: `python cyclic_dependency.py`
   - From the build directory, run all the tests using the command: `test -j8`
6. If you have made changes to the parser to add new tokens and/or grammar rules, please run the script `parser/genfiles.sh` and check the generated files into the directory `parser/preprocessed`.
7. `git commit -m "QUICKSTEP-101: Added My Awesome Feature."`
8. `git push origin quickstep-101`

Sometimes, there may be code review comments after a PR has been opened. Address these comments and then do the following:

1. `git add more-changed-or-new-files`
2. `git commit -m "Addressed Review Comments."`

Finally, once the PR is ready to be merged (i.e. it has passed all the tests and the python validation scripts as described above), we need to squash all the commits into one commit. We cherry pick the first commit, and squash all the rest as follows:

1. `git rebase -i origin/master`
2. In the editor, you should see something like:
   ```
   pick QUICKSTEP-101: Added My Awesome Feature.
   pick Addressed Review Comments.
   ```
3. While still in the editor, modify the second line to the following.
   ```
   s Addressed Review Comments.
   ```
4. While still in the editor, modify the squashed commit messages if needed.
5. Check the git log using: `git log`.
6. Check the "Author" field in the last commit with the intended name and email. Amend if needed.
7. Now you are ready to push the code. You can do that using: `git push origin quickstep-101 -f`

Create a Pull Request (PR) for the new feature

Use the GitHub ASF mirror repo: https://github.com/apache/incubator-quickstep, and set the title for the PR to match the title of the JIRA ticket.

In this case, `QUICKSTEP-101: Added My Awesome Feature.`

Merge a PR

To merge a PR, go through the following steps:

1. Update the local fork with the new branch `quickstep-101`.
   a. If the new branch is in the ASF repo, do `git fetch origin`
   b. Otherwise, we need to fetch from a remote fork repo. Below `some-contributor` is the github id of the contributor, and in the Quickstep github PR, the PR is assumed to have a tag like `some-contributor:quickstep-101`
      `git remote add some-contributor https://github.com/some-contributor/incubator-quickstep.git`
      `git remote -v`
      `git fetch some-contributor quickstep-101`
2. Ensure that the branch / PR is one commit ahead of the `master` branch, with a well-written commit message:
   a. If the new branch is in the ASF repo, do the following commands:
      i. `git checkout quickstep-101`
      ii. `git rebase -i origin/master`
      iii. `git push origin quickstep-101 -f`
   b. Otherwise, ask the contributor to do the same as above.
3. `git checkout master`
4. Rebase the PR with the origin master branch in the ASF repo.
   a. If the new branch is in the ASF repo, `git rebase -i origin/quickstep-101`
   b. Otherwise, `git rebase -i some-contributor/quickstep-101`
      Now if we see "noop", we are good to merge. Otherwise, the committer should notify the PR submitter to rebase the branch to make it a single commit ahead the current master branch.
5. `git log`
   If we see a commit message like "`Merge branch 'master' of https://git-wip-us.apache.org/repos/asf/incubator-quickstep`", we should abort the merge; otherwise we will mess up the master history.
6. `git push origin master`
   This command triggers `asfgit` to merge the PR in GitHub ASF mirror repo.
7. Delete the branch if it is in the ASF repo, otherwise skip this step.
   `git push origin --delete quickstep-101`
   This command triggers `asfgit` to delete the merged branch.
8. Close the JIRA ticket regarding this PR by visiting: https://issues.apache.org/jira/browse/QUICKSTEP