HowToCommit

Guide for Hive Committers

This page contains guidelines for committers of the Apache Hive project. (If you're currently a contributor, and are interested in how we add new committers, read BecomingACommitter)

New committers

New committers are encouraged to first read Apache's generic committer documentation:

- Apache New Committer Guide
- Apache Committer FAQ

The first act of a new core committer is typically to add their name to the credits page. This requires changing the source in https://github.com/apache/hive-site/blob/main/content/community/people.md

Review

Hive committers should, as often as possible, attempt to review Pull Requests submitted by others. Ideally every submitted PR will get reviewed by a committer within a few days. If a committer reviews a PR they've not authored, and believe it to be of sufficient quality, then they can merge the PR, otherwise the PR should be cancelled with a clear explanation for why it was rejected.

The list of open Pull Requests can be found here: Hive Open Pull Requests. This is ordered by time of creating. Committers should scan the list from bottom-to-top, looking for Pull Requests that they feel qualified to review and possibly merge.

Hive committers can not +1/Approve their own Pull Requests, i.e. you are allowed to commit/merge your own changes only if the Pull Request first receives a +1 vote from **another committer**. In the past this rule has typically been ignored when making small changes to the website (e.g. adding a new committer to the credits page), but you should follow the standard process for anything else.

Reject

Pull Requests should be rejected which do not adhere to the guidelines in HowToContribute. Committers should always be polite to contributors and try to instruct and encourage them to contribute better Pull Requests. If a committer wishes to improve an unacceptable change, then he/she drop review comments and ask the contributor to update.

PreCommit runs, and committing patches

1. Run Pre-Commit tests on a Pull Request before committing.
2. If the test run is clean (and there's a +1 from a committer), the Pull Request can be merged/committed.
3. Test runs may not be clean due to issues in the PR itself, or due to flaky tests. These issues must be fixed and **PR should not be committed until the run is clean.**

If a commit introduces new test failures, the preferred process is to revert the patch, rather than opening a new JIRA to fix the new failures.

Commit

When you commit/merge a Pull Request, please:

1. Ensure that the Pull Request has a +1 vote, and that **24 hours have elapsed since the first +1 vote was cast** on the Pull Request. Note that this rule appears in the Hive Bylaws. Do not ignore it.
2. Include the Jira issue id in the commit message, along with a short description of the change and the name of the contributor. Be sure to get the issue id right, as this causes Jira to link to the change in Git/Github.
   a. if contributor is you then add the following suffix to commit message "(<you>, reviewed by <reviewer>)". Example: "HIVE-123. Add awesomesauce to the optimizer. (jvs, reviewed by Ashutosh Chauhan)"
   b. if contributor is not you then add the following suffix to commit message "(<contributor> via <you>)". Example: "HIVE-123. Add awesomesauce to the optimizer. (Mike Brakestoner, reviewed by Ashutosh Chauhan)"
      i. Additionally: "Co-authored-by: Ayush Saxena <ayushsaxena@apache.org>" can be used to attribute any additional code contributors.

3. Resolve the issue as fixed, thanking the contributor and the reviewers. **Always set the "Fix Version"** at this point, but please only set a single fix version, the earliest release in which the change will appear. However, if a patch is backported to a point release (such as 1.0.2) then multiple fix versions should be set so that the automated release notes can list the Jira issue for the point release as well as the primary release.

Committing Documentation

Hive's official documentation is hosted at Github-Hive-Site. To commit major documentation changes you must raise a Pull Request to the hive-site repo.

Changes committed to the hive site repo will automatically get published on: https://hive.apache.org/

Backporting commits to previous branches

If a patch needs to be backported to previous branches, follow these steps.

1. Raise a Pull Request directed toward the target branch with the actual Jira Id/message & commit id.
2. If the build is green, the PR can be merged to the desired branch

Dialog

Committers/Contributors can hang out in the #hive channel in Apache Slack workspace for real-time discussions. However any substantive discussion (as with any off-list project-related discussion) should be re-iterated in Jira or on the developer list.

Note: Committers or individuals with Apache Id can directly join the #hive slack channel on Apache Workspace, any other individual if interested should drop a mail to hive dev mailing list with his email id and any existing member of the #hive apache channel should be able to send him the invite to join the group.

Instructions to add folks to ASF hive channel: https://infra.apache.org/slack.html