Committer Guide

Your first commit

Congratulations on becoming a Tez Committer. Once you become Tez Committer, you should have commit access to tez repository. You will need to clone https://gitbox.apache.org/repos/asf/tez.git to be able to commit to the Tez repo.

You should first create a JIRA ticket for adding yourself to the Tez Team List and do this as your first commit. To make the update, you will need to update "docs/pom.xml" to add your info. Once, you have followed the process listed below for committing patches, you will need to publish the updated docs to the website. The instructions of how to update Tez website are here.

General Committer Guidelines

Before Committing

- Before committing a patch of one JIRA, you should first get +1 from at least 1 other committer.
- Also, the patch to be committed should have been run through a Precommit Build and obtained a +1 (green light) on the JIRA by the build job. This will happen automatically if the JIRA status has been set to Patch Available.
    - The build job will add a comment to the JIRA after the patch has been tested.
- You should work on this repository which is writable https://gitbox.apache.org/repos/asf/tez.git
- Each patch requires a +1 from a committer. If the author of the patch is a committer, it requires a +1 from a different committer.

How to Commit

1. First you need to ask yourself, which branch you need to commit into ? Usually you should commit it to master, if there's one release based on another branch, you should also commit into that branch (use cherry-pick as below). If you are not sure about this , send a message to the Tez developer mailing list.
2. (No longer necessary for master branch) For each commit, you should also update the CHANGES.txt, including which release this JIRA will go in and add details to the INCOMPATIBLE CHANGES section if it is an incompatible change. Our convention is to put your change on the top of change list. CHANGES.txt should contain an entry in each version that the JIRA is committed to. e.g. If a patch is committed to master (0.8.1) and branch-0.7 (0.7.1), then CHANGES.txt in master will contain an entry under version 0.8.1 and 0.7.1. CHANGES.txt in branch-0.7 will contain the JIRA under the 0.7.1 change list. Put the new entry at the top of the change list.
3. Any patch that is applied and committed should have been previously uploaded to JIRA by the author. In certain scenarios, when fixing minor nits, the committer can make a change, upload the patch with the minor modification to JIRA and then commit the modified patch.

Usually you will create a separate branch for your JIRA you are working, you can also work on master directly if you think the JIRA is pretty simple. Here I assume you are working on a separate branch. e.g TEZ-100

```
git pull origin master // pull master each time before you commit
git branch TEZ-100
git checkout TEZ-100
... // work on jira, don't forget update CHANGES.txt
git commit -m 'TEZ-100. JIRA Title (<Contributor> via <committer id>)' // or just (<committerid>) if the
committer is the author. please follow this format of commit message
git log // check whether the commit is succeeded
git checkout master
git merge TEZ-100
git log // check whether the merge is succeeded and ensure that the log contains just the new commit. It should
not contain any "Merge" related messages.
git push origin master:master // push local master to remote master

// cherry-pick to another branch (example branch-0.5)
git log // copy the hash value of the commit you want to cherry-pick
git checkout branch-0.5
git cherry-pick -x ${HashValue}   // don't forget the "-x"
git log // check whether the cherry-pick is succeeded, you should see logs like this, the last line will only
show when you add "-x"
        // commit eb054c8cd26144d83b6b3d91c8d5d3dd882f6ae0
        // Author: Hitesh Shah <hitesh@apache.org>
        // Date:   Sat Oct 18 09:35:15 2014 -0700
        // TEZ-1683. Do ugi::getGroups only when necessary when checking ACLs. (hitesh)
        // (cherry picked from commit 83b0c3db9777dc4ebe76963571d4be4ce6985873)

// Resolve the conflict if the cherry-pick fails sometimes
// After you resolve the conflict, call the following command
git cherry-pick --continue
git push origin branch-0.5:branch-0.5 // push local branch-0.5 to remote branch-0.5
```

After Committing

Please remember to resolve the JIRA with the required Fix Version set. The Fix Version is based on the final branch of commit. For example, a commit to branch-0.5 will imply that the fix version is the next unreleased version of 0.5.x. A commit to master only implies that the Fix Version should be set to the next major release (for now, 0.x are considered major releases). (**Do not mark the JIRA as Closed. This will be done by the Release Manager after a release with the changes has been published**)

If it's incompatible change, mark it in hadoop flags. Don't use labels to mark this.

When you complete the committing, you'd better to check the repository https://gitbox.apache.org/repos/asf/tez.git whether the commit is pushed correctly.

Update the JIRA with the details to which branches the patch was committed. Please be sure to thank the author of the patch as well as any reviewers.

Setting various Version fields on JIRA

Affects Version

The affects version of a JIRA should be set to the released versions which the JIRA affects. If the JIRA represents an issue which was introduces in one of the active lines - the affects version should be set to the relevant unreleased version numbers.

Target Version

This should be set to the oldest unreleased version where the JIRA will be committed. This would imply all newer unreleased versions automatically get the fix.

Fix Version

This should be set to all versions to which the patch on the JIRA is committed.