Making a TEZ Release

How To Release Tez

GPG Keys Setup

Follow Apache guidelines on setting up your GPG keys and ensure that your fingerprint is updated at id.apache.org.
Also, append your keys to the KEYS file at https://dist.apache.org/repos/dist/release/tez/KEYS as well as the KEYS file at the top of the source tree.

```
gpg --armor --fingerprint --list-sigs <keyid>
gpg --armor --export <keyid>
```

Ensure that you publish your key at http://pgp.mit.edu/

Creating a Release Candidate

Create or Checkout the Release Branch

```
git checkout -b branch-x.y.z
```

Run Basic Release checks

```
mvn clean install -DskipTests=true
mvn clean apache-rat:check
```

Generate CHANGES.txt

Currently we don't keep CHANGES.txt in repo and only generate it on demand. To find previous example, search in previous release branch.

To generate CHANGES.txt, please pull information from JIRA and git log. To get all JIRAs of this release, use filter 'fixVersion = <version> AND resolution = resolved'. To get all commits of this release, you can find the last commit in last release in corresponding release branch and do a git diff against current release branch.

There are cases where information from JIRAs is inconsistent with that from git log (which shouldn't happen but unfortunately happens sometime...). For example, committer may forget to mark JIRA as resolved or set the fix version, or commit message contains wrong JIRA number. So it's better to check both side.

Several things need notice:

- Incompatible changes should be found by using filter "Hadoop Flags" = "Incompatible change". **Don't** search by labels 'incompatible' or something similar.
- Umbrella JIRA should be listed in individually (instead of getting mixed in non-umbrella jira)
- Changes are grouped by release version. If one patch gets committed into multiple branches, put this change in the oldest one. For example, if a patch is committed in branch 0.9.0 (current release), 0.8.6 (unreleased of 0.8 line), this change should be listed in the section of 0.8.6.

Update Version Number if Required

```
mvn versions:set -DnewVersion="x.y.z"
Modify CHANGES.txt to set the release date to when the vote will likely end ( +3 or 4
days from the vote start ).
If needed, add missing entries in CHANGES.txt for this release's ChangeLog using the
output from "git log --pretty=oneline"
Commit and push all the changes.
```

Create a Release Tag

```
git tag -a release-x.y.z-rc0 -m 'Tez x.y.z RC0'
git push --tags origin
```

Deploy the jars to Staging

Your .m2/settings.xml should have something along these lines:

```
<settings xmlns="http://maven.apache.org/SETTINGS/1.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/SETTINGS/1.0.0
                      http://maven.apache.org/xsd/settings-1.0.0.xsd">
  <servers>
    <server>
     <id>apache.staging.https</id>
     <username>XXXXXX</username>
     <password>YYYYYY</password>
    </server>
  </servers>
  <profiles>
    <profile>
      <id>gpg</id>
      <properties>
        <gpg.passphrase>XXXXXXX</gpg.passphrase>
        <gpg.executable>gpg2</gpg.executable>
      </properties>
    </profile>
  </profiles>
</settings>
```

To deploy jars, run:

```
mvn clean deploy -Psources,javadoc,sign -DskipTests=true
```

Log on to https://repository.apache.org and look at the Staging Repositories. "Close" the tez jars recently uploaded to get the temporary staging repo URL.

Create a Release tarball for the Source

```
mvn clean
git clean -f -x -d
cd ..
cp -R tez-git-x.y.z apache-tez-{x.y.z}-src (where tez-git-x.y.z is the tez code
directory)
COPYFILE_DISABLE=1 tar --exclude=.git -zcvf apache-tez-{x.y.z}-src.tar.gz
apache-tez-{x.y.z}-src
```

Sign the Release

```
gpg2  --armor --output apache-tez-{x.y.z}-src.tar.gz.asc --detach-sig
apache-tez-{x.y.z}-src.tar.gz
```

Generate MD5 checksums

```
md5sum apache-tez-{x.y.z}-src.tar.gz > apache-tez-{x.y.z}-src.tar.gz.md5
shasum -a 512 apache-tez-{x.y.z}-src.tar.gz > apache-tez-{x.y.z}-src.tar.gz.sha
```

or

```
openssl md5 apache-tez-{x.y.z}-src.tar.gz > apache-tez-{x.y.z}-src.tar.gz.md5
openssl dgst -sha512 apache-tez-{x.y.z}-src.tar.gz > apache-tez-{x.y.z}-src.tar.gz.sha
```

Create Binary Tarball as a helper for Users

Create a final binary tarball using the minimal and full tarballs:

- Create a top-level directory called apache-tez-{x.y.z}-bin
- Untar the contents of the minimal tarball into this directory.
- Copy the full tarball into the directory under apache-tez-{x.y.z}-bin/share/ and name the file tez.tar.gz
- Copy the LICENSE* files and NOTICE file from the full tarball and replace the files under apache-tez-{x.y.z}-bin/ which came from the minimal tarball.
- If the version is higher than 0.8.x, use the mvn javadoc command to generate the xml-based config documentation and copy the files under apache-tez-{x.y.z}-bin/conf/ with a .template suffix.
- Tar the top-level directory and do the necessary signing as well as the checksum creation as done for the src tarball.

```
#Define the following environment variables
TEZ_SRC_DIR (e.g. tez-src or the absolute path)
RELEASE_VERSION (e.g. 0.8.4)

cd ${TEZ_SRC_DIR};
mvn clean install -DskipTests
mvn site
cd ..


mkdir apache-tez-${RELEASE_VERSION}-bin
tar -C apache-tez-${RELEASE_VERSION}-bin -zxvf
${TEZ_SRC_DIR}/tez-dist/target/tez-${RELEASE_VERSION}-minimal.tar.gz
mkdir tmp-tez-full
tar -C tmp-tez-full -zxvf ${TEZ_SRC_DIR}/tez-dist/target/tez-${RELEASE_VERSION}.tar.gz
cp tmp-tez-full/LICENSE* apache-tez-${RELEASE_VERSION}-bin/
cp tmp-tez-full/NOTICE* apache-tez-${RELEASE_VERSION}-bin/
rm -rf tmp-tez-full


mkdir apache-tez-${RELEASE_VERSION}-bin/conf
# copy over tez templates into conf dir.
cp ${TEZ_SRC_DIR}/tez-api/target/site/apidocs/configs/tez-default-template.xml
apache-tez-${RELEASE_VERSION}-bin/conf/
cp
${TEZ_SRC_DIR}/tez-runtime-library/target/site/apidocs/configs/tez-runtime-default-tem
plate.xml apache-tez-${RELEASE_VERSION}-bin/conf/


# TODO: If TEZ-3322 is not fixed, modify the config files under
apache-tez-${RELEASE_VERSION}-bin/conf/ and add the Apache header


mkdir apache-tez-${RELEASE_VERSION}-bin/share
cp ${TEZ_SRC_DIR}/tez-dist/target/tez-${RELEASE_VERSION}.tar.gz
apache-tez-${RELEASE_VERSION}-bin/share/tez.tar.gz
COPYFILE_DISABLE=1 tar --exclude=.git -zcvf apache-tez-${RELEASE_VERSION}-bin.tar.gz
apache-tez-${RELEASE_VERSION}-bin
# Sign release and create checksums using commands similar to those called out earlier
for the source tarball
gpg2  --armor --output apache-tez-${RELEASE_VERSION}-bin.tar.gz.asc --detach-sig
apache-tez-${RELEASE_VERSION}-bin.tar.gz
openssl md5 apache-tez-${RELEASE_VERSION}-bin.tar.gz >
apache-tez-${RELEASE_VERSION}-bin.tar.gz.md5
openssl dgst -sha512 apache-tez-${RELEASE_VERSION}-bin.tar.gz >
apache-tez-${RELEASE_VERSION}-bin.tar.gz.sha512

#Remove unnecessary text from checksum files
sed -i "" s/"^.*= "// apache-tez-${RELEASE_VERSION}-bin.tar.gz.md5
sed -i "" s/"^.*= "// apache-tez-${RELEASE_VERSION}-bin.tar.gz.sha512
#Verify integrity of checksum files
```

Upload Artifacts for Release Vote

The artifacts (source tarball, binary tarball, checksums etc) need to be copied over to https://dist.apache.org/repos/dist/dev/tez/. (This is an SVN repo where the release artifacts need to be committed to). The artifacts should be created under a directory with the release number and RC number.

Release Voting process

Start a Vote

Call for a vote on the dev mailing list with something like this:

Email subject should be: [VOTE] Release Apache Tez-x.y.z RC0

I have created an tez-x.y.zrelease candidate rc0.
GIT source tag (r***)

https://git-wip-us.apache.org/repos/asf/tez/repo?p=tez.git;a=log;h=refs/tags/release-x.y.z-rc0 (git hash <commit hash>)
Staging site: https://dist.apache.org/repos/dist/dev/tez/tez-release-rc/tez-x.y.z-rc0/ (svn revision <revision>)
Nexus Staging URL:

PGP release keys (signed using <GPG KEY>) http://pgp.mit.edu:11371/pks/lookup?op=vindex&search=<GPG KEY>
KEYS file available at https://dist.apache.org/repos/dist/release/tez/KEYS

One can look into the issues fixed in this release at https://issues.apache.org/jira/**

Vote will be open for atleast 72 hours.

[ ] +1 approve
[ ] +0 no opinion
[ ] -1 disapprove (and reason why)

Ending a Successful/Failed Vote

Once the vote passes/fails, send out an email with subject like "[RESULT][VOTE] Apache Tez x.y.z rc0" to dev@tez.apache.org.

List +1s, -1s, 0s with clear distinctions between binding (PMC and committers) and nonbinding (general contributors) votes.

Canceling a Vote

For a cancelled vote due to any reason, change the subject to "[CANCEL][VOTE] Apache Tez x.y.z rc0"

After A Successful Vote

Release the staging repository

Log on to https://repository.apache.org and look at the Staging Repositories. "Release" the tez repository previously sent out in the vote thread, so that it is available for general consumption.

Copy the bits to distribution folder for Apache Tez

Copy the Release Candidate bits from https://dist.apache.org/repos/dist/dev/tez/ to https://dist.apache.org/repos/dist/release/tez/. Create a new directory with the release number and copy the artifacts into that directory (use an svn mv to retain history). If needed, update STABLE to point to the new release. This step need to be done by PMC, or PMC has to grant the permission of dist/release to release manager.

As an example

```
svn mv https://dist.apache.org/repos/dist/dev/tez/x.y.z-rc0
https://dist.apache.org/repos/dist/release/tez/x.y.z
```

Create the release tag

git tag -a rel/release-x.y.z -m 'Tez x.y.z'

git push --tags origin

Note that it takes 24 hours for the changes to propagate to the mirrors.
Wait 24 hours and verify that the bits are available in the mirrors before sending an announcement. Sometime it takes longer than 24 hours.

Update DOAP file for Tez

Update Tez_DOAP.rdf at the top of the source tree in the master branch to update the releases section for this new release.

Update website at tez.apache.org

Follow instructions from Updating the Tez Website for the mechanics of site update.

Items that should be updated on the website. A new link under releases for the new release. The link should go to a landing page that provides release-notes, release artifacts link (See link in the announcement email below) and links to javadocs for the various projects. Ensure that the Releases section as well as the "All Releases" page on tez.apache.org is updated with the contents for this release.

The javadocs should be generated under the correct branch so that they are tagged properly. To generate the javadocs, go to each project directory (that has user APIs) and run.

```
mvn clean javadoc:aggregate -DskipTests
mvn site
```

The javadocs would be created under target/site/apidocs inside each project directory. Spot check the javadocs for obvious errors. Create a new folder x.y.z under the releases directory in the svn site repo and copy release-notes and apidocs from all the projects into that directory. The website is built from the git repo but the javadocs etc are committed only to svn to reduce the size of the git repo.

Release Announcement

Send out Announcement to dev@tez.apache.org and user@tez.apache.org
Suject: [ANNOUNCE] Apache Tez x.y.z.

The Apache Tez team is proud to announce the release of Apache Tez version x.y.z

The Apache Tez project is aimed at creating a framework to build efficient and scalable data processing applications that can be modelled as data flow graphs.

<Describe the highlights of the release>

The release bits are at: http://www.apache.org/dyn/closer.lua/tez/x.y.z

We would like to thank all the contributors that made the release possible.

Regards,

The Tez Team

JIRA Updates

The followings have to be done by PMC since only PMC has the required permission.

Find all JIRAs that were fixed in this release (i.e. Fix Version set to the released version) and do a bulk edit to mark them all as Closed. A useful point to note is that when doing this bulk update, mail notifications should be turned off to avoid triggering a barrage of emails being sent out. Find all JIRAs with fix version x.y.z and not-resolved/not-closed and remove the fix version. Once this is complete, using the JIRA Administration tab, mark the x.y.z as released and set the appropriate release date. Create the next version if its already not done. Find all JIRAs with target version x.y.z and not-resolved/not-closed and change the target version to the next release following the above bulk update process.