HowToRelease

The release directory usually contains just three releasesThis page is to document the release procedure for Pig. Pig is a fairly young Apache project. Its release process is work in progress and is modeled from [Hadoop Release Procedure](#).

Note that only Pig committers can create a release.

Preparation

TODO:

1. Define issue management process like assigning/removing issues from release.

Creating Release Branch

We only branch for major (X.0.0) and minor(X.Y.0) releases but not for patches (X.Y.Z). Patch is an update to an existing branch created for X.Y.0.

1. Send email to `dev@pig.apache.org` to notify that you about to branch the tree and ask to hold off any commits till this is finished.
2. Update `CHANGES.txt` to include the release version and date (use `Unreleased` for the date if it is unknown) and remove `Trunk (unreleased changes)`. Below is the example of the top of the `CHANGES.txt` file after the update:

```
Pig Change Log

Release 0.1.1 - Unreleased

INCOMPATIBLE CHANGES

NEW FEATURES

IMPROVEMENTS
PIG-253: integration with hadoop-18

BUG FIXES
PIG-342: Fix DistinctDataBag to recalculate size after it has spilled. (bdimcheff via gates)
```

3. Commit these changes to trunk:

```
svn commit -m "Preparing for release X.Y.Z"
```

4. Create a branch for the release series:

```
svn copy https://svn.apache.org/repos/asf/pig/trunk \
https://svn.apache.org/repos/asf/pig/branches/branch-X.Y -m "Branching for X.Y releases"
```

5. Update CHANGES.txt to add back in Trunk (unreleased changes). Top of the `CHANGES.txt` should look like this now:

```
Pig Change Log

Trunk (unreleased changes)

   INCOMPATIBLE CHANGES

   IMPROVEMENTS

   OPTIMIZATIONS

   BUG FIXES

Release 0.1.1 - Unreleased
....
```

6. Update the default version in `build.xml` on trunk to X.(Y+1).0-dev.
7. Commit these changes to trunk:

```
svn commit -m "Preparing for X.(Y+1).0 development"
```

Updating Release Branch

The steps in this section are needed for all the releases (major, minor, and patches).

1. Check out the branch with:

```
svn co https://svn.apache.org/repos/asf/pig/branches/branch-X.Y
```

2. Run rat report and make sure that all files that can have apache license agreement. Save this rat report.

```
ant releaseaudit
```

3. For patches, update CHANGES.txt to include the release version and date with the new version. See Creating Release Branch section.
4. Update RELEASE_NOTES.txt for this release:
    a. Make sure to change all of the version number references to the current version number.
    b. Note highlights for this release. CHANGES.txt is a great place to find these.
    c. Note incompatibilities for this release. These should be listed under INCOMPATIBLE CHANGES in CHANGES.txt.
5. Update the version number in `build.xml` to be X.Y

6. ## Commit this changes:

```
svn commit -m "Preparing for release X.Y.Z"
```

7. Tag the release candidate:

```
svn copy https://svn.apache.org/repos/asf/pig/branches/branch-X.Y \
https://svn.apache.org/repos/asf/pig/tags/release-X.Y.Z-rcR -m "Pig X.Y.Z-rcR release."
```

Building

1. Build and run unit tests:

```
ant clean
ant jar test
```

2. Check that contrib and tutorial directories compile and tests pass:

```
cd contrib/piggybank/java
ant
ant test
cd ../../../tutorial
ant

cd ..
```

3. Build the source release and save the src.tar.gz artifact:

```
ant -Dversion=X.Y.Z src-release
```

4. Test the source tar file by unpacking the release and
    a. building pig.jar: `ant`
    b. building and running tutorial

```
cd tutorial
ant

cd ..
```

     c. build piggybank

```
cd contrib/piggybank/java
ant
ant test

cd ../../..
```

     d. running unit tests `ant test-commit`

5. Build the convenience artifacts:

```
ant clean
ant test
ant clean
ant jar
cd contrib/piggybank/java
ant
cd ../../..
ant -Dversion=X.Y.Z -Dforrest.home=<Forrest installation dir>  tar-h12
```

6. Generate the MD5 checksum of the release artifact and convenience binaries:

```
cd build
md5sum pig-X.Y.Z-src.tar.gz > pig-X.Y.Z-src.tar.gz.md5
md5sum pig-X.Y.Z.tar.gz > pig-X.Y.Z.tar.gz.md5
```

7. If you do not have a gpg key pair, do the following steps:
     a. Generating key pair using the following command. You can simply accept all default settings and give your name, email and Passphase.

```
gpg --gen-key
```

     b. Export your public key.

```
gpg --armor --output pubkey.txt --export 'Your Name'
```

     c. Open pubkey.txt, copy the full text and append it to the following files by pasting, then commit these changes:

```
https://svn.apache.org/repos/asf/pig/branches/branch-X.Y.Z/KEYS
https://svn.apache.org/repos/asf/pig/trunk/KEYS
```

     d. Upload updated KEYS to Apache.

```
scp KEYS people.apache.org:/www/www.apache.org/dist/pig/KEYS
```

     e. Export your private key, keep it with you.

```
gpg --export-secret-key -a "Your Name" > private.key
```

8. Sign the release artifact only (see Step-By-Step Guide to Mirroring Releases for more information).

```
gpg --armor --output pig-X.Y.Z-src.tar.gz.asc --detach-sig pig-X.Y.Z-src.tar.gz
```

9. Verify gpg signature.

```
gpg --import KEYS  (if necessarily)
gpg --verify pig-X.Y.Z-src.tar.gz.asc pig-X.Y.Z-src.tar.gz
```

10. Copy release artifact, convenience binaries, release notes and the rat report to a public place (usually into public_html in your home directory):

```
ssh people.apache.org mkdir public_html/pig-X.Y.Z-rcR
scp -p pig-X.Y.Z* people.apache.org:public_html/pig-X.Y.Z-rcR
cd ..
scp RELEASE_NOTES.txt people.apache.org:public_html/pig-X.Y.Z-rcR
scp <rat_report> people.apache.org:public_html/pig-X.Y.Z-rcR
```

11. Push proposed release to Maven staging area
    a. Create file ~/.m2/settings.xml with following contents (NOTE: It is highly recommended to use Maven's password encryption capabilities for your passwords.):

```
<settings>
  <servers>
    <server>
      <id>apache.releases.https</id>
      <username>your apache id</username>
      <password>your apache password</password>
    </server>
    <server>
      <id>apache.snapshots.https</id>
      <username>your apache id</username>
      <password>your apache password</password>
    </server>
    <server>
      <id>apache.staging.https</id>
      <username>your apache id</username>
      <password>your apache password</password>
    </server>
  </servers>

  <profiles>
    <profile>
      <id>apache-release</id>
      <properties>
        <gpg.passphrase>your GPG password</gpg.passphrase>
      </properties>
    </profile>
  </profiles>
</settings>
```

    b. Run ant command `ant mvn-deploy` to publish Pig artifacts to the apache snapshot repository.
    c. Run ant command `ant -Drepo=staging -Dversion=X.Y.Z mvn-deploy` to publish Pig artifacts to the apache staging repository.
12. Call a release vote. The initial email should be sent to `dev@pig.apache.org`. Here is a sample of email:

```
From: Olga Natkovich [mailto:olgan@yahoo-inc.com]
Sent: Tuesday, November 25, 2008 3:59 PM
To: dev@pig.apache.org
Subject: [VOTE] Release Pig 0.1.1 (candidate 0)

Hi,

I have created a candidate build for Pig 0.1.1. This release is almost identical to Pig 0.1.0 with a
couple of exceptions:

(1) It is integrated with hadoop 18
(2) It has one small bug fix (PIG-253)
(3) Several UDF were added to piggybank - pig's UDF repository

Keys used to sign the release are available at http://svn.apache.org/viewvc/pig/trunk/KEYS?view=markup.

Please download, test, and try it out:

http://people.apache.org/~olga/pig-0.1.1-rc0

Release notes and the rat report are available from the same location.

Should we release this? Vote closes on Wednesday, December 3rd.

Olga
```

Forward the initial email to `private@pig.apache.org` for Pig PMC members to vote.

Publish

Once three PMC members have voted for a release and a majority vote to release and the required 3 days have passed, it may be published.

1. Tag the release:

```
svn move https://svn.apache.org/repos/asf/pig/tags/release-X.Y.Z-rcR \
https://svn.apache.org/repos/asf/pig/tags/release-X.Y.Z -m "Pig X.Y.Z release."
```

2. Checkout Pig distribution svn.

```
svn co https://dist.apache.org/repos/dist/release/pig pig-dist
```

3. Copy release files to the distribution directory add them to svn.

```
cd pig-dist
cp -pr ~/release/pig-X.Y.Z-candidate-0 pig-X.Y.Z

svn add pig-X.Y.Z
```

4. The release directory should only contain the current releases, usually one or two (one is previous maintenance release, one is the most recent version release), with a link named 'latest' to the most recent version.

```
svn rm pig-A.B.C
svn rm latest
link -s pig-X.Y.Z latest
```

5. Commit .the distribution artifacts changes

```
svn commit --message "Pig X.Y.Z Release"
```

6. Check if svn changes propagates to the staging directory

```
ssh people.apache.org
cd /www/www.apache.org/dist/pig
```

After several minutes, you should see your distribution changes here.
7. Push Maven release from staging to production
   a. Go to https://repository.apache.org/index.html#view-repositories;staging~browsestorage
   b. Log in, using your Apache LDAP credentials. The sign in link is in the upper right hand corner.
   c. In the frame on the left side of the page, select "Staging Repositories", you should now see a list of artifacts in the main frame.
   d. Select the appropriate artifacts for this release and click "Close" on the bar above the list of artifacts.
   e. Select the appropriate artifacts for this release and click "Release" on the bar above the list of artifacts.
8. Wait 24 hours for release to propagate to mirrors.
9. Prepare to edit the website.

```
svn co https://svn.apache.org/repos/asf/pig/site
```

10. Update the front page news in author/src/documentation/content/xdocs/index.xml.
11. Update the release news in author/src/documentation/content/xdocs/releases.xml.
12. Update the documentation links in author/src/documentation/content/xdocs/site.xml
13. Copy in the release specific documentation

```
    mkdir publish/docs/rX.Y.Z
    cp -pr <releasedir>/docs/* publish/docs/rX.Y.Z/
    svn add publish/docs/rX.Y.Z
```

14. Regenerate the site, review it and commit in HowToDocument#UpdatingthePigSiteDocumentation.
15. Wait until you see your changes reflected on the Apache web site. This might take a few minutes.
16. Send announcements to the user and developer lists as well as (dev@pig.apache.org; user@pig.apache.org; announce@apache.org; general@hadoop.apache.org) once the site changes are visible. Note that emails sent to announce@apache.org must be sent from your apache.org email address and you must be subscribed to each of the lists.

```
[ANNOUNCE] Apache Pig X.Y.Z released

The Pig team is happy to announce the Pig X.Y.Z release.

Apache Pig provides a high-level data-flow language and execution framework for parallel computation on
Hadoop clusters.
More details about Pig can be found at http://pig.apache.org/.

The highlights of this release are ... The details of the release can be found at http://pig.apache.org
/releases.html.
```

17. In JIRA, mark the release as released.
    a. Goto JIRA and click on Administration tab.
    b. Select the Pig project.
    c. Select Versions.
    d. Select Release for the version you have released.
    e. If a description has not yet been added for the version you are releasing, select Add description and give a brief description of the release.
    f. If the next version does not exist (that is, if you are releasing version 0.x, if version 0.x+1 does not yet exist) create it using the Add box at the top of the page.
18. In JIRA, mark the issues resolved in this release as closed.
    a. Goto JIRA and click on the "Search for Issues" on "Issues" menu.
    b. In the left hand Edit section, set Project to Pig.
    c. In Status select "Resolved"
    d. In Resolutions select "Fixed"
    e. Click "Search" button
    f. In the next screen, further select fix For select the version you are releasing.
    g. Click on the "Search" button
    h. Select "Tools->Bulk change all XX issues" (near the top right)
    i. Select all the issues and click on "Next"
    j. Select "Transition Issues" radio button and click on "Next"
    k. Select "Close Issue" radio button and click on "Next"
    l. Uncheck the box near the bottom at says "Send mail for this update" lest you spam every Pig developer with a message for every bug resolved in this release. Click "Next".
    m. Click "Confirm". Don't worry if it gives you a HTTP 500 error, it still does the transitions.
19. Update jdiff for next release (step 16 to 19).

```
    svn co https://svn.apache.org/repos/asf/pig/trunk
```

20. Open build.xml. Change this line:

```
<property name="jdiff.stable" value="X.Y-1.Z"/>
To
<property name="jdiff.stable" value="X.Y.Z"/>
```

21. Copy jdiff comparison base to trunk

```
cp {releasedir}/src/docs/jdiff/pig_X.Y.Z.xml src/docs/jdiff
```

22.
```
svn add src/docs/jdiff/pig_X.Y.Z.xml
svn remove src/docs/jdiff/pig_X.Y-1.Z.xml
svn commit -m "Jdiff change for X.Y.Z"
```

TODO Need to integrate javadoc into this.

Post Release

1. Update the version number in `build.xml` in branch to be X.Y.N-SNAPSHOT, where N is one greater than the release being made.
2. Commit these changes:

```
svn commit -m "Preparing for X.Y.N release development"
```