How To Release

This page describes how to make a release of Whirr. It is based on a combination of the guide for Tapestry (thanks to the Tapestry team for sharing this!), and the Hadoop release process.

See also How To Contribute for more background on building Whirr.

TODO: These instructions need updating to reflect http://apache.org/dev/release-signing.html regarding i) key length, ii) signature strength.

First time as a release manager.

1. Generate PGP code signing keys
2. Ensure that your PGP signing keys are available in: http://svn.apache.org/repos/asf/whirr/dist/KEYS More details can be found here.
3. Copy the new KEYS file to the release folder /www/www.apache.org/dist/whirr on people.apache.org

If you are not already a member of the Web Of Trust (WOT) it would be a good idea to do so. (contact one of the prior release managers, e.g. Tom, Patrick, etc...). You can read more about key signing here.

You need to be a member of the "incubator" unix group on people.apache.org. Ask for help on the incubator general@ list. Here's an example http://bit.ly/9Wkdzg

Ensure that you have setup your ssh keys on people.apache.org, otherwise you'll have to enter your login password a number of times (best use ssh-agent for this as well). A good overview of this process can be found here (ssh-copy-id and ssh-agent in particular)

Create a Release Series Branch

TODO: Update to use git the next time a new release series is created.

This only needs doing if this is the first release in a series (X.Y.0).

- Update CHANGES.txt in trunk to replace `Trunk (unreleased changes)` with `Release X.Y.0 - YYYY-MM-DD`. Commit:

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

- Create a branch for the release series:

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

- Add back `Trunk (unreleased changes)` to CHANGES.txt in trunk.
- Bump the version number in trunk (the update-versions script mangles the whitespace in the root XML element):

  ```
  for file in $(find . -name pom.xml); do
    sed -i "" -e "s/0.1.0-SNAPSHOT/0.2.0-SNAPSHOT/" $file;
  done
  ```

- Bump the version in src/site/xdoc/index.xml.
- Bump the version in src/site/xdoc/whirr-in-5-minutes.xml.
- Bump the version in src/site/site.xml.
- Commit these changes to trunk.
- Checkout the release branch

  ```
  svn checkout https://svn.apache.org/repos/asf/whirr/branches/branch-X.Y
  ```

Update the Release Branch

The version number in the release branch ends in `-SNAPSHOT`, but we need to remove this for the release. For example, `0.1.0-SNAPSHOT` needs to be changed to `0.1.0`.

```
for file in $(find . -name pom.xml); do
  sed -i "" -e "s/0.1.0-SNAPSHOT/0.1.0/" $file;
done
```

- Fix the version numbers in src/site/xdoc/index.xml, src/site/xdoc/whirr-in-5-minutes.xml, src/site/site.xml.

Run Tests

Run all the unit and integration tests (which means you should have cloud credentials to run on a cloud provider) with the following:

```
mvn clean verify -Pintegration -Posgi -DargLine='-Dwhirr.test.provider=<cloud-provider> -Dwhirr.test.
identity=<cloud-provider-user> -Dwhirr.test.credential=<cloud-provider-secret-key>'
```

Generate the Release Notes

JIRA has the ability to generate release notes automatically (this is why it is so important to keep the fix version number accurate).

https://issues.apache.org/jira/secure/ConfigureReleaseNote.jspa?projectId=12311110

Manually check this list for accuracy! I've repeatedly seen closed bugs that were not fixed (i.e., duplicate) marked with a fix version, so that they incorrectly show up in this list. Find those, edit them to remove the fix release (only actually fixed bugs should have a fix release) and re-run the report.

Select the correct version, in HTML format.

Paste the notes to the top of *src/site/xdoc/release-notes.xml*. Check this into trunk and the branch.

Wrap the title ("Release Notes - Whirr - Version X.Y.Z") inside an <h1> element.

Build and Deploy Artifacts

Create a maven settings file *~/.m2/settings.xml* with the following content:

```
<settings>
<servers>
<server>
     <id>apache.snapshots.https</id>
     <username>APACHE-ID</username>
     <password>APACHE-PASSWORD</password>
   </server>
   <server>
     <id>apache.staging.https</id>
     <username>APACHE-ID</username>
     <password>APACHE-PASSWORD</password>
   </server>
   <server>
     <id>apache.releases.https</id>
     <username>APACHE-ID</username>
     <password>APACHE-PASSWORD</password>
   </server>
</servers>
</settings>
```

Build the artifacts:

```
mvn site -Posgi
mvn package assembly:assembly -Posgi
```

The following command deploys the artifacts, checksums, and signatures (you will need to enter a GPG passphrase) to the Apache Staging repo.

```
mvn deploy -Ppackage -Pdeploy -Pjavadoc -Posgi
```

If this step fails with an *Access denied* error check that you have the required permissions on Nexus.

Login to https://repository.apache.org using your Apache credentials. Click on **Staging** on the left. Then click on **org.apache.whirr** in the list of repositories. In the panel below you should see an open repository that is linked to your username and IP. Select this repository and click **Close**. This will close the repository from future deployments and make it available for others to view.

Copy Release Artifacts

The artifacts that end up in the distribution directory are the source and binary distributions (along with their checksums and signatures), so they need to be copied from the Maven repo to the release directory on people.apache.org so folks can vote on them:

TODO: update the instructions, the artifacts should be placed in a subdirectory rather than all at the toplevel

```
ssh people.apache.org
VERSION=X.Y.Z
mkdir ~/public_html/whirr-$VERSION-candidate-0
cd ~/public_html/whirr-$VERSION-candidate-0
for ext in "" .asc .md5 .sha1; do
  wget --no-check-certificate https://repository.apache.org/content/repositories/orgapachewhirr-[YOUR
REPOSITORY ID]/org/apache/whirr/whirr/$VERSION/whirr-$VERSION-src.tar.gz$ext
done
```

Now we need to create the binary distribution checksums and signatures (since Maven doesn't currently do this - if anyone know how to fix this, then please let us know!):

```
VERSION=X.Y.Z
cd target
md5 whirr-$VERSION.tar.gz > whirr-$VERSION.tar.gz.md5
shasum whirr-$VERSION.tar.gz > whirr-$VERSION.tar.gz.sha1
gpg --armor --output whirr-$VERSION.tar.gz.asc --detach-sig whirr-$VERSION.tar.gz
scp -p whirr-$VERSION.tar.gz* people.apache.org:public_html/whirr-$VERSION-candidate-0
```

Sanity Check

Download the artifacts and try out some of these things:

- Whirr in 5 minutes or the quick start instructions
- Check the MD5 checksums
- Licensing check: (download RAT and run `java` ~~`jar apache-rat`~~`*.jar` on the Whirr tarballs. Known violations include generated files (e.g. docs), .confluence files, .json files, and a few plain text files like CHANGES.txt.

Commit and Tag

Commit changes in the branch and create a release tag:

```
git commit -am "Preparing for release X.Y.Z"
git push origin branch-X.Y:branch-X.Y
git tag -a release-X.Y.Z -m "Whirr X.Y.Z release"
git push origin release-X.Y.Z
```

Run the Vote

Run the vote on the dev@whirr.

Here is an example email:

```
To: "Whirr Developers List" <dev@whirr.apache.org>
Subject: [VOTE] Release Whirr version X.Y.Z

This is the first release candidate for Apache Whirr, version X.Y.Z.

It fixes the following issues:
https://issues.apache.org/jira/secure/ReleaseNote.jspa?version=12315111&styleName=Html&projectId=12311110

*** Please download, test and vote by [3 working days after sending].

Note that we are voting upon the source (tag), binaries are provided for convenience.

Source and binary files:
http://people.apache.org/~[YOUR USERID]/whirr-X.Y.Z-candidate-0

Maven staging repo:
https://repository.apache.org/content/repositories/orgapachewhirr-[YOUR REPOSITORY ID]

The tag to be voted upon:
https://git-wip-us.apache.org/repos/asf?p=whirr.git;a=tag;h=[HASH]

Whirr's KEYS file containing PGP keys we use to sign the release:
http://svn.apache.org/repos/asf/whirr/dist/KEYS
```

The release needs 3 +1 votes from the PMC.

Roll Out

Assuming the vote passes, the release can be rolled out as follows:

Move Artifacts into Place

This step makes the artifacts available on the mirrors.

```
ssh people.apache.org
VERSION=X.Y.Z
PREVIOUS_VERSION=X.Y.Z-1
CANDIDATE=C
cp -r ~/public_html/whirr-$VERSION-candidate-$CANDIDATE /www/www.apache.org/dist/whirr/whirr-$VERSION
cd /www/www.apache.org/dist/whirr
chgrp -R whirr whirr-$VERSION
chmod -R g+w whirr-$VERSION
rm stable
ln -s whirr-$VERSION stable
rm -r /www/www.apache.org/dist/whirr/whirr-$PREVIOUS_VERSION # remove older versions
```

The last line is to remove the *previous* version, since only the most recent version on a particular branch should be in the dist directory (older versions are archived automatically, see http://archive.apache.org/dist/incubator/whirr/ and http://www.apache.org/dev/mirrors.html).

Log in to https://repository.apache.org, click on **Staging** on the left. Select the repository that you closed earlier, and click **Release**, using a description like "Apache Whirr X.Y.Z artifacts". This will make the artifacts publicly available.

Build and deploy documentation

TODO: this section needs updating following the change to svnpubsub in WHIRR-685. See the "Build and Deploy Site" section below for guidance.

```
mvn site-deploy
```

Add the Next Release to JIRA

Add the next version number (e.g. 0.2.0 after 0.1.0) to JIRA using this link: https://issues.apache.org/jira/secure/project/ManageVersions.jspa?pid=12311110.

In JIRA mark the released version as "released" on the "manage versions" page. Be sure to fill in a date if not already specified.

Wait 24 Hours

It takes up to 24 hours for all the mirrors to sync, so don't announce the new release just yet.

Build and Deploy Site

Update the top level site by checking out https://svn.apache.org/repos/asf/whirr/site (note that this is still in svn) and editing src/site/site.xml to update the version number. Then type:

```
mvn site-deploy
```

Open the locally-generated website in your browser and check it looks OK. When you are happy with it, type

```
cp -r /tmp/whirr.apache.org/ production/
```

and commit the changes. The updated website will be published to http://whirr.apache.org immediately (it uses svnpubsub).

Announce the Release

TODO: Add a news section to the website.

Send an email to announce@apache.org (the from: address must be @apache.org). E.g.

```
To: announce@apache.org, user@whirr.apache.org, dev@whirr.apache.org
Subject: [ANNOUNCE] Apache Whirr 0.1.0 released

The Apache Whirr team is pleased to announce the release of Whirr 0.1.0.

This is the first release of Apache Whirr, a set of libraries for
running cloud services such as Apache Hadoop, HBase, ZooKeeper, and
Cassandra.

The release is available here:
http://www.apache.org/dyn/closer.cgi/whirr/

The full change log is available here:
https://issues.apache.org/jira/browse/WHIRR/fixforversion/12315111

We welcome your help and feedback. For more information on how to
report problems, and to get involved, visit the project website at
http://whirr.apache.org/

The Apache Whirr Team
```