Index (Copy)

Setting up the signing keys

Before you do a release you'll need a signing key that is registered with apache. If you already have one, you can skip this section. Otherwise, here are the steps to do:

1. use **gpg2 --gen-key** to generate a new key. Make sure that the key size is 4096 bits, the key doesn't expire, and use CODE SIGNING KEY for the comment.
2. Now you need to register your key at http://pgp.mit.edu/ using the output of **gpg --armor --export <keyid>**
3. Add your key to the KEYS file:
    a. svn co https://dist.apache.org/repos/dist/release/zookeeper/ zookeeper_dist
    b. cd zookeeper_dist.
    c. edit the KEYS file. See the top of the KEYS file for instructions on how to edit this file
    d. svn ci
4. Upload digest to your account on id.apache.org (gpg2 --fingerprint)

Important Notes

- when updating JIRA be sure to use the "batch change" operator under the "tools" menu. Disable email notifications when using batch change, subsequent to large JIRA operations (moving issues btw releases, closing JIRAs after a release, etc...) send email to the dev@ list detailing the changes.

Branching

Skip this section if this is NOT the first release in a series (i.e. release X.Y.0).

1. Notify developers on the mailing list that you are about to branch a release.
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)**.
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/zookeeper/trunk https://svn.apache.org/repos/asf/zookeeper
    /branches/branch-X.Y -m "Branching for X.Y releases"
    ```

5. Update **CHANGES.txt** to add back in Trunk (unreleased changes).
6. Update the default version in **build.xml** on trunk to X.Y+1.0-SNAPSHOT.
7. Update the version number in **src/c/configure.ac** and **src/c/include/zookeeper_version.h** to be X.Y+1.0.
8. Commit these changes to trunk.

    ```
    svn commit -m "Preparing for X.Y+1.0 development"
    ```

Updating the release branch

These operations take place in the release branch:

1. Check out the branch with:

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

    git clone -b branch-X.Y https://git-wip-us.apache.org/repos/asf/zookeeper.git
2. If not already done, merge desired patches from trunk into the branch and commit these changes. You can find the revision numbers using **svn log CHANGES.txt** in the branch and in trunk.

```
cd branch-X.Y
svn merge -rR1:R2 ../trunk .
svn commit -m "Merge -r R1:R2 from trunk to X.Y branch. Fixes: ZOOKEEPER-A, ZOOKEEPER-B."
```

3. Do a final pass regeneration of docs by running "ant -Dforrest.home=<path to Forrest> clean docs". If there are any modifications, then they'll show up under the docs folder. If there are any changes, then commit them. **However, DO NOT commit changes in docs/releasenotes.html or docs/releasenotes.pdf.** (Release notes are handled in a separate step.) Release notes should be regenerated each time a documentation patch is committed. This final pass before the release is intended to catch mistakes in case this step was missed at commit time.
4. Update **CHANGES.txt** to include the release version and date (this change must be committed to trunk and any intermediate branches between trunk and the branch being released).
   Also, update CHANGES.txt with the committed jira details targeted for this release version. As we follow PR merging, jira info is not updated in this file while committing.
5. Update **docs/releasenotes.html** with release notes for this release. You can get the HTML by following the "Release Notes" link for the relevant release on the https://issues.apache.org/jira/browse/ZOOKEEPER?report=com.sourcelabs.jira.plugin.portlet.releases:releases-projecttab#selectedTab=com.atlassian.jira.plugin.system.project%3Aroadmap-panel tab in Jira. Note that you need to exclude the won't fix or invalid tickets.
6. **TODO - document howto generate the release notes using the new mechanism**
7. Update the version number in **build.xml** to be "X.Y.Z"
8. Update the version number in **src/c/include/winconfig.h**, **src/c/configure.ac**, and **src/c/include/zookeeper_version.h** to be "X.Y.Z"
9. Update the copyright years in NOTICE.txt if it's outdated.
10. Commit these changes.

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

    git commit -a -m "Preparing for release X.Y.Z"
    git push <remote> <branch>
11. Tag the release candidate (R is the release candidate number, and starts from 0):

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

    git tag -s release-X.Y.Z-rcR -m "ZooKeeper X.Y.Z-rcR release."
    Push the newly created rc tag to the remote repo.
    git push <remote> release-X.Y.Z-rcR

Building

1. Build the release & run unit tests.

   ```
   ant test package tar
   ```

2. Untar the generated zookeeper-X.Y.Z.tar.gz and do the following:
   a. Sign the toplevel zookeeper-X.Y.Z.jar

      ```
      gpg --armor --output zookeeper-X.Y.Z.jar.asc --detach-sig zookeeper-X.Y.Z.jar
      ```

   b. Sign each file in dist-maven

      ```
      gpg --armor --output file.asc --detach-sig file
      ```

3. Tar/GZ the release directory as zookeeper-X.Y.Z.tar.gz

   ```
   tar -caf zookeeper-X.Y.Z.tar.gz zookeeper-X.Y.Z
   ```

4. Generate the MD5/SHA1 checksums for the release artifact.

   ```
   md5sum zookeeper-X.Y.Z.tar.gz > zookeeper-X.Y.Z.tar.gz.md5
   sha1sum zookeeper-X.Y.Z.tar.gz > zookeeper-X.Y.Z.tar.gz.sha1
   ```

5. Sign the release

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

6. Check that release file looks ok - e.g. install it and run examples from tutorial.
   a. untar the release artifact in a test directory
7. Copy release files to a public place.

   ```
   ssh people.apache.org mkdir public_html/zookeeper-X.Y.Z-candidate-0
   scp -p zookeeper-X.Y.Z.tar.gz* people.apache.org:public_html/zookeeper-X.Y.Z-candidate-0
   ```

8. Stage the release artifacts to the Apache Nexus repository
   a. Create ~/.m2/settings.xml introducing the values that are specific to you:

      ```
      <settings>
        <servers>
           <server>
             <id>apache.snapshots.https</id>
             <username>XXXXXXX</username>
             <password>XXXXXXXX</password>
           </server>
           <server>
             <id>apache.staging.https</id>
             <username>XXXXXXX</username>
             <password>XXXXXXX</password>
           </server>
         </servers>
         <profiles>
           <profile>
             <id>gpg</id>
             <properties>
               <gpg.passphrase>XXXXXXX</gpg.passphrase>
             </properties>
           </profile>
         </profiles>
      </settings>
      ```

      See http://maven.apache.org/guides/mini/guide-encryption.html for details on encrypting the passwords rather than leaving them in the clear.

   b. Run the following on the branch check out

      ```
      ant mvn-deploy -Dstaging=true
      ```

   c. Enter Apache Nexus and do the following:
      i. Click on Log In in the upper right corner. Log in using your apache user name and password.
      ii. In the left navigation pane, select Staging Repositories.
      iii. Identify the release candidate you just pushed, by your user name (in parentheses as part of the "Repository" name) and the "Created On" date. Click on the check box to the left of your Repository name to select it. (If you accidentally click on the Repository name itself, another tab will pop open. If so, just close it.)
      iv. Click the Close button above the Repository names. This makes your release candidate available at the Staging level.
      v. If you have previously staged an older release candidate with the same version number, and it is still showing in the Repository list, you must select and Drop the old one now.
      vi. Confirm that your new release candidate is visible at https://repository.apache.org/content/groups/staging/org/apache/zookeeper/zookeeper/ with correct file modification dates.
9. Call for a release vote on **dev** (note dev@ and not user@, the user list is for discussion of released software only) at zookeeper.apache.org. Here is a sample email (from 3.4.6 release):

   In the below release candidate dev mail format, we need to change the 'The tag to be voted upon:' section like,

   "The RC tag in git to be voted upon: release-X.Y.Z-rcR"

```
Subject: [VOTE] Apache ZooKeeper release 3.4.6 candidate 0

This is a bugfix release candidate for 3.4.6. It fixes 117 issues, including issues that
affect leader election, Zab, and SASL authentication.

The full release notes is available at:

https://issues.apache.org/jira/secure/ReleaseNote.jspa?projectId=12310801&version=12323310

*** Please download, test and vote by March 9th 2014, 23:59 UTC+0. ***

Source files:
http://people.apache.org/~fpj/zookeeper-3.4.6-candidate-0/

Maven staging repo:
https://repository.apache.org/content/groups/staging/org/apache/zookeeper/zookeeper/3.4.6/

The tag to be voted upon:
https://svn.apache.org/repos/asf/zookeeper/tags/release-3.4.6-rc0

ZooKeeper's KEYS file containing PGP keys we use to sign the release:
http://www.apache.org/dist/zookeeper/KEYS

Should we release this candidate?
```

Publishing

Once three PMC members have voted for a release, it may be published.

1. Tag the release:

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

   git tag -s release-X.Y.Z -m "ZooKeeper X.Y.Z release."
   Push the newly created release tag to the remote repo.
   git push <remote> release-X.Y.Z
2. Update the version number to the next logical SNAPSHOT version (e.g. X.Y.Z to X.Y.Z+1-SNAPSHOT)
3. Check in release files to the distribution server

   ```
   svn co https://dist.apache.org/repos/dist/release/zookeeper/ zookeeper_dist
   cd zookeeper_dist
   mkdir zookeeper-X.Y.Z
   # copy tgz/asc/md5/sha1 files from your public_html to zookeeper-X.Y.Z
   svn add zookeeper-X.Y.Z
   svn ci -m "Add ZooKeeper X.Y.Z release"
   ```

   a. After checking in the release, you'll receive an automated email from reporter.apache.org with a link requesting additional details about the release. This form must be completed by a PMC member. If you are a PMC members, follow the link and complete the form. If you are not a PMC member, forward the email to dev@zookeeper.apache.org and ask for assistance from a PMC member.
4. The release directory usually contains just two releases, the most recent from two branches, with a link named 'stable' to the most recent recommended version.

   ```
   cd zookeeper_dist
   # don't do the following rm for the time being, until we are sure the new rel is actually stable --> ###
   rm -rf zookeeper-A.B.C
   rm stable; rm current
   ln -s zookeeper-A.B.D stable
   ln -s zookeeper-A.B.D current
   svn ci -m 'Updating links'
   ```

5. Release the Maven artifacts on Apache Nexus:

   a. Click on 'Staging repositories'.
   b. Select the repository corresponding to the release candidate.
   c. Click on the 'Release' button and follow instructions.
6. Wait 24 hours for release to propagate to mirrors.
7. Prepare to edit the website.

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

8. Copy the new release documentation into the content/doc directory

```
cd site/trunk/content/doc
mkdir rX.Y.Z
cd rX.Y.Z
tar xvf /www/www.apache.org/dist/zookeeper/zookeeper-X.Y.Z/zookeeper-X.Y.Z.tar.gz 'zookeeper-X.Y.Z/docs'
mv zookeeper-X.Y.Z/docs/* .
rm -r zookeeper-X.Y.Z

cd ..
svn add rX.Y.Z
```

9. Update the release news in **site/trunk/content/releases.textile**
10. Update the nav panel to include X.Y.Z (which may now be "current") **site/trunk/templates/sidenav.textile**
11. Regenerate the site, review it, then commit it. (checkout https://svn.apache.org/repos/infra/websites/cms/build if you haven't already)

```
cd site/trunk
.../apache_cms_build/build_site.pl --source-base . --target-base /tmp/sitegen
<open /tmp/sitegen/index.html in your browser and verify the site>
svn commit -m "Updated site for release X.Y.Z."
```

12. Update the "current" doc pointer if necessary (as CMS doesn't currently support symlinks in build/staging we need to push this directly to production)

```
svn co https://svn.apache.org/repos/infra/websites/production/zookeeper/content
cd content/doc
rm current
ln -s rX.Y.Z current
svn commit -m "Updated current pointer for release X.Y.Z"
```

13. Review your changes on the staging site: http://zookeeper.staging.apache.org
14. Deploy your site changes.

```
https://cms.apache.org/redirect?uri=http%3A//zookeeper.apache.org/
publish the site changes
```

15. Send announcements to the **user and developer** lists once the site changes are visible. Here is a sample email:

```
Subject: [ANNOUNCE] Apache ZooKeeper X.Y.Z

The Apache ZooKeeper team is proud to announce Apache ZooKeeper version X.Y.Z

ZooKeeper is a high-performance coordination service for distributed
applications. It exposes common services - such as naming,
configuration management, synchronization, and group services - in a
simple interface so you don't have to write them from scratch. You can
use it off-the-shelf to implement consensus, group management, leader
election, and presence protocols. And you can build on it for your
own, specific needs.

For ZooKeeper release details and downloads, visit:
http://zookeeper.apache.org/releases.html

ZooKeeper X.Y.Z Release Notes are at:
http://zookeeper.apache.org/doc/rX.Y.Z/releasenotes.html

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

Regards,

The ZooKeeper Team
```

16. In Jira, ensure that only issues in the "Fixed" state have a "Fix Version" set to release X.Y.Z.
17. In Jira, "release" the version. Visit the "Administer Project" page, then the "Manage versions" page. You need to have the "Admin" role in ZooKeeper's Jira for this step and the next.
18. In Jira, close issues resolved in the release. Disable mail notifications for this bulk change.

Maven

Support is now available for pushing the jar artifacts to the Apache maven repository.

WARNING: the process has changed but the docs not yet updated, see https://issues.apache.org/jira/browse/ZOOKEEPER-1430

1. login to people.apache.org
2. copy the files from the release artifact "dist-maven" directory into /www/people.apache.org/repo/m2-ibiblio-rsync-repository/org/apache/zookeeper/zookeeper/X.Y.Z
3. chmod -R g+w /www/people.apache.org/repo/m2-ibiblio-rsync-repository/org/apache/zookeeper/zookeeper/X.Y.Z
4. cd to /www/people.apache.org/repo/m2-ibiblio-rsync-repository/org/apache/zookeeper/zookeeper
5. edit maven-metadata.xml
    a. add X.Y.Z to versions
    b. update the version to X.Y.Z
    c. update the lastUpdated
6. sign and md5sum maven-metadata.xml

```
md5sum maven-metadata.xml > maven-metadata.xml.md5
gpg2 --armor --output maven-metadata.xml.asc --detach-sig maven-metadata.xml
```

Smoke Tests

Before the release, run the following smoke tests (at least).

- Run unit tests on different hardwares: mac, windows, linux.