Geronimo 2.1.x Release Process

Geronimo 2.1.x Release Process

Reference http://www.apache.org/dev/publishing-maven-artifacts.html

Have 2.1.7 release as an example.

Release Checklist

1. Better to use a non-Windows system to create the release candidate

- Dos line endings makes all unix shell scripts unexecutable
  - https://issues.apache.org/jira/browse/GERONIMO-5440

2. mvn rat:check

- Dealing with the 2.1.7 release, I found the maven rat plugin(1.0-alpha-3, latest) seems have some bugs.
  - incorrect number of unapproved licenses reported by rat:check. This should be resolved in 1.0-alpha-4, but it has not released yet..(http://incubator.apache.org/rat/apache-rat-plugin/changes-report.html)
  - If I designated the profile, for example -Pall-subprojects, rat:check won't run in any of the subprojects.
    If I designated -Pno-it, rat:check will run only in testsuite subproject.
    If I designated -Ptools, which does not contain any subprojects, it will run in all subprojects.
    So, the program logic is inverted.. I have to use -Ptools to do rat:check..

3. Manually update some files:

- delete all pom.xml files, and then search "SNAPSHOT"
  - Manually check plugin-list url in $SRC\framework\configs\plugin\pom.xml
- update ##VERSION## in README.txt and RELEASE_NOTES.txt
- list JIRAs in RELEASE_NOTES.txt (bugs, improvement, new features, limitations)
- Check the copyright year number in NOTICE file
- commit them

4. mvn release:prepare -DdryRun=true -Pall-subprojects

- Diff the original file pom.xml with the one called pom.xml.tag to see if the license or any other info has been removed. This has been known to happen if the starting <project> tag is not on a single line.
- The only things that should be different between these files are the <version> and <scm> elements. Any other changes you must backport yourself to the original pom.xml file and commit before proceeding with the release.
- compare the numbers of pom.xml, pom.xml.tag, pom.xml.releaseBackup
  - NOTE: the following 5 pom.xml files won't generate the pom.xml.tag and pom.xml.releaseBackup files
    \buildsupport\geronimo-assembly-archetype\src\main\resources\archetype-resources\pom.xml
    \buildsupport\geronimo-plugin-archetype\src\main\resources\archetype-resources\pom.xml
    \buildsupport\testsuite-archetype-plugin\src\main\resources\archetype-resources\pom.xml
    \buildsupport\testsuite-archetype-plugin\src\main\resources\archetype-resources\testset-1\pom.xml
    \testsuite\enterprise-testsuite\ejbcontainer-tests\pom.xml

5. Release Prepare

- Before doing release prepare, clean up you local repository to avoid the bad staging release artifacts to be included in the geronimo release. see reference.
- This will update the versions in branch 2.1 and create the release tag
  - mvn release:clean -Pall-subprojects
  - mvn release:prepare -Pall-subprojects
    - you need "mvn clean install -Dstage=bootstrap" in midway

6. Release Perform

- This will stage the release artifacts,
  - mvn release:perform -Pall-subprojects
- In Apache nexus repository, click "close" https://repository.apache.org/index.html#welcome

7. Vote

- Vote in mailing list, meanwhile wait TCK results. The artifacts up for vote are the geronimo-2.1.x-source-release.tar.gz and geronimo-2.1.x-source-release.zip.
- Post "RESULTxxx" when vote close.

8. Release artifacts

- In Apache nexus, click "release"
    - the artifacts will be synchronized to maven central repository in some time.

9. Update geronimo-plugins.xml

- delete your local ~/.m2/repository/geronimo-plugins.xml
- build tag 2.1.7, which will generate a new geronimo-plugins.xml in ~/.m2/repository/
- do the actions as described here – https://cwiki.apache.org/GMOxPMGT/geronimo-server-release-process.html – in step 12.

10. Upload artifacts to dist

- Upload artifacts to http://www.apache.org/dist/geronimo/
    - Copy https://svn.apache.org/repos/asf/geronimo/KEYS to people.apache.org:/www/www.apache.org/dist/geronimo/KEYS if your public key is not in it (one time)
    - Upload the artifacts and their checksums to people.apache.org:/www/www.apache.org/dist/geronimo/2.1.5
    - Ensure distribution directories have 0775 dir permission and a 0664 file permission set on them
        - Try fetch_checksum and upload_dist scripts to free you out of tedious manual work 😎
    - ps: the changes in people.apache.org:/www/www.apache.org/dist/geronimo need take some time to get sync in http://www.apache.org/dist/geronimo/

11. Announce in Mailing list and Post news in homepage

- Modify http://geronimo.apache.org/downloads.html.
- Add a new page list the artifacts that can be downloaded.
- Modify frontpage and add a news.

12. Update the security advisory page

- Add a section to the new release at https://cwiki.apache.org/confluence/display/GMOxSITE/2.1.x+Security+Report
- If there are outstanding advisories for vulnerabilities fixed by this release, move the vulnerability descriptions to the new release section.

13. Manaually update files in the 2.1 branch after release

- update 2.1.7 to ##VERSION## in README.txt and RELEASE_NOTES.txt
- remove the JIRA list in RELEASE_NOTES.txt (bugs, improvement, new features, limitations)
- search "2.1.7" and change them to "2.1.8-SNAPSHOT"
- Update artifact-alias, add version 2.1.7 in artifact-alias after 2.1.7 release
    - /framework/configs/pom.xml
    - /plugins/client/pom.xml
    - /plugins/corba/client-corba-yoko/pom.xml
    - /plugins/pom.xml
- commit them

Prerequisite

1. Use Genesis 2.0 as a parent pom

```
<parent>
  <groupId>org.apache.geronimo.genesis</groupId>
  <artifactId>genesis-java5-flava</artifactId>
  <version>2.0</version>
</parent>
```

- genesis-java5-flava-2.0.pom
    - genesis-default-flava-2.0.pom
        - genesis-2.0.pom
            - apache-6.pom

2. Use Maven 2.2.1

- Enable Apache Servers (refer: http://maven.apache.org/developers/committer-settings.html)

```
<settings>
...
  <servers>
    <!-- To publish a snapshot of some part of Maven -->
    <server>
      <id>apache.snapshots.https</id>
      <username> <!-- YOUR APACHE LDAP USERNAME --> </username>
      <password> <!-- YOUR APACHE LDAP PASSWORD --> </password>
    </server>
    <!-- To publish a website of some part of Maven -->
    <server>
      <id>apache.website</id>
      <username> <!-- YOUR APACHE LDAP USERNAME --> </username>
      <filePermissions>664</filePermissions>
      <directoryPermissions>775</directoryPermissions>
    </server>
    <!-- To stage a release of some part of Maven -->
    <server>
      <id>apache.releases.https</id>
      <username> <!-- YOUR APACHE LDAP USERNAME --> </username>
      <password> <!-- YOUR APACHE LDAP PASSWORD --> </password>
    </server>
    <!-- To stage a website of some part of Maven -->
    <server>
      <id>stagingSite</id> <!-- must match hard-coded repository identifier in site:stage-deploy -->
      <username> <!-- YOUR APACHE LDAP USERNAME --> </username>
      <filePermissions>664</filePermissions>
      <directoryPermissions>775</directoryPermissions>
    </server>
    ...
  </servers>
</settings>
```

reference:
It is highly recommended to use Maven's password encryption capabilities for your passwords.http://maven.apache.org/guides/mini/guide-encryption.html

3. Setup PGP Keys (for the ones who be the release manager the first time)

- Download gnupg2
- Generate your PGP Key (refer: http://www.apache.org/dev/openpgp.html) so that maven-release-plugin can sign your built artifacts when do release:perform
    - How To Avoid SHA-1
    - How To Generate a Strong Key
- Update Maven's settings.xml with following:

```
<settings>
  ...
  <profiles>
    <profile>
      <id>apache-release</id>
      <properties>
        <gpg.passphrase> <!-- YOUR KEY PASSPHRASE --> </gpg.passphrase>
      </properties>
    </profile>
  </profiles>
  ...
</settings>
```

- Meanwhile, append your public key to https://svn.apache.org/repos/asf/geronimo/KEYS and people.apache.org:/www/www.apache.org/dist/geronimo/KEYS so that user can verify the artifacts you released.
    - gpg --gen-key
        - RSA and RSA (default), 4096
    - gpg --list-sigs "xxxxxx" && gpg --armor --export "xxxxxx" > xxxxxx.key
        - "cat" your public key to above KEYS file

reference:
http://maven.apache.org/developers/release/pmc-gpg-keys.html