

Release Procedure

Jetty6 Release Process

Currently the release procedure for Jetty6 is a mostly manual operation:

1. Verify that there are no blocking issues on the lists or in JIRA.
2. Update VERSION.txt with the release number and date and commit to trunk. Copy this to trunk VERSION.txt
3. Create tags in jetty and jetty-contrib repositories with a copy from trunk (or the branch used as a basis for the release):

```
svn cp -m 'release x.y.z'
https://svn.codehaus.org/jetty/jetty/branches/jetty-6.1
https://svn.codehaus.org/jetty/jetty/tags/jetty-x.y.z
svn cp -m 'release x.y.z'
https://svn.codehaus.org/jetty-contrib/jetty/branches/jetty-6.1
https://svn.codehaus.org/jetty-contrib/jetty/tags/jetty-contrib-x.y.z
```

4. Checkout tag:

```
svn co https://svn.codehaus.org/jetty/jetty/tags/jetty-x.y.z
```

5. Edit svn:externals to include contrib tag:

```
cd jetty-x.y.z
svn propedit svn:externals . # change branches to
jetty/tags/jetty-contrib-x.y.z
rm -fr contrib
svn up
```

6. Update the poms and other files to change 6.1-SNAPSHOT to x.y.z:

```
find -name '*.xml' -o -name '*.TXT' -o -name '*.txt' | while read file ;
do sed --in-place -e 's/6.1 SNAPSHOT/x.y.z/g' $file ; done
```

The diff in the script will sometimes need to be updated. Recursive greps should be done to make sure that all 6.1-SNAPSHOT references have been updated.

7. Commit the changes to the tag:

```
svn commit -m 'release x.y.z'
svn commit -m 'release x.y.z' contrib
```

8. build the release with javadoc + jxr for bundle

```
mvn -Pcodehaus-release deploy

# find all instances of plugin-plugin in pom.xml and comment them out.
cd distribution/jetty-assembly
mvn -Pcodehaus-release deploy
cd ../../
# revert the plugin-plugin pom.xml files. Complain on lists that we should
fix this.
```

#build the release bundles:

```
bin/build_release_bundles.sh .
```

This builds a binary and source zip bundle in the parent directory.

#build additional bundles and jars:

NOTE: you will need the rpmbuild utility installed in order to build the rpm module. this is also a somewhat optional step and can be completed after the release if need be

NOTE: you will need special utilities installed in order to build the debian module (see the README in contrib/debian for details).

```
cd contrib/j2se6 (NOTE: you need to do this
with jdk1.6!)
```

```
mvn -Pcodehaus-release deploy
```

```
cd ../../contrib/maven-beanshell-plugin
```

```
mvn -Pcodehaus-release deploy
```

```
cd ../../contrib/jetty-ant
```

```
mvn -Pcodehaus-release deploy
```

```
cd contrib/rpm
```

```
mvn install
```

```
cd ../../extra/jboss
```

```
mvn -Djboss.home=<your jboss install dir>
```

```
install
```

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

1. Tell somebody else about the tag and ask them to check it out and test it!
2. Test the tag yourself in situ (if you need to fix anything you will need to completely clean and go back to first build step).
3. Test the built bundle
4. Test the RPM
5. Wait a day for other feedback from other testers
6. Create directories in distribution directories and copy artifacts to codehaus (webdav to <https://dav.codehaus.org/dist/jetty/>) and mortbay (scp to mortbay@mortbay.com:/home/ftp/pub/jetty-x.y.z/). The artifacts to upload are:
 - jetty-x.y.z.zip
 - jetty-x.y.z-src.zip
 - jetty-ant-x.y.z.jar
 - jetty6-*-x.y.z.noarch.rpm
 - jetty-x.y.z-jboss-p.q.y.GA-jsp-2.1.sar
 - jetty-j2se6-x.y.z.jar
 - **jetty6-all.deb**
7. ONLY WHEN YOU ARE REALLY REALLY REALLY REALLY REALLY SURE EVERYTHING IS REALLY REALLY REALLY OK, then push the maven artifacts:

promote the releases

8. Update jira to release the version, and enter the next version number
9. Tell everybody: blogs, lists, etc.
10. Wait for the bug reports to come in