

# ReleaseManagement

## Release Management

```
* get your gpg key into a file called KEYS at

https://svn.apache.org/repos/asf/excalibur/trunk/KEYS

* copy this file to

/wwwwww.apache.org/dist/excalibur

ie

wget https://svn.apache.org/repos/asf/excalibur/trunk/KEYS
scp KEYS minotaur.apache.org:/www/www.apache.org/dist/excalibur

* update project.xml. Probably need to change the version numbers.
Commit the changes.

cd ~/svn/excalibur/trunk
vim instrument/project.xml
vim instrument-manager/project.xml
vim instrument-client/project.xml
svn commit -m 'prepare for releases'

or something like that.

* optional: temporary get rid of your gpg passphrase

gpg --edit-key <YOUR ID>
> key <YOURKEY>
> passwd
<YOUR PASSPHRASE>
<enter>
<enter>
yes
quit
yes

or something like that.

* make sure you're using the latest version of the Sun JDK 1.3 to build the distributions.
Otherwise subtle incompatibility issues may arise later. Exceptions are packages which
utilize JDK1.4 logging like excalibur-framework-impl. Those need to be built using JDK1.4.

* build distributions, tag svn, and deploy them using

maven multiproject:install
cd instrument
maven excalibur:release
cd ../instrument-manager
maven excalibur:release
cd ../instrument-client
maven excalibur:release

The way the script works is if you have the string "RC" or the string "dev"
in the version then it assumes it is a release candidate
and deploys to your own personal account
(/home/${user}/public_html/dist) otherwise it deploys to
main area (/www/www.apache.org/dist).

To deploy the artifacts you add something like the following
into your ~/build.properties or you can specify it on command
line.

maven.username=donaldp
maven.privatekey=C:\Documents and Settings\Peter\Application Data\SSH\UserKeys\PeterHome
```

The goal to deploy artifacts is "excalibur:deploy", tagging repo is "excalibur:tagRelease" and releasing (which tags and deploys) is "excalibur:release". See maven.xml file(s).

With the newer maven releases JSCH is used in the deploy plugin, and ssh agent forwarding and the like won't work. To get around this, you will probably need to specify

```
maven.repo.default.passphrase=XXXX
```

either on the commandline or in ~/build.properties.

- \* optional: restore your gpg passphrase

```
gpg --edit-key <YOUR ID>
> key <YOURKEY>
> passwd
<YOUR PASSPHRASE>
<YOUR PASSPHRASE>
quit
yes
```

or something like that.

- \* ask for feedback on dev@excalibur

- \* probably go through several iterations of the above as some last-minute bugs are hammered out

- \* call for a PMC vote to release the files

- \* if the vote passes, change the version numbers to final and rerun the build process. With no "RC" in the version name the files should end up at

<http://www.apache.org/dist/excalibur/>

- \* send out announcements to the appropriate mailing lists

see also: <http://wiki.apache.org/avalon/AvalonReleaseManagerHowto>. It has sample announcements and the like. Don't pay attention to the rest of the process though :-D