

# ReleaseGuidelines

## Shale Release Guidelines

### Document Status

This document is currently being updated to note the steps required to perform a Shale release based on our new Maven 2 based build infrastructure. After refinement, it will likely be voted on as a *normative* process document. Until then, it should be considered to be *informative* only.

### Steady State Conditions

Between releases, it is assumed that the following conditions hold:

- Version number in all of the POMs is **x.y.z-SNAPSHOT** (for instance, 1.0.3-SNAPSHOT)
- JIRA contains a version labelled **x.y.z-SNAPSHOT** used to tag issues to be fixed in this release
- Developers complete an issue by setting status to **Resolved** with an appropriate *Resolution*

FIXME - maybe need note about developers testing against both [MyFaces](#) and the JSF RI? And running integration tests as well as unit tests? Or maybe this should all go on a page about "stuff to do before you check code in."

Occasionally, it may be necessary to update the Shale Master POM (maven/master-pom/pom.xml). If this becomes necessary, it must be released separately, before a Shale release that depends upon it can be performed. See [ShaleReleaseMasterPom](#) for more information.

### Begin Release Planning

Appoint a release manager for the upcoming release. He or she will manage the remainder of the process described in this document.

Release manager should set a target date for the release, and set up a Release Plan page on the wiki, using [ReleasePlanTemplate](#) (FIXME - this document is not ready to use yet) as a page template. The new release plan page should be linked from [ReleasePlans](#) also.

Review outstanding unresolved JIRA issues, to determine whether to target setting the *Fix Version* for each issue to **x.y.z-SNAPSHOT** or not. This will likely involve discussions on the dev list to come to consensus on what is in and what is not.

### Work Off The Road Map

From this point forward the JIRA "Road Map" for release **x.y.z-SNAPSHOT** will be an invaluable resource in seeing what is left to be accomplished.

Encourage developers to focus on the road map issues. After the initial review, no new issues should be set with a fix version of **x.y.z-SNAPSHOT** without the explicit agreement of the release manager, and the consent (implied by silence or explicit) of the other developers.

### Final Snapshot Review

After all road map issues have been resolved, release manager should coordinate an in depth review of the latest snapshot, in preparation for switching to the real **x.y.z** version number and publishing the release. This involves preparing both Maven artifacts to be published into a repository, and release artifacts to be published on the Apache distributions server and mirror network.

To prepare and validate the Maven artifacts:

- Turn off any automated deployment or artifact publishing from continuous build or nightly build scripts
- Check out and locally build the entire trunk (see FIXME for details)
- Deploy all artifacts that will be deployed in the official release (see FIXME for details) into the snapshot repository
- Digitally sign each of the release artifacts (both POMs and JARs) (see FIXME for details) and manually upload the ".asc" signature files to be sitting next to the artifacts they have signed
- Ask on the dev list for people to examine the individual artifacts and report back:
  - Appropriate contents of each JAR file
  - Correct META-INF/MANIFEST.MF values
  - Correct POM settings, such as dependencies and versions
  - Validity of checksums and digital signatures

To prepare and validate the release artifacts:

- Check out and locally build the entire trunk (see FIXME for details)
- Build the release artifacts using appropriate assembly steps (see FIXME for details)
- Create .md5 and .sha1 checksums for each release artifact
- Digitally sign each release artifact (see FIXME for details)
- Upload the snapshot release artifacts, checksums, and digital signatures to a convenient place (such as a subdirectory of the release manager's public\_html directory)
- Ask on the dev list for people to examine the individual release artifacts and report back:
  - Appropriate contents of each artifact
  - Can each artifact be rebuilt from itself with Maven?
  - For sample apps, can the apps be successfully dropped into a servlet container and run?

- Validity of checksums and digital signatures

Iterate over these review processes until the release manager is satisfied that it is time to actually cut a test release and post it.

## Cut An Actual Release

FIXME - create release notes here?

Call for a moratorium on commits to the trunk (or whatever branch the release is being created from).

Update the artifact version number from **x.y.z-SNAPSHOT** to **x.y.z** in the *shale-parent* POM, as well as any other top level POMs that do not inherit from *shale-parent*.

Update the parent version number from **x.y.z-SNAPSHOT** to **x.y.z** in all of the dependent POMs.

Build and deploy the Maven artifacts, as described above. Because of the version number changes, these artifacts will be posted to different directories on the repository than the snapshots. Digitally sign these artifacts, and upload the signatures as well. (NOTE - due to the way Maven treats non-SNAPSHOT version numbers, we **really** only want to do this once.)

Build the release artifact assemblies, create checksums and signatures, and upload to a public directory as described above. This time, the filenames should all have **x.y.z** versions in their filenames, rather than **x.y.z-SNAPSHOT**.

Call for a release vote. This vote is **only** to determine whether the release should take place. A vote on the quality ranking of the release will occur later. Therefore, something like the following text should be used in the vote message on the dev list:

```
FIXME - example vote text
```

## If The Release Vote Fails

FIXME - decide what to do here, but it might mean throwing away a version number.

## If The Release Vote Passes

FIXME - Tag the release here?

FIXME - Need to tie into branching strategy, which is TBD at this point.

Copy all of the Maven artifacts for this release from the snapshot repository (/www/people.apache.org/repo/m2-snapshot-repository/org/apache/shale) to the ibiblio sync repository (/www/people.apache.org/repo/m2-ibiblio-repository/org/apache/shale). See FIXME for more information.

Write a message like this to the Maven developer list, asking that the Apache ibiblio repository get rsync'd over to the public ibiblio server.

Copy the release artifacts, checksums, and digital signatures to the appropriate distribution directory (/www/people.apache.org/dist/shale/release/x.y.z). See FIXME for more information.

Set up a download page on the website that allows users to download from an appropriate mirror. See FIXME for more information.

Wait until the Maven repository has been synchronized, and until the mirrors have been updated. (Can take a couple of days.)

Announce the release to the Shale dev and user lists, and to the Jakarta announcements list (FIXME - anywhere else?) with a message like this:

```
FIXME - example release announcement
```

## Post Release Cleanup

FIXME - fix the JIRA version issues (x.y.z-SNAPSHOT to x.y.z)

FIXME - mark the x.y.z version as "released" in JIRA from the admin screens

FIXME - reset the POM version numbers in the trunk (or whatever branch the release was cut from)

FIXME - at some future time hold a quality ranking vote.

FIXME - anything else? Should be back to steady state at this point.

## References

1. [Signing Releases](#)
2. Apache [Mirroring Guidelines](#)