

Spec Versioning

Single Version

A single version number is used for all specifications. When a release is made all specification jars are released together with the single version number.

Pros

- Release process is simple and can be fully automated with the release plugin.
- User only has to know one version number to get the most recent release.

Cons

- New versions of the specification jars are created with each release even if the code has not changed.
- User will end up with duplicate jars in local repository which is confusing and a waste of resources on the repository server and local system.
- New specs appear with high version numbers.
- An uncertified spec can hold up the release of tested and specifications.

Version Per Spec

Each specification has a separate version number and is released independently of other specification jars.

Pros

- Each revision number for a specification contains code changes.
- Users do not have to download new jar releases without code changes.
- Specifications can be developed and released without having to release all other specs some of which may not be ready to release.

Cons

- Releases are more difficult because the person performing the release must be aware of any dependencies and must also rerelease those jars. (eliminated with working version-ranges)
- Duplicate jars are created when no code has changed which is a waste of resources on the repository server and local system. (eliminated with working version-ranges)
- Users must be aware of the newest version number for the specification jars they are using. (eliminated with working version-ranges)

Supporting Facts

- There are currently 26 specifications.
- Version ranges don't work several (most?) important maven plugins
- 11 of the specs have inner dependencies:
 - ejb 2.1 -> jta 1.0.1b
 - ejb3 -> jta 1.1, interceptor, annotation
 - connector 1.5 -> jta 1.0.1b
 - jacc -> servlet 2.4
 - j2ee mgmt -> ejb 2.1
 - javamail 1.3.1 -> activation
 - javamail 1.4 -> activation
 - jaxr -> activation
 - jaxrpc -> saaj, qname, servlet 2.4
 - jsp -> servlet 2.4
 - saaj -> activation
- Only 5 of the 11 specs will change often due to inner dependencies:
 - javamail 1.3.1 -> activation
 - javamail 1.4 -> activation
 - jaxr -> activation
 - jaxrpc -> saaj, qname, servlet 2.4
 - saaj -> activation
- Several specifications have not had code changes in 2+ years:
 - ejb
 - servlet
 - jms
 - transaction
 - connector
 - qname