

# Release Management

The issue we are aiming to resolve here is primarily how to release multi-module products.

This includes:

- management of parent version references
- inheritance of the version from a parent
- implications for source control

## Management of Parent Version References

The current thinking on this is that in general the parent version will not be specified in the POM and that it will be located using the current search algorithm:

- found inside reactor
- found using USD
- found using the RELEASE version mechanism (as for plugins)

This prevents the need to make changes consistently to that when the parent increments its version. At the point of release, the version will be populated with what was found at the time to make the build reproducible.

## Inheritance of the Version from a Parent

While this was originally planned and executed on continuum and m2, among others, it proves problematic for making a release, and prevents making interim releases of individual components for patches or use in other applications (unless snapshots are used which is not particularly desirable as they haven't been properly promoted).

The planned approach is to:

- version each component
- set the product version on the project producing the actual assembly
- release individually (conceptually, the tool can actually do them at the same time as a series of individual releases)
- assist in syncing name using the release tool if that is the desired approach of the product (eg some products like geronimo want to keep the product version in all the JARs to be able to recognise where they belong)

## Implications for Source Control

While making each module independent it has implications for tagging releases, and how checking out from source control works.

Desired outcomes:

- checking out an individual module on its own will build correctly
- being able to check out the entire source tree for a particular release and build in one
- tagging an entire release correctly
- tagging an individual module is possible
- branching an individual module is possible

## Checking out an individual module

This will build on its own since the parent is referenced using a repository location. However, if the version is omitted and RELEASE is assumed, then the behaviour between the search paths above is different (ie, in the reactor or USD you get the latest from your checkout, but when checked out on its own, you get what is in the repository as a release).

I believe this is not a problem, and that the behaviour can be expected. It should encourage proper releasing of the parent when it is changed.

## Being able to check out the entire source tree, tags and branches

What this involves is for all projects to have a common root. Obviously if they are being built together with the modules definition then this will be the case. This also means that they need to be tagged together, and uniformly.

The assembly should also be made at the root and incorporate the modules at that point of time, otherwise the dependency versions could well be out of sync with what is tagged in the SCM.

This becomes a problem in subversion if the individual components have a trunk/branches/tags structure due to being tagged or branched separately. It is expected that the lowest point you would check out from will be the trunk/branches/tags, and those below cannot be tagged or

branched.

- Where should an individual create a branch or tag?
- How is it different for other SCMs?
- Does maintaining different branches (either via svn switch/cvs style or different paths in the checkout) affect the development cycle in a multimodule scenario?
- If it is decided that a whole set of modules will only ever be tagged and branched together, is there any merit to versioning them together again then?