

Artifact API Improvements

Improving Maven's Artifact API

Context

After almost a year of large-scale deployment in the community, Maven is poised to improve some of its APIs to improve developer productivity, particularly in the realm of plugin development.

Specifically, Maven's artifact-resolution subsystem could be easier to use. The following sections will discuss some specific improvements that would make plugin developers vastly more productive.

Colocate POM information with Artifact instances

Developers often could provide more intuitive plugin implementations if access to the POM information for a given Artifact instance was readily available. Therefore, we should look at providing an accessor method on the Artifact class which will retrieve the associated MavenProject instance for that artifact instance. This should be a relatively easy fix, since we always resolve POMs first, then the artifact itself...which means this information is always available, in some form. An alternative approach might be to provide a utility method to retrieve the MavenProject instance, given some Artifact instance parameter.

Make the Artifact a node in the dependency graph structure

In many plugins and reports, it would be very useful to have access to the subtree of artifacts which were transitively resolved by that artifact instance. Additionally, it might be useful to know which artifact instances caused the current artifact to be included.

Once again, this information is already constructed in some form during artifact resolution...we just need to keep track of it, and provide accessor methods on the Artifact class (or some utility).

Problem

These are simply productivity enhancements for artifact-handling plugins, and should pose no dramatic hurdles to maintaining backward compatibility. The main issue right now is that Maven discards some very useful information during the artifact resolution process; we need to think about capturing it for future use. This will have the side effects of making plugin implementations cleaner, and reducing the number of components involved in many plugins' executions.

Resources