

"Compile only" dependencies

Implementation

Provided Scope

Add a `provided` scope that makes a dependency essentially `compile` scope, but it is **not bundled, not on the execution classpath** (eg for surefire) and the dependency is **not transitive**. It is intended for use in the final "end POM" only.

This addresses "compile only" dependencies, while not losing the benefits of having the scopes tied to each other.

Integration Test Scope

Add a `integration-test` scope for the integration test phase plugins, like cactus.

Original Discussion

This is the case of servlet API/J2EE, and perhaps some libraries provided by the final JDK but not compiling JDK. It's not that they are compile-only - it's that they **may** be provided for you at runtime. They are still required at runtime.

All we really need to change is to ensure:

- they are not bundled when they target an environment where it will be provided (always the case for servlet/J2ee/etc type JARs).
- they are not used from plugins where the plugin provides them (eg, mock servlet api tests)

Current options seem to be:

- a "container" scope that implies compile, runtime and test, but can be excluded easily from bundling
- exclusions on a per plugin basis
 - a full implementation of specification dependencies would allow the plugins to exclude based on the specifications by default
 - this is a bit limited as you would always have to add weblogic specific ones, for example
- change to independant scopes which is more or less what you are proposing
 - I've listed a couple of downsides to this previously