

Plugin and Report Inheritance

There is definitely a desire to allow inheritance of some of the "active" portions of the POM: plugin definitions, report definitions and default goal. This has to be weighed up against the possibility of this doing something unexpected in a multi-module scenario - eg the inclusion of a "dashboard" report at the top level that you don't want inherited.

Here is the plan as I see it, feedback welcome:

1. default goal is inherited. This shouldn't be harmful, and it can just be redefined in each subproject or an intermediate ancestor (think maven-plugins) if necessary.
2. each mojo can declare whether it should be inherited by default. The default is that it **is** inherited, but "@inherited false" would make the default not to inherit. This is useful for mojos where inheritance rarely makes sense (eg, assembly:assembly, or a dashboard report).
3. the inheritance of a mojo can be overridden by `<inherit>true|false</inherit>` in the parent POM, under either goal or plugin
4. note that because the inheritance default is specified at the mojo level, plugins are always inherited so the lifecycle bindings will take effect - so its important for mojos binding to the lifecycle that shouldn't be inherited declare this.
5. plugin and report lists are merged when inheritance, not overridden. This means the inheritance responsibility is in the parent. There is no way to block it out from a single pom.

I think this works in all the scenarios brought forward so far, gives sensible defaults and reduces the amount of duplication. It also brings the behaviour inline with dependencies.

It doesn't affect the behaviour of pluginManagement, which is always inherited as a merge, but only applied when a plugin is made active by being in the plugin section, or introduced through the lifecycle.