Maven2 project and JDK5

Hudson makes extensive use of generics. Thus it has to be compiled with JDK 1.5. There is no possibility of removing the generics from the code as it is too well integrated.

Up until builds in the 1.90ish area, the Hudson build process would use retrotranslator to backport the Hudson to work on JDK 1.4. However, retrotranslator is not perfect. It does not catch every case where you have used a method that was introduced in JDK 1.5. Very often, it would take a couple of days before somebody would catch the problem.

Basically, using retrotranslator was a patch over the seeping wound at best. So retrotranslator was dropped, and you must run Hudson with JRE 1.5.

To make things more fun, there is a bug in JDK 1.5 to do with generic exceptions, so Hudson needs to be compiled with JDK 1.6. Thankfully, this is just a compilation requirements and not a runtime requirement, so we can run Hudson with JDK 1.5)

The Maven2 project type (alpha) uses Maven in a semi-embedded mode, therefore, it must run Maven in the same JRE (as far as I know).

Building with JDK5 and deploying to JDK1.4 (or earlier) is quite possible, but if you'd like to use compiler from the earlier version of JDK, or to run unit tests with the target JVM, you can do as follows:

In your ~/.m2/settings.xml put a ,

```xml
<properties>
  <jdk.ibm.1.4.2.path>c:\jdks\ibm-1.4.2</jdk.ibm.1.4.2.path> <!-- or whatever your local path is -->
  <!-- if windows -->
  <jdk.ibm.1.4.2.jvm>${jdk.ibm.1.4.2.path}bin\java.exe</jdk.ibm.1.4.2.jvm>
  <jdk.ibm.1.4.2.javac>${jdk.ibm.1.4.2.path}bin\javac.exe</jdk.ibm.1.4.2.javac><!-- if windows -->
  <!-- if *nix -->
  <jdk.ibm.1.4.2.jvm>${jdk.ibm.1.4.2.path}bin\java</jdk.ibm.1.4.2.jvm><
  <jdk.ibm.1.4.2.javac>${jdk.ibm.1.4.2.path}bin\javac</jdk.ibm.1.4.2.javac><!-- if *nix -->
</properties>
```

Then change your project's pom.xml such that you have something like

```xml
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-compiler-plugin</artifactId>
  <configuration>
    <verbose>true</verbose>
    <fork>true</fork>
    <executable>${jdk.ibm.1.4.2.javac}</executable>
    <compilerVersion>1.4</compilerVersion>
  </configuration>
</plugin>

<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <configuration>
    <jvm>${jdk.ibm.1.4.2.jvm}</jvm>
    <forkMode>once</forkMode>   <!-- can also use always. Cannot use never -->
  </configuration>
</plugin>
```

You might also consider adding the enforcer plugin to stop the build if the properties are not set or if the compiler and jvm don't exist.

Note: I have not tested this, the need for the .exe's on windows may not exist. Give it a try and update this page when you have something that works (if my initial guess is incorrect)