

Maven and TestNG

By [Roger Keays](#), 8 November 2006

I'm fairly new to maven, but so far it has really impressed me with its simplicity, productivity gains and IDE independence. So I was really looking forward to porting some unit tests to TestNG, running mvn test and watching in awe as maven found my tests, ran them and generated a pretty html report of the test results. Unfortunately, it didn't turn out quite like that.

It would seem there are some unfortunate bugs in surefire (maven's test runner) and these extended my five minute job into six hours of bewilderment - with occasional profanities. Anyway, to cut a long story short, here is a list of problems I encountered using TestNG with maven, and the ultimate solution I settled for.

The Problems

- Each of the Test* classes were found by surefire, but none of the tests in them were run. It turns out that this is a known bug [1], and the workaround is use an xml test suite instead of relying on class name matching.
- A TestNG exception caused by a missing parameter for an @BeforeSuite method was being hidden by surefire, and the build failing mysteriously. I only discovered the problem when I changed @BeforeSuite to @Test.
- The surefire-report reruns the test cases. Again, this is a known bug [2] and isn't critical, but it is quite annoying. To only execute the tests once, you can run mvn surefire-report:report instead of mvn test.
- The test report repeats the results from all the previous tests in each section, so you end up with longer and longer lists of repeated data (another known bug [3]).
- The straw which broke the camels back: Nested exceptions are not shown in the test output. Many of my test failures were caused by "An exception has occurred, please check the nested exception for details".

A Solution

Fortunately, maven can run ant scripts and the testng jars include some ant tasks for executing testng 'natively'. We can configure surefire not to run the test cases and use the <testng> ant task instead. Also, the <junitreport> task can be used to generate a similar html report to that generated by the surefire-report plugin.

```
<build>
  <plugins>
```

```

<plugin>
  <artifactId>maven-antrun-plugin</artifactId>
  <executions>

    <!-- this ant script runs testng natively -->
    <execution>
      <id>testng</id>
      <phase>test</phase>
      <configuration>
        <tasks>
          <taskdef resource="testngtasks" classpath="testng.jar"
            classpathref="maven.test.classpath"/>
          <testng classpathref="maven.test.classpath"
            outputdir="target/test-reports">
            <xmlfileset dir="src/test/suites" includes="*.xml"/>
          </testng>
          <junitreport todir="target/test-reports">
            <fileset dir="target/test-reports">
              <include name="**/*.xml" />
            </fileset>
            <report format="noframes" todir="target/test-reports" />
          </junitreport>
        </tasks>
      </configuration>
      <goals><goal>run</goal></goals>
    </execution>
  </executions>
  <dependencies>
    <dependency>
      <groupId>ant</groupId>
      <artifactId>ant-junit</artifactId>
      <version>1.6.2</version>
    </dependency>
  </dependencies>
</plugin>

<!-- disable surefire plugin (too many problems!) -->
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-surefire-plugin</artifactId>
  <configuration>

```

```
        <skip>true</skip>
    </configuration>
</plugin>
</plugins>
</build>
```

Of course you'll need to make your project dependent on TestNG also:

```
<dependency>
    <groupId>org.testng</groupId>
    <artifactId>testng</artifactId>
    <version>5.1</version>
    <scope>test</scope>
    <classifier>jdk15</classifier>
</dependency>
```

References

- [1] <http://jira.codehaus.org/browse/MSUREFIRE-134>
- [2] <http://jira.codehaus.org/browse/MSUREFIREREP-6>
- [3] <http://jira.codehaus.org/browse/MSUREFIRE-172>

About Roger Keays



Roger Keays is an artist, an engineer, and a student of life. He has no fixed address and has left footprints on 40-something different countries around the world. Roger is addicted to surfing. His other interests are music, psychology, languages, the proper use of semicolons, and finding good food.