SoftwareEntwicklung ▸ Beratung ▸ Schulung

Unit- and Integration Testing with Maven

Web Site:
www.soebes.com
EMail:
apachecon@soebes.com

Dipl.Ing.(FH) Karl-Heinz Marbaise

Agenda

1. Overview Maven
2. Maven Lifecycle
3. Unit Tests
4. Unit Tests Multi Module
5. Integration Tests
6. Maven Plugin Development

1. Overview - Maven

- Official Web Site
  - http://maven.apache.org

Apache Maven is a software project management and comprehension tool. Based on the concept of a project object model (POM), Maven can manage a project's build, reporting and documentation from a central piece of information.

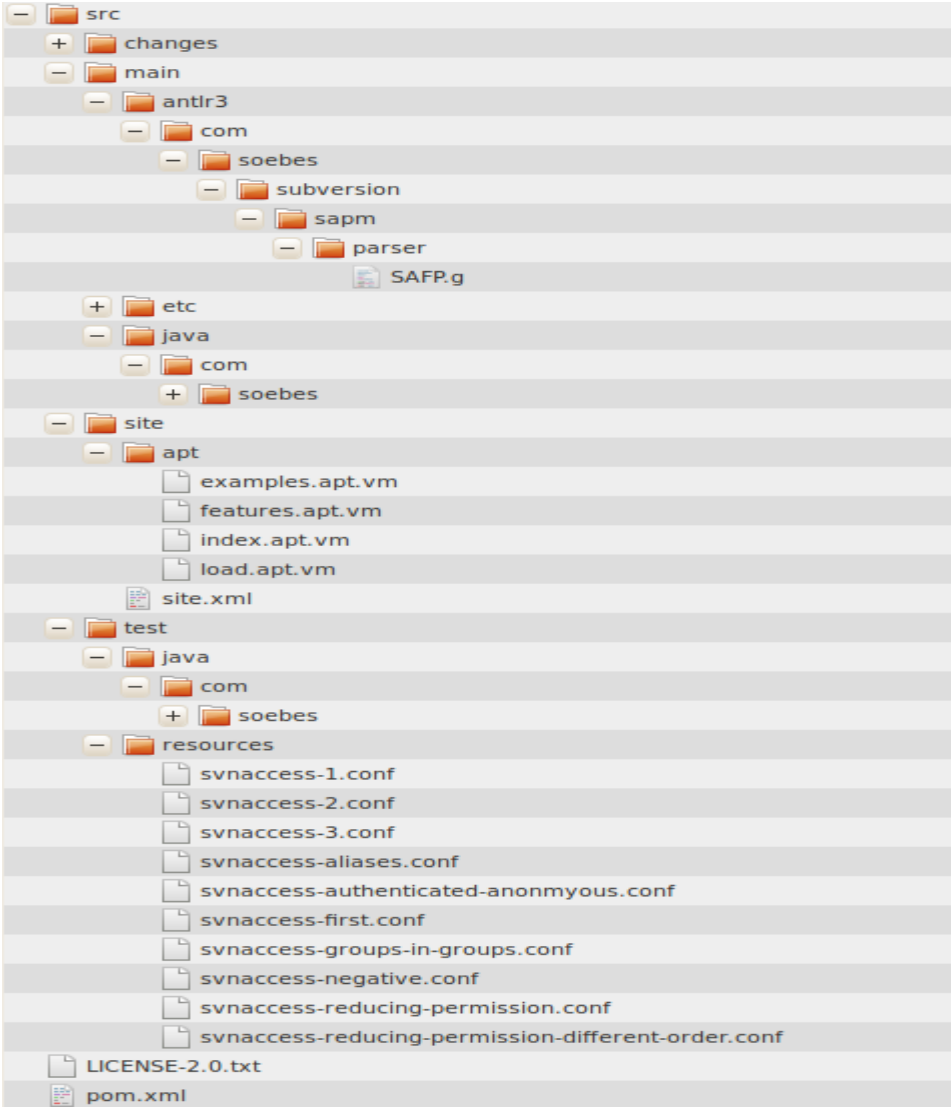
1. Overview - Maven

- Currently Maven 3.0.4 is most recent version
  - Convention over Configuration
  - Large number of plugins
    - jar, war, ear, ejb, rpm, assembly, appassembler etc.
  - Support in many tools like CI (Jenkins, Hudson, TeamCity, Bamboo etc.), IDE (Eclipse, IntelliJ, Netbeans etc.)

1. Overview - Maven

- Separate folder for production code and appropriate resources
- Separate folder for unit test code and appropriate resources

https://github.com/khmarbaise/sapm
http://khmarbaise.github.com/sapm/



1. Overview - Maven

- Maven Coordinates
  - Identify an artifact
    - groupId, artifactId, version (GAV), packaging (default: jar; ejb, war, ear, rpm etc.), classifier (jdk15 etc.)
    - Version
      - Released artifacts:
        - 1.2.0, 3.1,...
      - Not Released artifacts (These artifacts are currently under development):
        - 1.2.0-SNAPSHOT, 3.1-SNAPSHOT,...
  - Example: org.bouncycastle:bcprov-jdk15:jar:1.45

1. Overview - Maven

- Dependency Management
  - Simple as well as transitive dependencies.
    - Your project uses the Tika library.
    - What about the dependencies of the Tika library?
      - This is handled by Maven which are called **transivite dependencies.**

```
+- org.apache.tika:tika-core:jar:1.1:compile
+- org.gagravarr:vorbis-java-tika:jar:0.1:compile
|  \- org.gagravarr:vorbis-java-core:jar:tests:0.1:runtime
+- org.apache.felix:org.apache.felix.scr.annotations:jar:1.
+- edu.ucar:netcdf:jar:4.2-min:compile
|  \- org.slf4j:slf4j-api:jar:1.5.6:compile
+- org.apache.james:apache-mime4j-core:jar:0.7:compile
+- org.apache.james:apache-mime4j-dom:jar:0.7:compile
+- org.apache.commons:commons-compress:jar:1.3:compile
+- commons-codec:commons-codec:jar:1.5:compile
+- org.apache.pdfbox:pdfbox:jar:1.6.0:compile
|  +- org.apache.pdfbox:fontbox:jar:1.6.0:compile
|  +- org.apache.pdfbox:jempbox:jar:1.6.0:compile
|  \- commons-logging:commons-logging:jar:1.1.1:compile
+- org.bouncycastle:bcmail-jdk15:jar:1.45:compile
+- org.bouncycastle:bcprov-jdk15:jar:1.45:compile
+- org.apache.poi:poi:jar:3.8-beta5:compile
+- org.apache.poi:poi-scratchpad:jar:3.8-beta5:compile
+- org.apache.poi:poi-ooxml:jar:3.8-beta5:compile
|  +- org.apache.poi:poi-ooxml-schemas:jar:3.8-beta5:compil
|  |  \- org.apache.xmlbeans:xmlbeans:jar:2.3.0:compile
|  \- dom4j:dom4j:jar:1.6.1:compile
+- org.apache.geronimo.specs:geronimo-stax-api_1.0_spec:jar
+- org.ccil.cowan.tagsoup:tagsoup:jar:1.2.1:compile
+- asm:asm:jar:3.1:compile
+- com.googlecode.mp4parser:isoparser:jar:1.0-beta-5:compil
|  \- net.sf.scannotation:scannotation:jar:1.0.2:compile
|     \- javassist:javassist:jar:3.6.0.GA:compile
+- com.drewnoakes:metadata-extractor:jar:2.4.0-beta-1:compi
+- de.l3s.boilerpipe:boilerpipe:jar:1.1.0:compile
+- rome:rome:jar:0.9:compile
|  \- jdom:jdom:jar:1.0:compile
+- org.gagravarr:vorbis-java-core:jar:0.1:compile
+- junit:junit:jar:4.10:test
|  \- org.hamcrest:hamcrest-core:jar:1.1:test
+- org.mockito:mockito-core:jar:1.7:test
|  \- org.objenesis:objenesis:jar:1.0:test
\- org.slf4j:slf4j-log4j12:jar:1.5.6:test
   \- log4j:log4j:jar:1.2.14:test
```

2. Maven Default Lifecycle

- validate
  initialize
- generate-sources
  process-sources
  generate-resources
  process-resources
- compile
  process-classes

2. Maven Default Lifecycle

- generate-test-sources
  process-test-sources

  generate-test-resources
  process-test-resources

  test-compile
  process-test-classes

  test

to be continued

3. Unit Tests Responsibilities

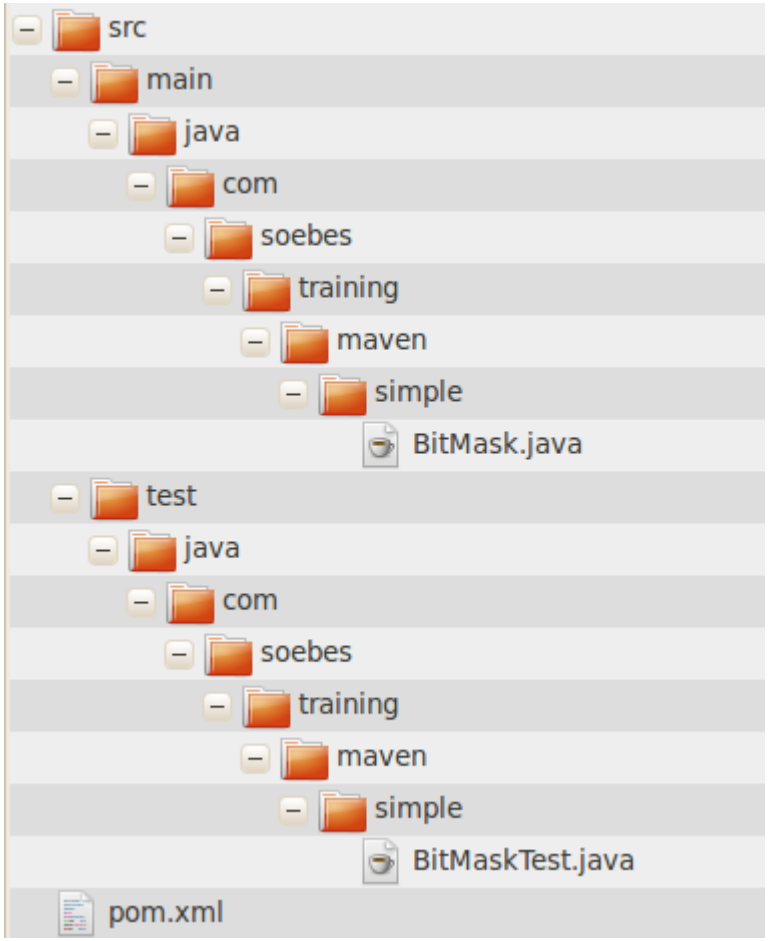
- Maven-resources-plugin
  - Goal: resources:testResources
    - copying resources from src/test/resources folder to target/test-classes/
  - Lifecycle-Phase: process-test-resources
- Maven-compiler-plugin
  - Goal: compiler:testCompile
    - Compiling test code src/test/java to target/test-classes
  - Lifecycle-Phase: test-compile

3. Unit Tests Responsibilities

- Maven-surefire-plugin
  - Goal: surefire:test
    - Execute the unit tests
  - Lifecycle-Phase: test

3. Unit Tests Basic Structure

- Execution of the unit tests is done by:
  - maven-surefire-plugin
- The naming convention for unit tests:
  - **/Test*.java
  - **/*Test.java
  - **/*TestCase.java



3. Unit Tests Basic Structure

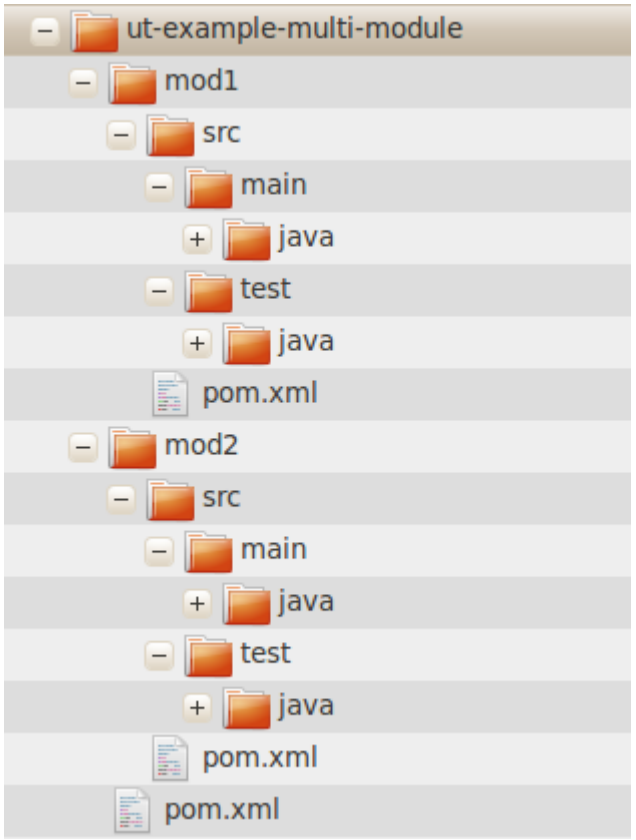
- Execution:

```
[INFO] --- maven-surefire-plugin:2.12:test (default-test) @ unit-test-example ---
[INFO] Surefire report directory: /example/src/main/resources/ut-example/target/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running com.soebes.training.maven.simple.BitMaskTest
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.089 sec

Results :

Tests run: 5, Failures: 0, Errors: 0, Skipped: 0
[INFO]
```

See real demo output

4. Unit Tests – Multi Module Basic Structure

- Execution of the unit tests is done module by module.
- Every module runs its unit test separately.



4. Unit Tests – Multi Module Execution

```
[INFO] --- maven-surefire-plugin:2.12:test (default-test) @ ut-example-mm-mod1 ---
[INFO] Surefire report directory: /project/mod1/target/surefire-reports
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running com.soebes.training.maven.simple.BitMaskTest
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time
elapsed: 0.066 sec

Results :

Tests run: 5, Failures: 0, Errors: 0, Skipped: 0
…
…
```

4. Unit Tests – Multi Module Execution

....
[INFO] --- maven-surefire-plugin:2.12:test (default-test) @ ut-example-mm-mod2 ---
[INFO] Surefire report directory: /project/mod2/target/surefire-reports

---
 T E S T S

Running com.soebes.training.maven.simple.BitMaskTest
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.078 sec

Results :

Tests run: 5, Failures: 0, Errors: 0, Skipped: 0

…

See real demo...

4. Unit Tests – Multi Module Common Code

- What about common code in unit tests?
  - Sometimes the problem occurs to have unit test code in common between modules and of course don't want to duplicate it.

  How to solve this?

4. Unit Tests – Multi Module Common Code

- In the module (mod-ut-propagate) you want to propagate code from to others:

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-jar-plugin</artifactId>
  <executions>
    <execution>
      <goals>
        <goal>test-jar</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

4. Unit Tests – Multi Module Common Code

- In the module you want to use the code of the other module:

```
<dependency>
  <groupId>${project.groupId}</groupId>
  <artifactId>mod-ut-propagate</artifactId>
  <version>${project.version}</version>
  <type>test-jar</type>
  <scope>test</scope>
</dependency>
```

4. Unit Tests – Multi Module Common Code

- Propagate all classes (src/test/java as well as src/test/resources) of the module mod-ut-propagate to other modules classpath which are using it.

- Sometimes you don't like that.
  - The solution for this problem create a separate unit-test-common module.

4. Unit Tests – Company Wide Common Code

- Create a separate project in Maven
- Create a release of it
- And use it as usual dependency

```
<dependency>
  <groupId>com.soebes.modules</groupId>
  <artifactId>unit-test</artifactId>
  <version>1.0.0</version>
  <type>test-jar</type>
  <scope>test</scope>
</dependency>
```

2. Maven Default Lifecycle – Part 2

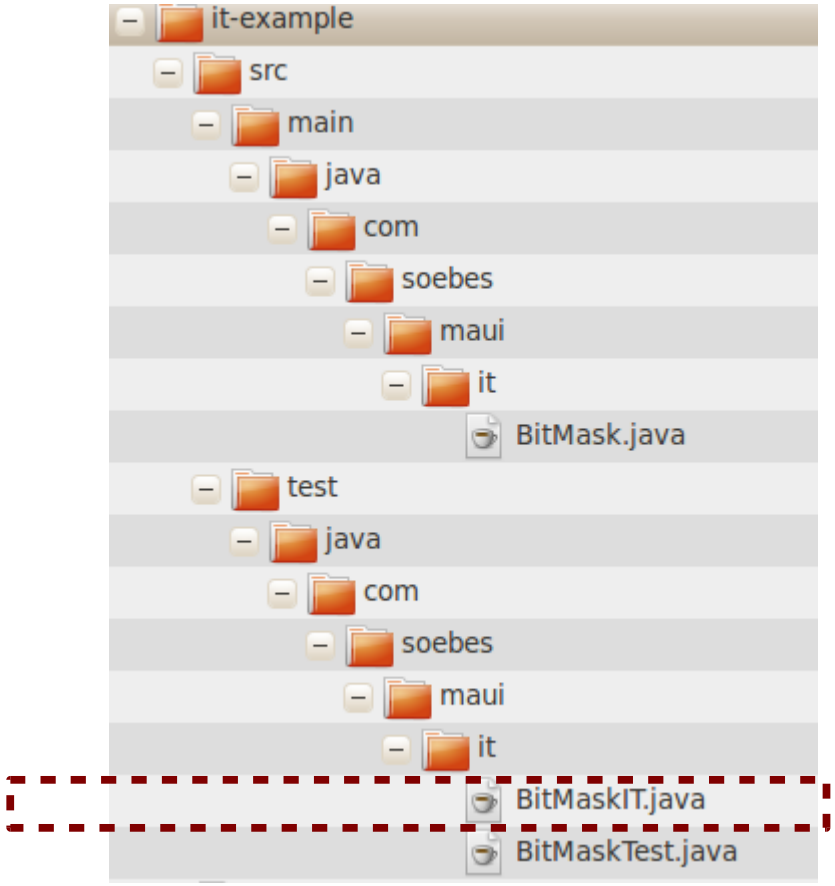
- prepare-package
  package
- pre-integration-test
  integration-test
  post-integration-test
- verify
- install
  deploy

5. Integration Tests Responsibilities

- Maven-failsafe-plugin
  - Goal: failsafe:integration-test
    - Execute the integration tests
  - Lifecycle-Phase: integration-test

5. Integration Tests Basic Structure

- Execution of the unit tests is done by:
  - maven-failsafe-plugin
- The naming convention for integration tests:
  - **/IT*.java
  - **/*IT.java
  - **/*ITCase.java



5. Integration Tests Basic Structure

- The configuration of maven-failsafe-plugin is needed as follows:
- Target folder for compiled classes: **target/test-classes**.

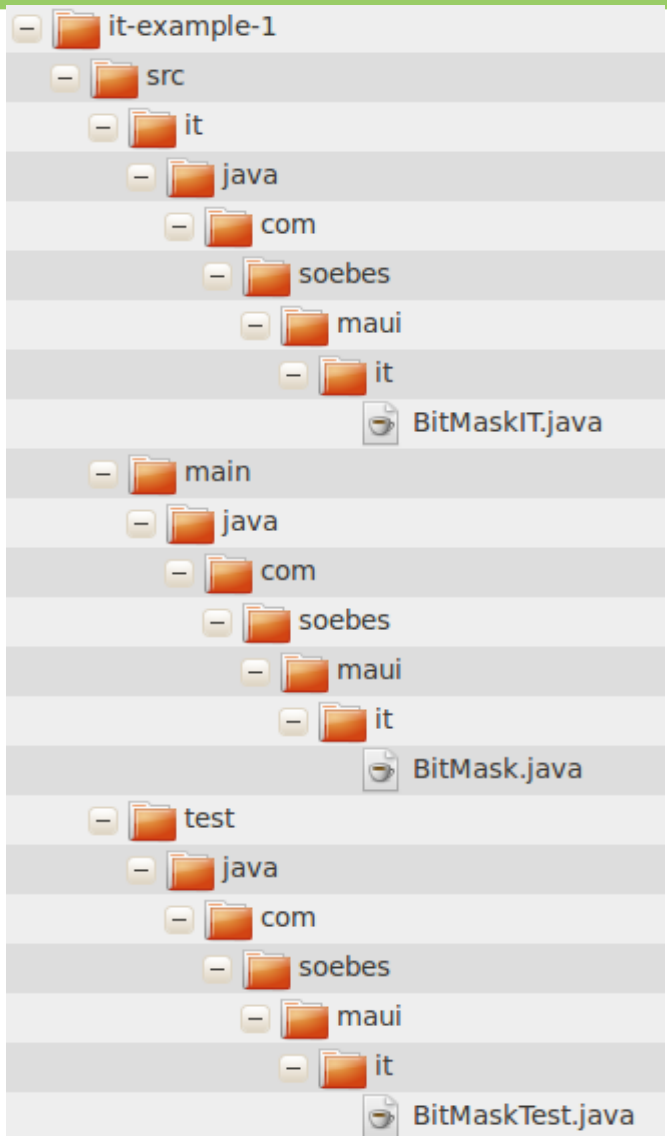
```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-failsafe-plugin</artifactId>
  <version>2.12</version>
  <executions>
    <execution>
      <id>integration-test</id>
      <goals>
        <goal>integration-test</goal>
      </goals>
    </execution>
    <execution>
      <id>verify</id>
      <goals>
        <goal>verify</goal>
      </goals>
    </execution>
  </executions>
</plugin>
```

5. Integration Tests Basic Structure

```
[INFO]
[INFO] --- maven-surefire-plugin:2.10:test (default-test) @ it-test-example ---
[INFO] Surefire report directory: /home/kama/ws-git/maui/src/main/resources/it-example/target/surefire-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running com.soebes.maui.it.BitMaskTest
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.144 sec

Results :

Tests run: 5, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.3.2:jar (default-jar) @ it-test-example ---
[INFO] Building jar: /home/kama/ws-git/maui/src/main/resources/it-example/target/it-test-example-0.1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-failsafe-plugin:2.12:integration-test (integration-test) @ it-test-example ---
[INFO] Failsafe report directory: /home/kama/ws-git/maui/src/main/resources/it-example/target/failsafe-reports

-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running com.soebes.maui.it.BitMaskIT
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.088 sec

Results :

Tests run: 5, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-failsafe-plugin:2.12:verify (verify) @ it-test-example ---
[INFO] Failsafe report directory: /home/kama/ws-git/maui/src/main/resources/it-example/target/failsafe-reports
```

5. Integration Tests Separate Source Folder

- Separate source folder for integration tests.



5. Integration Tests Separate Source Folder

- Use build-helper-maven-plugin to add the separate source location.

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>build-helper-maven-plugin</artifactId>
  <version>1.5</version>
  <executions>
    <execution>
      <id>add-test-source</id>
      <phase>generate-test-sources</phase>
      <goals>
        <goal>add-test-source</goal>
      </goals>
      <configuration>
        <sources>
          <source>src/it/java</source>
        </sources>
      </configuration>
    </execution>
  </executions>
</plugin>
```

5. Integration Tests Separate Module

- Define a separate module with the integration tests.
- Separation of
  - compiling
  - configuration
  - Running
- activate/deactivate via profile.

See Example

6. Maven Plugin Development Integration Tests

- Typical Integration Tests for plugins should simulate a full Maven environment with:
  - A Repository
  - Calling different lifecycle-phases and/or goals
  - Checking the results
  - Etc.
- The maven-invoker-plugin is intended for such purposes.

6. Maven Plugin Development IT's - Basic Structure

```
+- pom.xml
+- src/
   +- it/
      +- settings.xml
      +- first-it/
      |  +- pom.xml
      |  +- src/
      |  +- invoker.properties
      |  +- verify.bsh
      +- second-it/
         +- pom.xml
         +- invoker.properties
         +- verify.bsh
         +- src/
```

6. Maven Plugin Development IT's - Repository

- settings.xml is used to simulate a Maven Remote Repository during the integration tests.
  - It's using the users local maven repository ($HOME/.m2/repository).

```
<?xml version="1.0"?>
<settings>
  <profiles>
    <profile>
      <id>it-repo</id>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>
      <repositories>
        <repository>
          <id>local.central</id>
          <url>@localRepositoryUrl@</url>
          <releases>
            <enabled>true</enabled>
          </releases>
          <snapshots>
            <enabled>true</enabled>
          </snapshots>
        </repository>
      </repositories>
      <pluginRepositories>
        <pluginRepository>
          <id>local.central</id>
          <url>@localRepositoryUrl@</url>
          <releases>
            <enabled>true</enabled>
          </releases>
          <snapshots>
            <enabled>true</enabled>
          </snapshots>
        </pluginRepository>
      </pluginRepositories>
    </profile>
  </profiles>
</settings>
```

6. Maven Plugin Development Integration Tests

- "first-it" describes a usual Maven project.

```
+- pom.xml
+- src/
   +- it/
      +- settings.xml
      +- first-it/
      |  +- pom.xml
      |  +- src/
      |  +- invoker.properties
      |  +- verify.bsh
      +- second-it/
         +- pom.xml
         +- invoker.properties
         +- verify.bsh
         +- src/
```

6. Maven Plugin Development Integration Tests

- The pom.xml is more or less a usual pom.xml except:
  - @project.groupId@
  - @project.artifactId@
  - @project.version.@

```
<project
  xmlns="http://maven.apache.org/POM/4.0.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.or

  <modelVersion>4.0.0</modelVersion>

  <groupId>com.soebes.maven.plugins</groupId>
  <artifactId>basicTest</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <name>Maven Echo Plugin - BasicTest</name>
  <url>http://maven.apache.org</url>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  </properties>

  <build>
    <plugins>
      <plugin>
        <groupId>@project.groupId@</groupId>
        <artifactId>@project.artifactId@</artifactId>
        <version>@project.version@</version>
        <executions>
          <execution>
            <id>echo-first-time</id>
            <phase>validate</phase>
            <goals>
              <goal>echo</goal>
            </goals>
            <configuration>
              <echos>
                <echo>This message is very early in the build process.</echo>
              </echos>
            </configuration>
          </execution>
          <execution>
            <id>echo-info</id>
```

6. Maven Plugin Development Integration Tests

```
<project>
  ...
  <build>
    <plugins>
      <plugin>
        <artifactId>maven-invoker-plugin</artifactId>
        <version>1.6</version>
        <configuration>
          <projectsDirectory>src/it</projectsDirectory>
          <cloneProjectsTo>${project.build.directory}/it</cloneProjectsTo>
          <pomIncludes>
            <pomInclude>*/pom.xml</pomInclude>
          </pomIncludes>
          <settingsFile>src/it/settings.xml</settingsFile>
          <localRepositoryPath>${project.build.directory}/local-repo</localRepositoryPath>
          <postBuildHookScript>verify.bsh</postBuildHookScript>
        </configuration>
        <executions>
          <execution>
            <id>integration-test</id>
            <goals>
              <goal>install</goal>
              <goal>run</goal>
            </goals>
          </execution>
        </executions>
      </plugin>
    </plugins>
  </build>
  ...
</project>
```

6. Maven Plugin Development Integration Tests

- Problem
  - Artifacts are not in the local repository
- Solution
  - Usage of the Mock Repository Manager Plugin

Questions?

- Contact:

  apachecon2012@soebes.de

References

- Maven Homepage
  - http://maven.apache.org
- Mailing lists (User)
  - http://maven.apache.org/mail-lists.html
- Maven References
  - http://www.sonatype.com/Support/Books