

MRJARs with Maven



1

Outline

MRJAR using Maven

- Overview of Maven plugins
- Configuring Maven Compiler plugin for MRJARs

M. Romdhani, 2021

2

2

Overview of Maven Plugins

3

How Maven Plugins Work ?

MRJAR using Maven

- **Maven is actually a plugin execution framework where every task is actually done by plugins. Maven Plugins are generally used to –**

- compile code files
- unit testing of code
- create jar file
- create war file
- create project documentation
- create project reports

- **A plugin generally provides a set of goals, which can be executed using the following syntax –**

```
mvn [plugin-name]:[goal-name]
```

For example, a Java project can be compiled with the maven-compiler-plugin's compile-goal using.

```
mvn compiler:compile
```

M.Romdhani, 2021

4

4

MRJAR using Maven

How Maven Plugins Work ?

- **Exercise : configure the exec:java**

```

<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>exec-maven-plugin</artifactId>
  <version>3.0.0</version>
  <configuration>
    <mainClass>be.justice.MyApp</mainClass>
  </configuration>
</plugin>

```

 - So, to trigger the plugin from the command line, just run:

```
mvn exec:java
```

Now, if you want to execute the exec:java goal as part of your standard build, you'll need to bind the goal to a particular phase of the default lifecycle.

M.Romdhani, 2021
5

5

MRJAR using Maven

Maven default lifecycle

- **Lifecycle is composed of Phases**
- **For each phase, a specific Plugin:goal is performs the processing**

```

graph TD
    subgraph Phases
        direction TB
        P1[process-resources]
        P2[compile]
        P3[process-classes]
        P4[process-test-resources]
        P5[test-compile]
        P6[test]
        P7[prepare-package]
        P8[package]
    end
    subgraph Goals
        direction TB
        G1[resources:resources]
        G2[compiler:compile]
        G3[resources:testResources]
        G4[compiler:testCompile]
        G5[surefire:test]
        G6[jar:jar]
    end
    P1 --- G1
    P2 --- G2
    P4 --- G3
    P5 --- G4
    P6 --- G5
    P8 --- G6
  
```

The diagram illustrates the Maven default lifecycle as a sequence of phases. Each phase is represented by a grey box, and the corresponding plugin goal is shown in a blue box next to it. The phases and goals are: process-resources (resources:resources), compile (compiler:compile), process-classes, process-test-resources (resources:testResources), test-compile (compiler:testCompile), test (surefire:test), prepare-package, and package (jar:jar). A legend indicates that grey boxes represent 'Lifecycle Phase' and blue boxes represent 'Plugin: Goal'. An arrow points from the start of the sequence to the 'package' phase.

M.Romdhani, 2021
6

6

Maven default lifecycle

MRJAR using Maven

■ Compiler Plugging is bound to these phases:

- compile (goal compile)
- test-compile (goal test-compile)

■ Exercise : Bind the exec plugin to the package phase

```
<plugin>
  <groupId>org.codehaus.mojo</groupId>
  <artifactId>exec-maven-plugin</artifactId>
  <version>3.0.0</version>
  <executions>
    <execution>
      <id>my-execution</id>
      <phase>package</phase>
      <goals>
        <goal>java</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <mainClass>be.justice.MyApp</mainClass>
  </configuration>
</plugin>
```

- mvn package command will trigger the plugin

M.Romdhani, 2021

7

7

Compiler plugin Configurations and Executions

MRJAR using Maven

■ Configure Compiler to use Java 9 for the goal compile

```
<plugin>
  <artifactId>maven-compiler-plugin</artifactId>
  <version>3.8.0</version>
  <executions>
    <execution>
      <id>compile-java-9</id>
      <goals>
        <goal>compile</goal>
      </goals>
      <configuration>
        <release>9</release>
      </configuration>
    </execution>
  </executions>
</plugin>
```

M.Romdhani, 2021

8

8

Configure Compiler Plugin for MRJARs

9

MRJAR using Maven

Configure Compiler Plugin for MRJARs

■ Steps

1. Configure how Compiler Plugin should build the releases
 - Each release should have an execution entry
2. Configure how JAR Plugin should produce the MRJAR

M.Romdhani, 2021

10

10

Step 1. Building the MRJAR

MRJAR using Maven

```

<executions>
  <execution>
    <id>compile-java-8</id>
    <goals>
      <goal>compile</goal>
    </goals>
    <configuration>
      <release>8</release>
    </configuration>
  </execution>
  <execution>
    <id>compile-java-9</id>
    <phase>compile</phase>
    <goals>
      <goal>compile</goal>
    </goals>
    <configuration>
      <release>9</release>
      <compileSourceRoots>
        <compileSourceRoot> ${project.basedir}/src/main/java9
      </compileSourceRoots>
      <outputDirectory>${project.build.outputDirectory}/META-INF/versions/9
    </configuration>
  </execution>
</executions>

```

■ Invoke the plugin with :
mvn compile

M.Romdhani, 2021

11

11

MRJAR using Maven

M.Romdhani, 2021

12

12

Step 2. MR Jarring

MRJAR using Maven

■ JAR Plugin configuration for MRJAR

```
<plugin>
  <artifactId>maven-jar-plugin</artifactId>
  <version>3.0.2</version>
  <configuration>
    <release>9</release>
    <compileSourceRoots>
      <compileSourceRoot>
        ${project.basedir}/src/main/java9
      </compileSourceRoot>
    </compileSourceRoots>
    <multiReleaseOutput>true</multiReleaseOutput>
  </configuration>
</plugin>
```

■ Invoke the plugin with :

mvn package

M.Romdhani, 2021

13

13

Testing the MRJAR

MRJAR using Maven

■ The sample project : MyProject-Java-9-MRJAR-Maven

■ Calling the java-9 release

```
java -cp MyProject-Java-9-MRJAR-Maven-0.0.1-SNAPSHOT.jar be.justice.App
```

■ Calling the java-8 release

```
"C:\Program Files\OpenJDK\openjdk-8u292-b10\bin\java"
-cp MyProject-Java-9-MRJAR-Maven-0.0.1-SNAPSHOT.jar be.justice.App
```

M.Romdhani, 2021

14

14