

Maven Local Repository

The maven local repository is a local folder that is used to store all your project's dependencies (plugin jars and other files which are downloaded by Maven). In simple, when you build a Maven project, all dependency files will be stored in your Maven local repository.

By default, Maven local repository is default to `.m2` folder:

1. Unix/Mac OS X – `~/.m2/repository`
(`/Users/BhaskarReddy/.m2/repository`)
2. Windows – `C:\Documents and Settings\{your-username}\.m2/repository`

Normally, We will change the default local repository folder from default `.m2` to another more meaningful name as follows.

Find `{M2_HOME}\conf\setting.xml`, update `localRepository` to something else.

`<!-- localRepository`

The path to the local repository maven will use to store artifacts.

Default: `${user.home}/.m2/repository`

`<localRepository>/path/to/local/repo</localRepository>`

`-->`

`<localRepository>D:/maven_custom_repo</localRepository>`

Maven with SonarQube

Note: For this we no need to create the `sonar-project.properties` file. It will take all the details from `pom.xml`.

To generate SonarQube scanner report for maven projects need to add below lines in `pom.xml`.

```
<profiles>
  <profile>
    <id>sonar</id>
    <activation>
      <activeByDefault>true</activeByDefault>
    </activation>
```

```
<properties>
  <sonar.host.url>http://localhost:9001/</sonar.host.url>
  <sonar-maven-plugin.version>3.3.0.603</sonar-maven-plugin.version>
</properties>
<build>
  <plugins>
    <plugin>
      <groupId>org.sonarsource.scanner.maven</groupId>
      <artifactId>sonar-maven-plugin</artifactId>
      <version>${sonar-maven-plugin.version}</version>
    </plugin>
  </plugins>
</build>
</profile>
</profiles>
```

Once you add you need to trigger the below command from the project directory to which you are generating report.

```
mvn sonar:sonar
```

(OR)

```
mvn clean install sonar:sonar
```

In SonarQube dashboard you will find the project with the name which you mentioned in **<name>Maven-Java-Project</name>** tag and the key it will take from **<groupId>com.mt</groupId> and <artifactId>maven-java-project</artifactId>** as follows.

com.mt: maven-java-project , and version it will take from **<version>1.0</version>** tag.

Note: If we want to generate sonarqube report for all Maven projects, we have to add above lines in each project, pom.xml. Instead of copying in each pom.xml, just copy into **settings.xml** which is available in **conf** directory.

In two locations **settings.xml** file may be existed.

The Maven install: \${maven.home}/conf/settings.xml

A user's install: \${user.home}/.m2/settings.xml

The former **settings.xml** are also called global settings, the latter **settings.xml** are referred to as user settings. If both files exists, their contents gets merged, with the user-specific **settings.xml** being dominant.

Maven with Remote Repositories

If we want to configure nexus remote repo with Maven, need to follow the below steps.

Remote Repository add

```
<repositories>
  <repository>
    <id>comt.mt</id>
    <url>https://maven.java.net/content/repositories/public/</url>
  </repository>
</repositories>
```

To install the Oracle jdbc drivers :

2.1 ojdbc6.jar

```
$ mvn install:install-file -Dfile={Path/to/your/ojdbc6.jar}
  -DgroupId=com.oracle -DartifactId=ojdbc6 -Dversion=11.2.0 -Dpackaging=jar
```

2.2 ojdbc7.jar

```
$ mvn install:install-file -Dfile={Path/to/your/ojdbc7.jar}
  -DgroupId=com.oracle -DartifactId=ojdbc7 -Dversion=12.1.0 -Dpackaging=jar
```

Full example to install a ojdbc7.jar

Terminal

```
> mvn install:install-file -Dfile=C:\\OracleJDBC\\ojdbc7.jar
```

```
-DgroupId=com.oracle -DartifactId=ojdbc7 -Dversion=12.1.0 -Dpackaging=jar
```

After installation, you must add below dependency in pom.xml as follows.

```
<!-- ojdbc6.jar example -->
```

```
<dependency>
  <groupId>com.oracle</groupId>
  <artifactId>ojdbc6</artifactId>
  <version>11.2.0</version>
</dependency>
```

```
<!-- ojdbc7.jar example -->
```

```
<dependency>
  <groupId>com.oracle</groupId>
  <artifactId>ojdbc7</artifactId>
  <version>12.1.0</version>
</dependency>
```

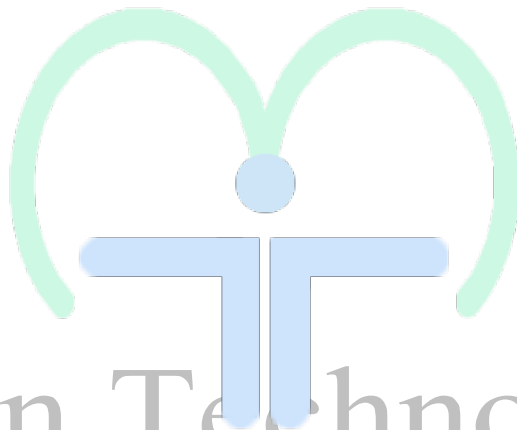
```
mvn install:install-file -DlocalRepositoryPath=repo -DcreateChecksum=true -
Dpackaging=jar -Dfile=/Users/BhaskarReddyL/Jars/BluePagesJavaToolKit-3.jar -
DgroupId=com.mt -DartifactId=BluePagesJavaToolKit-3.jar -Dversion=1.0
```

pom.xml

```
<repositories>
  <repository>
    <id>com.mt</id>
    <url>file://${project.basedir}/repo</url>
  </repository>
</repositories>
```

What is the difference between -DskipTests and -Dmaven.test.skip=true ?

Ans)



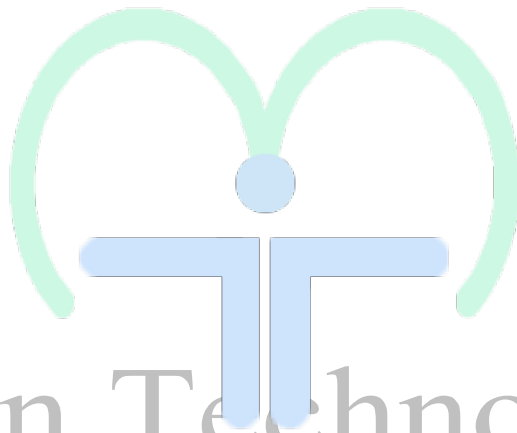
Mithun Technologies
Mithun
Technologies

References

<http://maven.apache.org/settings.html>

<http://maven.apache.org/general.html#available-plugins>

<https://howtodoinjava.com/maven/maven-dependency-scopes/>



Mithun Technologies
Mithun
Technologies