Maven build setup for your project

Prepared by:

Ashish Kumar Mondal

Contents

[Introduction 3](#_Toc415639476)

[Steps to setup maven 3](#_Toc415639477)

[Build a java project 4](#_Toc415639478)

[Call ant build script from maven 5](#_Toc415639479)

[Create a new Maven project for Java 5](#_Toc415639480)

[Troubleshooting 6](#_Toc415639481)

# Introduction

Maven build can be setup and executed from eclipse or cmd prompt. The important items in maven build are

**settings.xml:** Here you can define your local repository, proxy, JDK version etc. Sample settings.xml is attached. You can place your settings.xml file into %maven\_home%/conf/ path.



Please note: If you do not define your localRepository in the settings.xml file then default repository would be %USER\_HOME%/.m2/repository

**pom.xml:** You need to write your script to build your project. It should be placed under project folder.

# Steps to setup maven

1. **Check if maven is available in eclipse**

Normally maven comes with eclipse IDE for J2EE. To check, follow the below steps

1. Open eclipse

2. Go to Windows->Preference

3. Check for Maven entry in the left hand side

If maven plugin is not there with eclipse then install the phugin in Eclipse from Help->Install new software

Lots of materials are available in Google on this topic.

1. **Maven from command prompt**

If you want to build using "mvn install" from cmd prompt then maven should be installed in your system and M2\_HOME has to be setup as environmental variable.

1. **Change location of setting.xml file (if required)**

The default location for the settings.xml is ${user.home}/.m2/settings.xml. However, user can change this location in eclipse/cmd prompt. Changing in eclipse is very eary.

* 1. **for eclipse**: i) Go to Windows->Preference->Maven->User Settings and mention <path to settings.xml>
  2. **for cmd prompt**: settigns.xml should be at {M2\_HOME}\conf\setting.xml

1. **Use local repository**

Use the below tag under <settings> in settings.xml file

<localRepository>path\to\repository</localRepository>

1. **Dependency on Internet connection**

maven must connect to the internet (or copy the working repository then internet connection is not required) for the first time. To connect to the internet: make sure <proxy> tag is correct in your settings.xml which is in use (as per step 3)

*<proxy>*

*<id>optional</id>*

*<active>true</active>*

*<protocol>http</protocol>*

*<username>proxyuser</username>*

*<password>proxypass</password>*

*<host>proxy.host.net</host>*

*<port>80</port>*

*<nonProxyHosts>local.net|some.host.com</nonProxyHosts>*

*</proxy>*

1. **Define maven repository in pom.xml**

In pom.xml, if repository is not mentioned then by default it will look into local repository (as setup in step 4).

If not found then it will look into maven central repository (i.e. http://repo1.maven.org/maven2/). If not found then it will look for maven remote repository. For remote repository, below configuration is required in pom.xml

*<repositories>*

*<repository>*

*<id>java.net</id>*

*<url>https://maven.java.net/content/repositories/public/</url>*

*</repository>*

*</repositories>*

# Build a java project

The below attached script creates jar file and place into another location by the followings steps. Comments added in each steps in the pom.xml file

* + It defines the dependencies (jars) with scope.
    - If the scope is system then it picks the jar from hardcoded location
    - If the scope is test then the testing files does not get copied into the final jar
  + Define JDK version
  + Copy resource files (.properties/.bat) into respective folder
  + Copy dependant jar files into ${outputPath}/lib folder
  + Create executable jar with depandant jars as entry into manifest.mf file
  + Copy the jar into final destination
  + Also declare variables which are used across the pom.xml



# Call ant build script from maven

Sometimes (specially build with jdk1.4) you need to call ant build from pom.xml. Sample code is attached.



# Create a new Maven project for Java

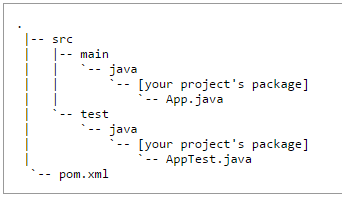
1. Create a project folder
2. Execute the following command to create maven project

mvn archetype:create

-DgroupId=com.test

-DartifactId=ArtifactName

1. This would create the following folder structure



# Troubleshooting

|  |  |  |
| --- | --- | --- |
| **SL No** | **Issue** | **Solution** |
| 1 | Could not calculate build plan: Plugin org.apache.maven.plugins:maven-resources-plugin:2.5 or one of its dependencies could not be resolved | Maven is not able to connect to the internet. Follow the step 5 or copy working repository into your system |
| 2 | 'build.plugins.plugin.version' for org.apache.maven.plugins:maven-compiler-plugin is missing | Add <version> for the plugin. Use  <plugin>  <artifactId>maven-compiler-plugin</artifactId>  <version>2.3.2</version>  **INSTEAD OF**  <plugin>  <artifactId>maven-compiler-plugin</artifactId> |
| 3 | [WARNING] File encoding has not been set, using platform encoding Cp1252, i.e. build is platform dependent! | Add the below line in your pom.xml  <properties>  <project.build.sourceEncoding>  UTF-8  </project.build.sourceEncoding>  </properties> |
|  |  |  |