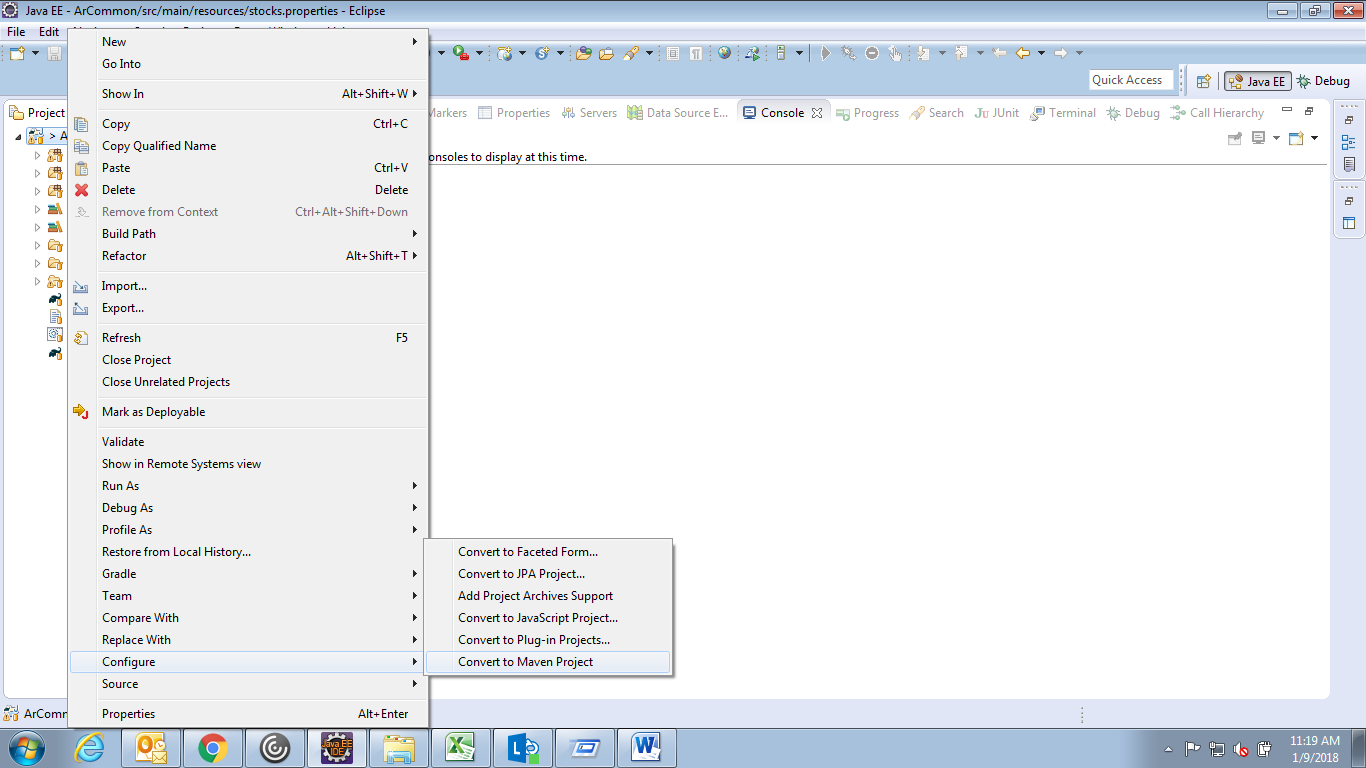
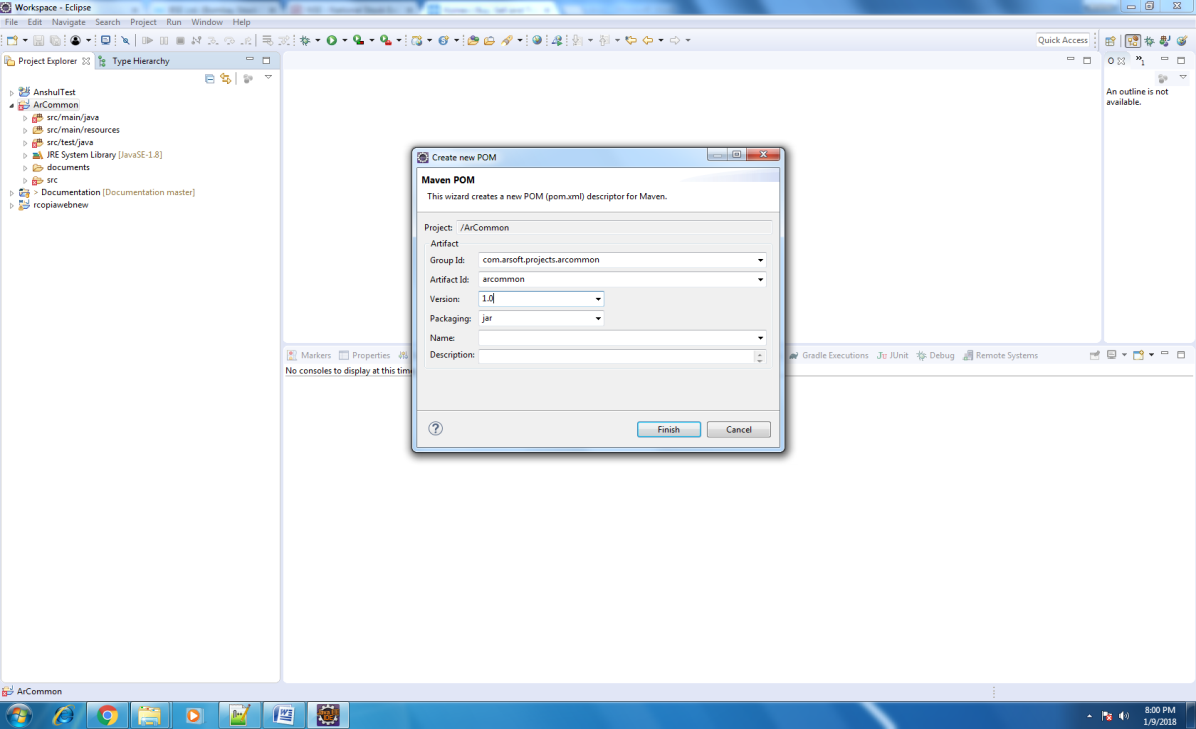
**Convert Gradle Project to Maven Project in Eclipse**

1. Right Click Project.
2. Go to Configure >> Convert to Maven Project
3. In the next window, set the package for the project as Group Id i.e. groupId will identify your project uniquely across all projects e.g. com.arsoft.projects.arcommon
4. Set the name for the jar as Artifact Id e.g. arcommon
5. Set the version for the jar as Version e.g. 1.0
6. Click Ok.



1. Remove entry <classpathentry kind="con" path="org.eclipse.buildship.core.gradleclasspathcontainer"/> from the file .classpath for the project.



1. Click Finish.
2. This creates file pom.xml inside the root of the project.

**How to fix common errors in tomcat**

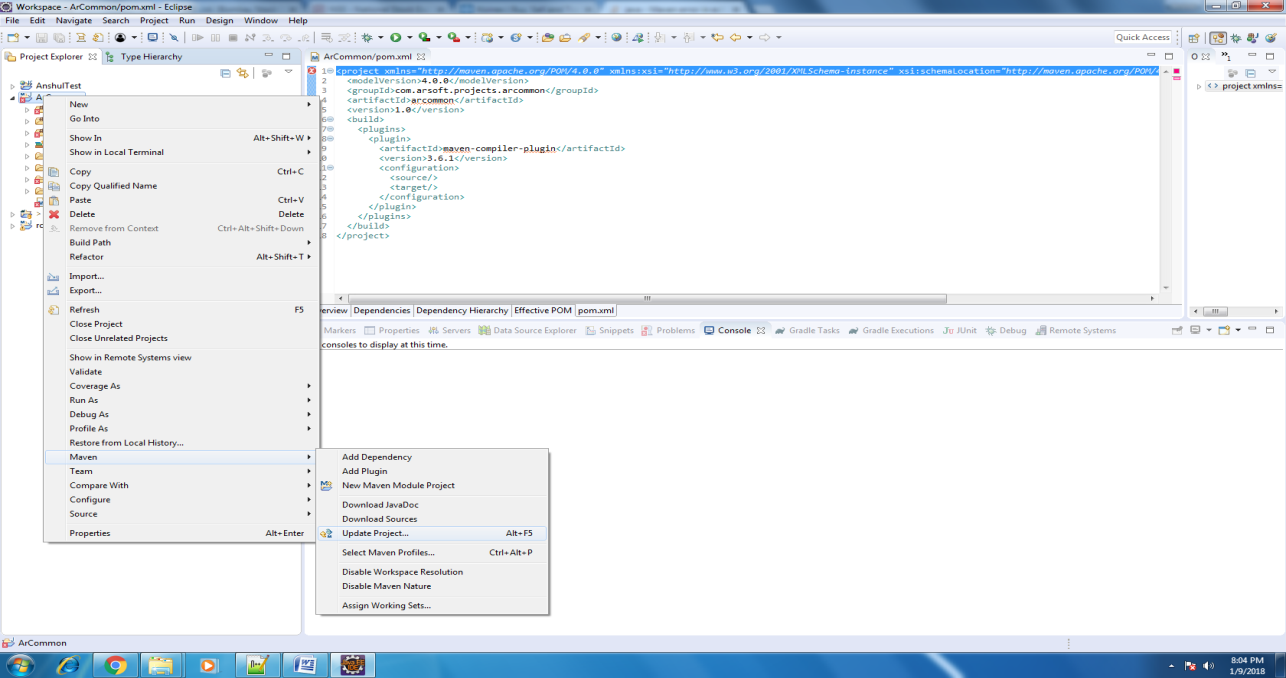
**How to fix common errors in maven**

**Error Description:**

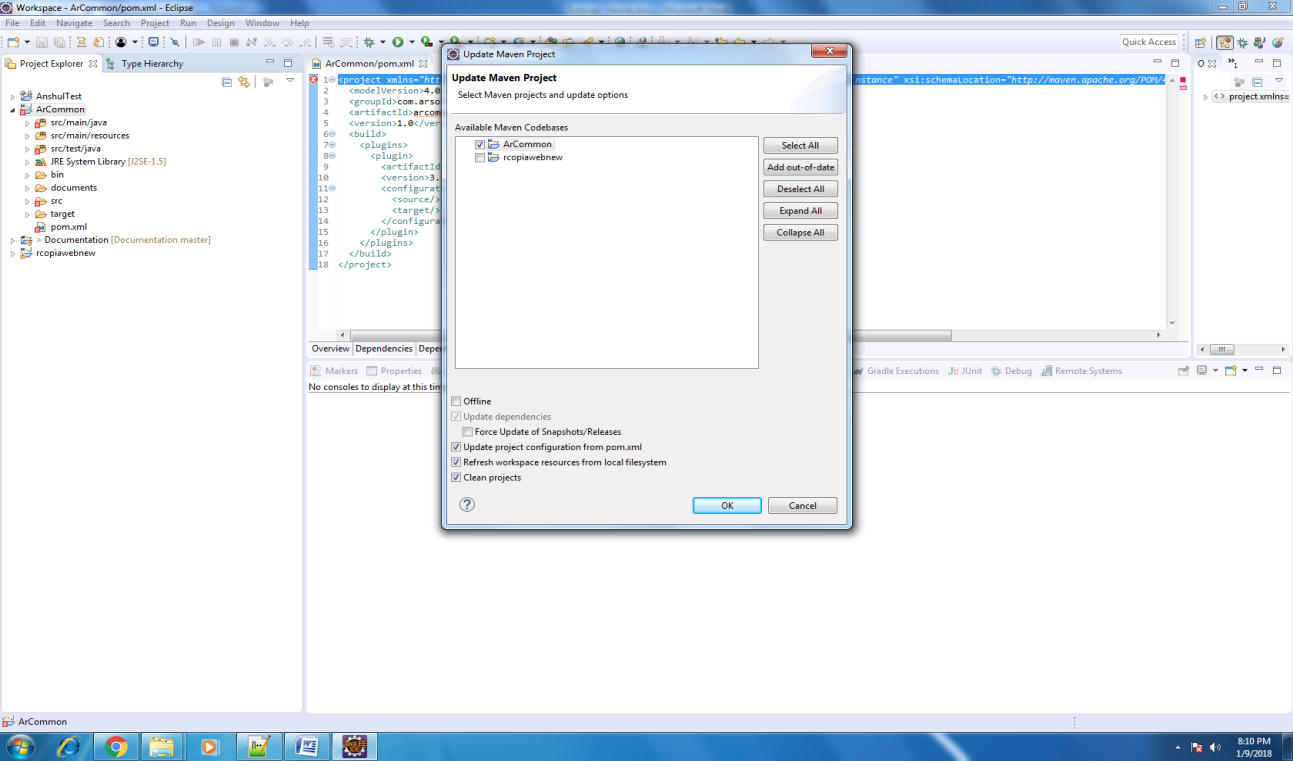
Error in pom.xml Failure to transfer org.apache.maven.plugins:maven-surefire-plugin:pom:2.12.4 from https://repo.maven.apache.org/maven2 was cached in the local repository, resolution will not be reattempted until the update interval of central has elapsed or updates are forced. Original error: Could not transfer artifact org.apache.maven.plugins:maven-surefire-plugin:pom:2.12.4 from/to central (https:// repo.maven.apache.org/maven2): The operation was cancelled.

**Solution:**

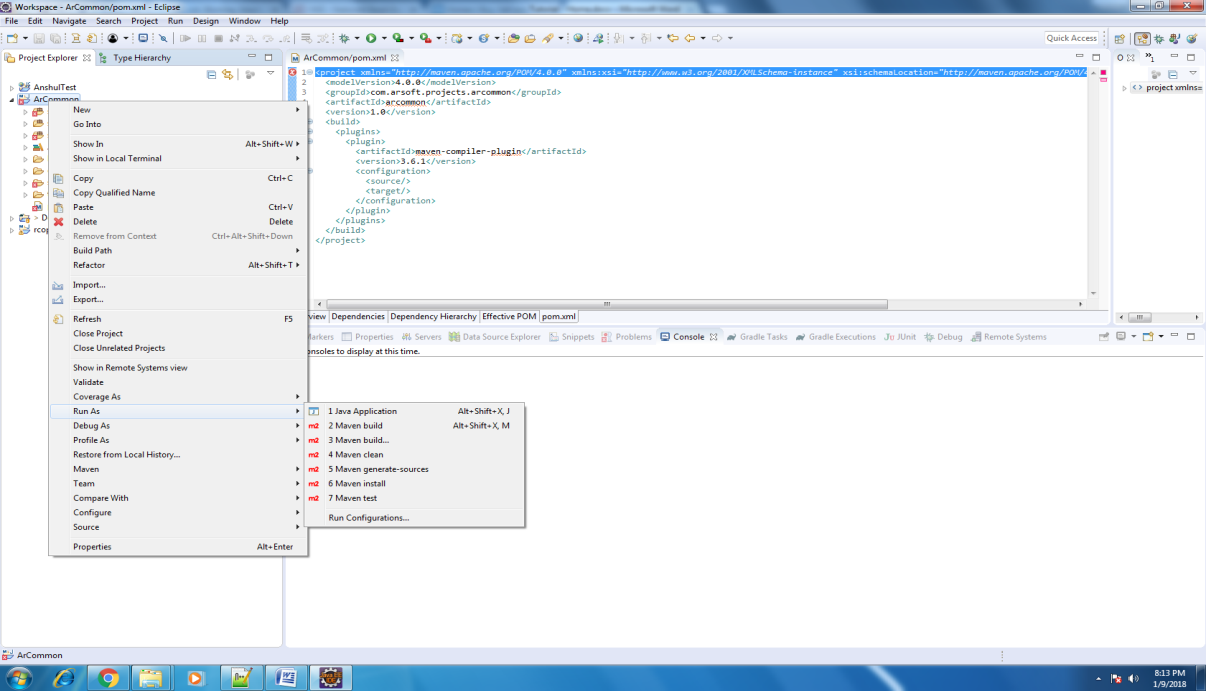
1. Right Click the project.
2. Click Maven >> Update Projects



1. In the next window, select the project.
2. Check the option "Force Update of Snapshots/Releases"
3. Click "Ok"



1. Right click the Project
2. Click Run As >> Maven clean
3. Click Run As >> Maven install



**Error Description:**

Could not find or load main class org.codehaus.plexus.classworlds.launcher.Launcher

**Solution:**

Update the m2e plug-in of the eclipse.

**Error Description:**

Unable to process Jar entry [module-info.class] from Jar ] for annotations org.apache.tomcat.util.bcel.classfile.ClassFormatException: Invalid byte tag in constant pool: 19

**Solution:**

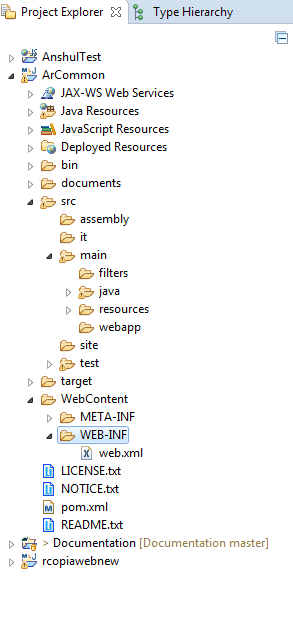
There seems to bug in tomcat maven plugin version 2.2. Used 2.1 or below in maven pom.xml.

**Error Description:**

SEVERE: Error starting static Resources java.lang.IllegalArgumentException: src\main\webapp does not exist or is not a readable directory

**Solution:**

This is because proper directory structure is not created for the web project. Make sure that the folder webapp is present inside the src/main Ideally the folder structure should be as below:



**Error Description:**

Failed to execute goal org.apache.maven.plugins:maven-compiler-plugin:Compilation failure: Compilation failure

Make sure that the classes in the main packages are having all the dependent classes in main only and the classes in test packages are having all the dependent classes in test only.

**Error Description:**

Source option 1.5 is no longer supported. Use 1.6 or later

**Solution:**

Change the compiler version of maven project

Add the below plug in to pom.xml

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>3.6.1</version>

<configuration>

<source>1.8</source>

<target>1.8</target>

</configuration>

</plugin>

or

Open pom.xml and add the below properties inside the properties tag.

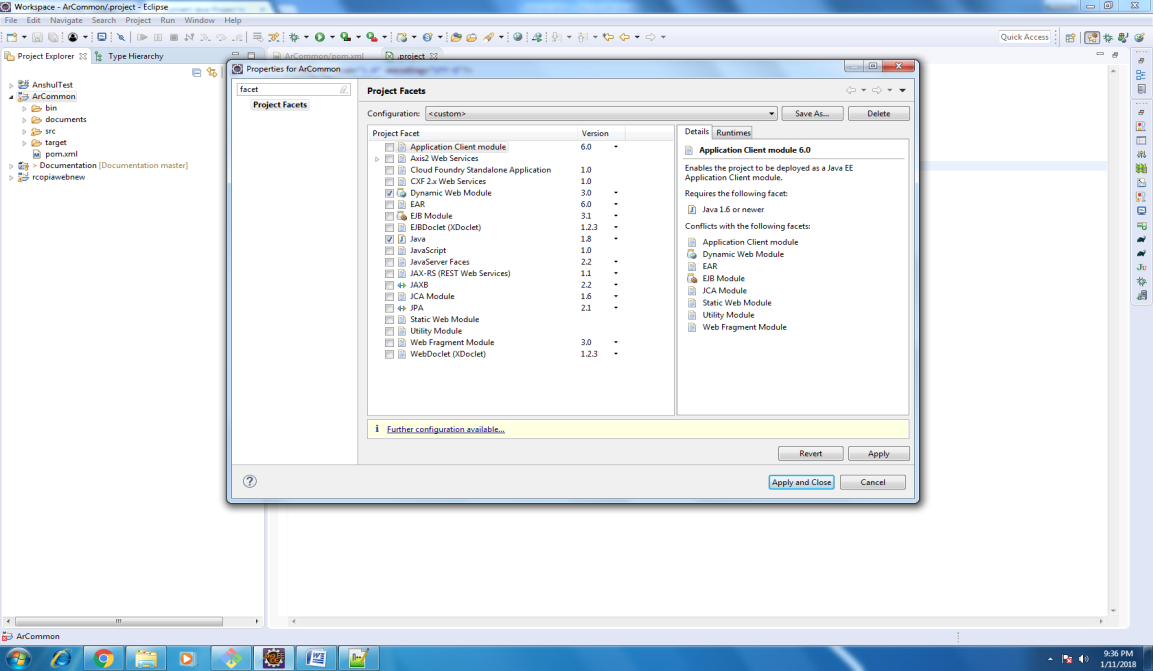
<properties>

<maven.compiler.source>1.8</maven.compiler.source> <maven.compiler.target>1.8</maven.compiler.target>

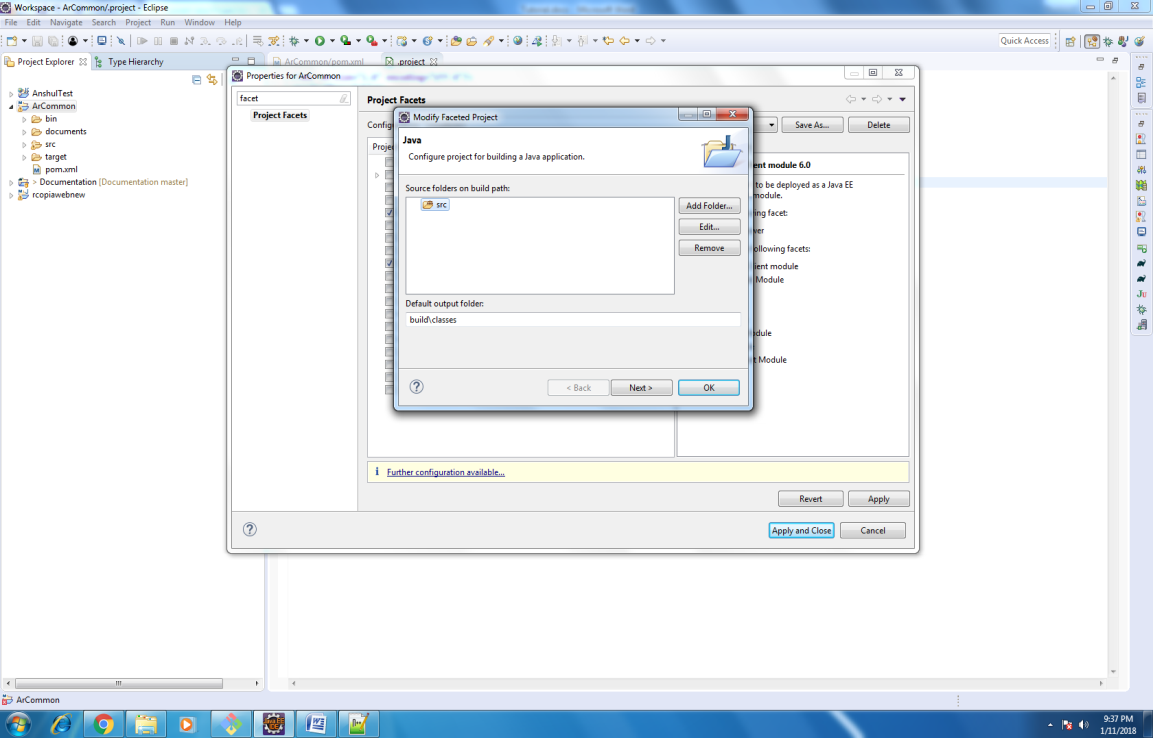
</properties>

**How to convert Java Project to Web Project using maven:**

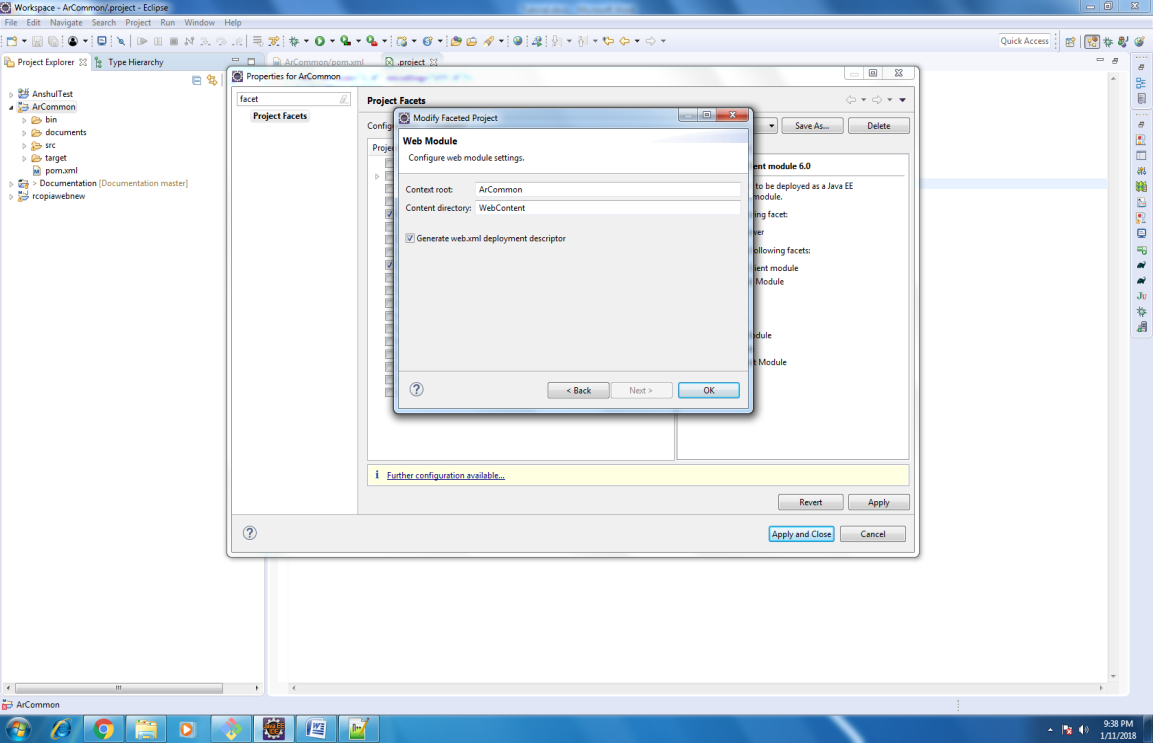
1. Right click the project >> Properties.
2. Search for Project Facets.
3. Select Project Facets "Dynamic Web Module" and "Java".



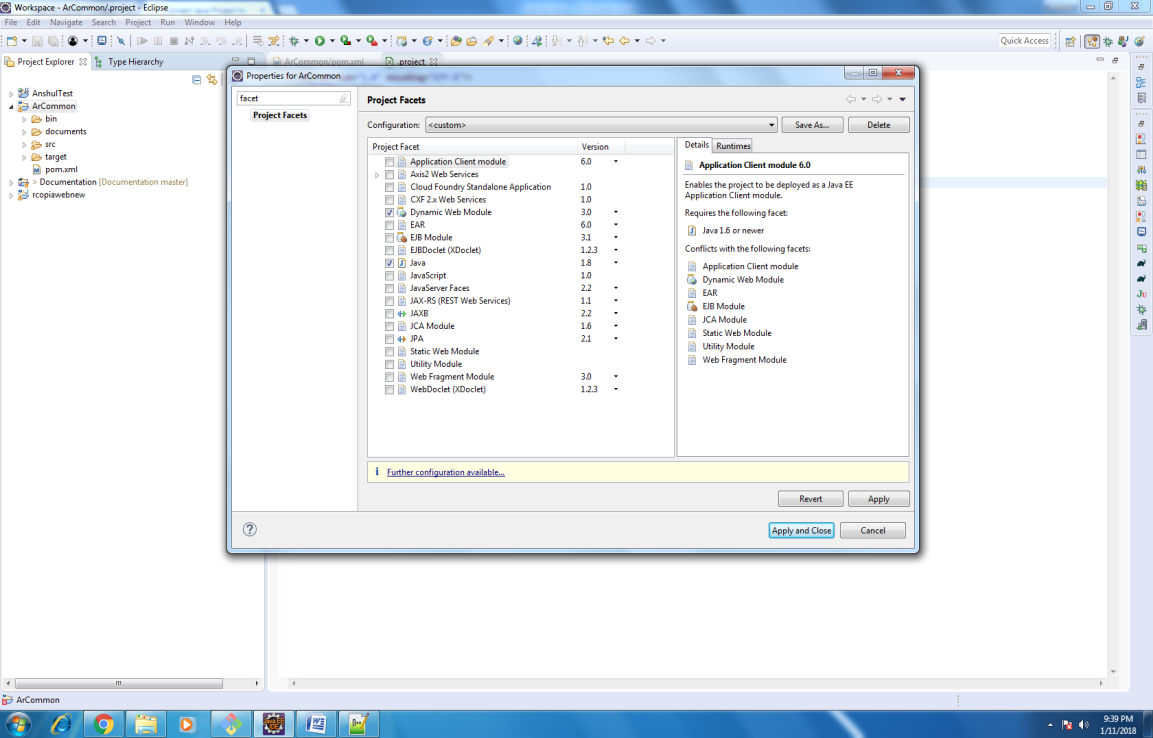
1. Click link "Further Configurations available..".



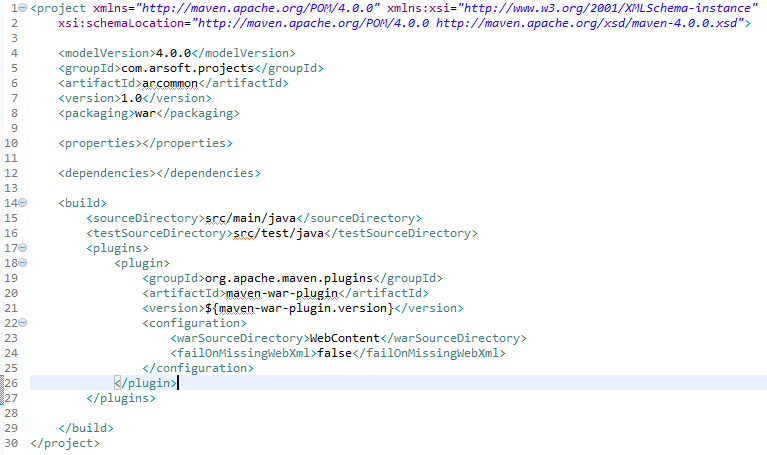
1. Click "Next".



1. Check checkbox "Generate web.xml deployment descriptor".
2. Click ok.



1. Click Apply and Close.
2. Open pom.xml .
3. Add tag <packaging>war</packaging> under the root project tag
4. Add tag <sourceDirectory>src/main/java</sourceDirectory> under build tag.
5. Add tag <testSourceDirectory>src/test/java</testSourceDirectory>
6. under build tag.
7. Add plugin maven-war-plug inside the plugins section of the build tag.



1. Update maven project and run goal maven clean install –U.

**How to start and stop embedded tomcat 7 using maven:**

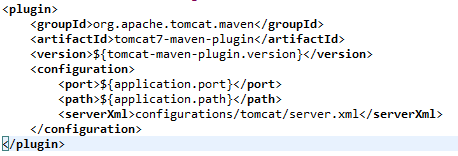
1. Add the plug-in for tomcat in pom.xml plugins section
2. Run goal 'mvn clean install'
3. To start tomcat, run goal 'tomcat7:run'
4. To stop tomcat, run goal 'tomcat7:shutdown'

Configuring tomcat:

Change the default port for application

1. To the maven tomcat plug in configuration add the tag

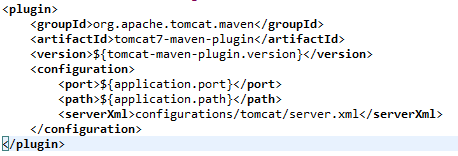
<port>{desired port}</ port >



Change the default path of server.xml

1. To the maven tomcat plug in configuration add the tag

<serverXml>{path of server.xml file}</serverXml>



Change the name of the tomcat log files created inside log directory.

1. Go to server.xml file and change the prefix to desired value.



Change the name of the tomcat log files created inside log directory.