

ORGANISATION	ITC INFOTECH INDIA LIMITED.
DRAFTED BY	SAURAV PAUL.
DATE	21-05-2010.
LAST UPDATED	25-05-2010.

Maven Setup with Eclipse

S/W REQUIRED:JDK 1.5 OR JDK 1.6,ECLIPSE,APACHE TOMCAT SERVER.

ASSUMPTION : JDK 1.5 OR JDK 1.6 & ECLIPSE PRE-INSTALLED.

STEPS TO SETUP MAVEN:

1. INSTALLATION OF MAVEN IN WINDOWS :

Step 1.1 : Download [Maven 2.0.9](#) and unzip it to [C:\](#) drive.

Step 1.2 : Go to **System Properties** → **Environment Variable**.

Step 1.3 : Create an "M2_HOME" User Variable and point it to your Maven installation.I pointed mine to [C:\maven-2.0.9](#).

Step 1.4 : Add %JAVA_HOME%\bin;%M2_HOME%\bin;
To your 'Path' System Variable.This puts your Java and Maven executables in the System Path so that they can be executed without their fully qualified paths.

Step 1.5 : Open a command prompt window, and at the command prompt, type '[mvn -version](#)'.You should see a message displaying the version of Maven [SW](#) and the version of Java.

Step 1.8 : In setting.xml,add the proxy settings in

```
<proxy>
  <id>optional</id>
  <active>true</active>
  <protocol>http</protocol>
  <username></username>
  <password></password>
  <host>10.6.13.77</host>
  <port>8080</port>
  <nonProxyHosts></nonProxyHosts></proxy>
```

Step 1.6 : Now at the command prompt.type '**mvn clean**'.After typing we will probably notice few jars downloaded from the Maven Repository to your machine.After that you will encounter will 'BUILD ERROR' The message is fine.

Step 1.7 : After running the '**mvn clean**',if we go to our home directory(for me :[C:\Documents and Settings\UserName](#)) we can see that an .m2 directory has been created by maven.This is the default location for your local maven repository.This local repository directory contains things such as the jar files that your projects use.

Step 1.9 : Our Maven is **Installed and Working**.

2.INTEGRATION OF MAVEN BUILD TOOL & APACHE TOMCAT SERVER USING ECLIPSE

2.1 SETTING UP MAVEN IN ECLIPSE

Step 2.1.1 : To run Maven from Eclipse,Go to → External Tool Configuration..

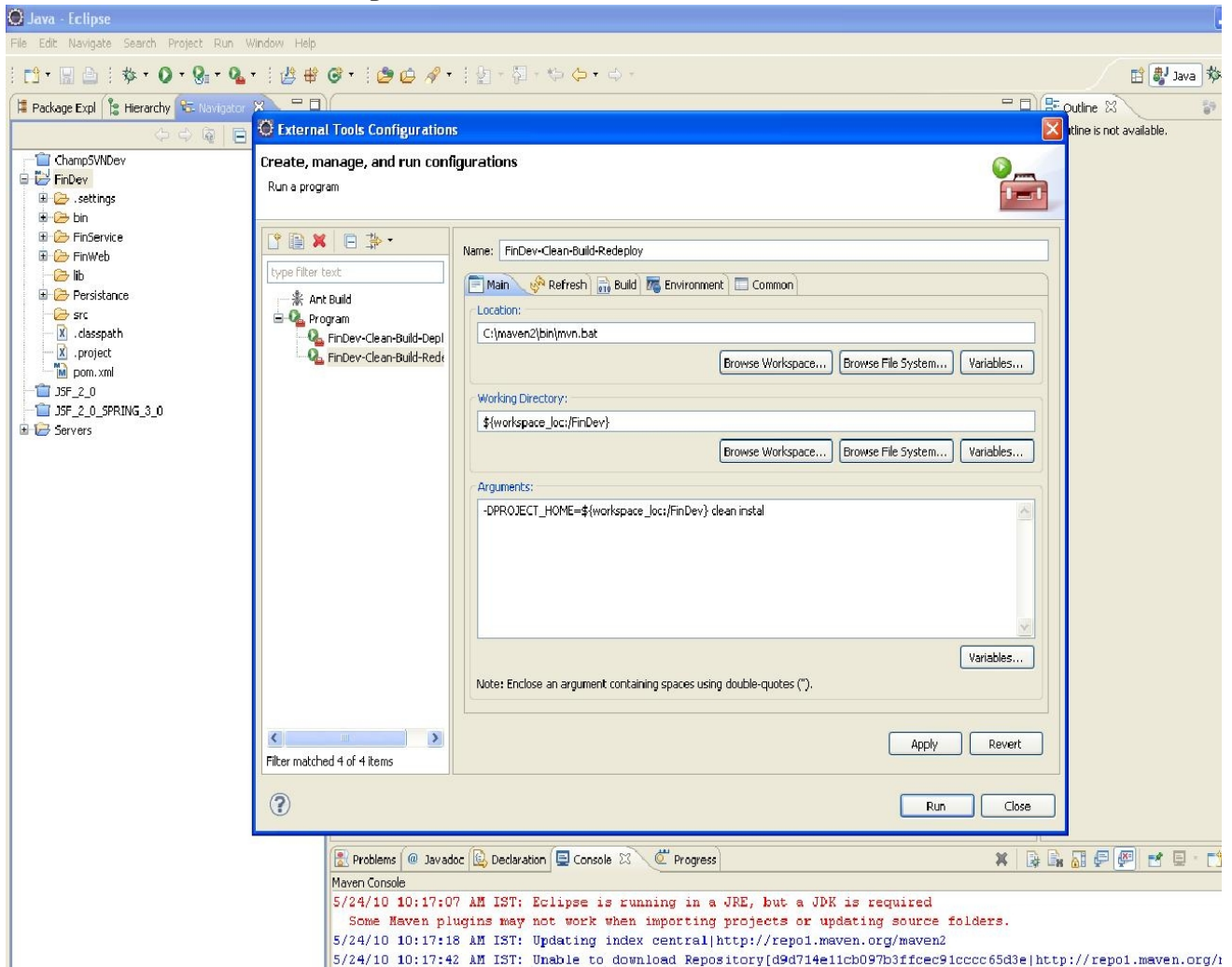
Step 2.1.2 : Right Click on the **Program** --> Click → **New**

Step 2.1.3 : Under **Location** → Give your Maven directory Where you unzipped it.I pointed mine to [C:\maven2\bin\mvn.bat](#)

Under **Working Directory** → `${workspace_loc:/<Project_name>}`

Under **Arguments** → `-DPROJECT_HOME=${workspace_loc:/<Project_name> clean install`

Refer the below Image:

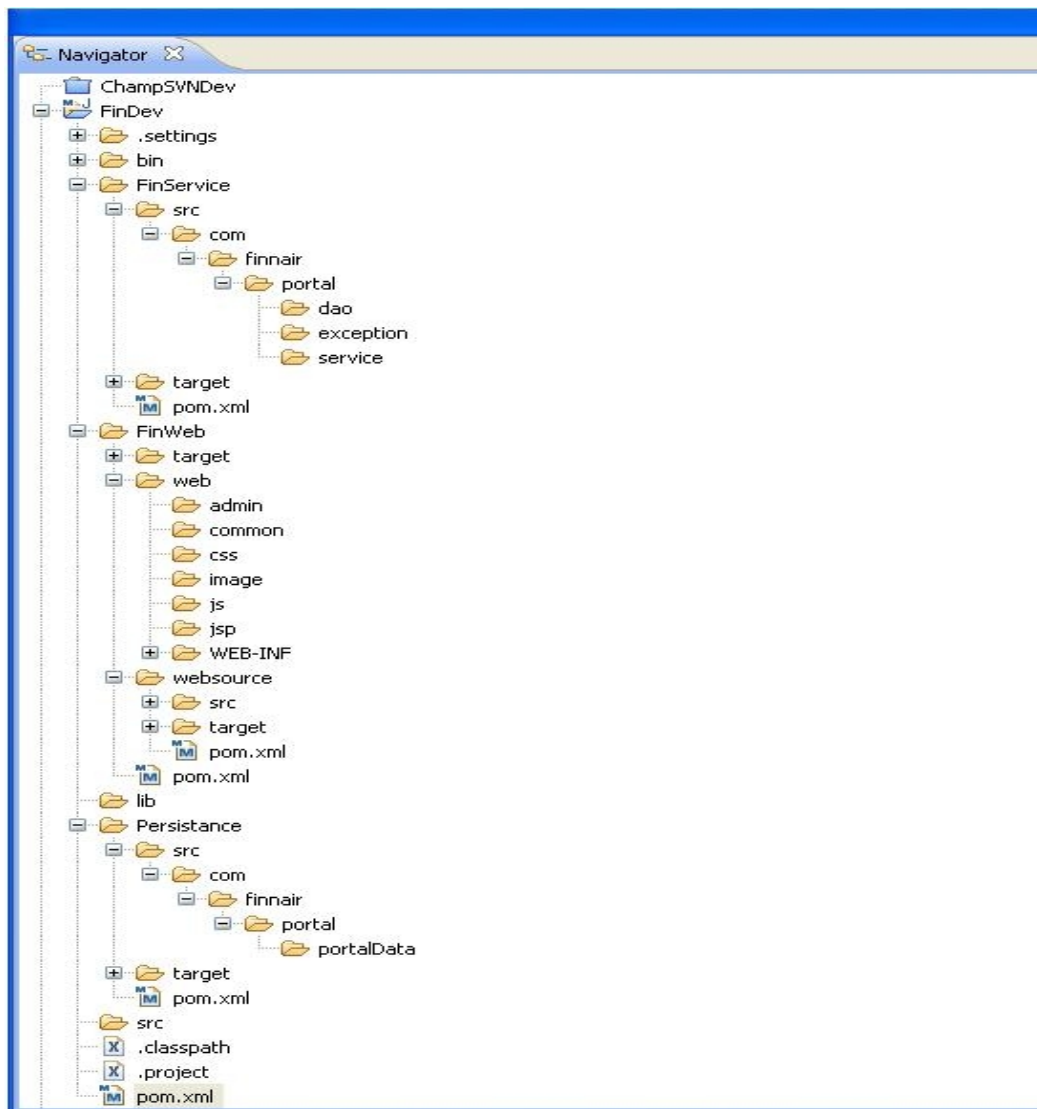


Step 2.1.4 : Under Windows → Preferences → Java → Classpath Variable
Eclipse will automatically take the 'M2_REPO' Path (which is non-modifiable in our case as we don't have admin rights) by default it will be C:\Documents and Settings\firstName.lastName\.m2\repository.

2.2 SETTING UP MAVEN AND TOMCAT THROUGH ECLIPSE :

Step 2.2.1 : After setting up the Maven with Eclipse, we intend to build a WAR file. To build a WAR file, we need to maintain a folder structure. In general, we do prefer to do things the "Maven" way, but as for project layouts, I prefer the "Eclipse" way and use a flat folder structure where all projects are located at the same level and eventually, it permits debugging across the in hassle free manner.

Please refer the below image for skeleton of folder structure which I have maintained:



Step 2.2.2 : To deploy the Web Application to Tomcat from eclipse, All we need is **Tomcat-Maven Plugin**. Download it and provide system path or make it online to access from repository.

Step 2.2.3 : Now, let's see how to setup a maven project to use **maven tomcat plugin**.

Step 2.2.3.1 : Open **tomcat-users.xml** and add the below mentioned code for eg:

```
<tomcat-users>
    <role rolename="manager"/>
    <user username="test" password="test" roles="manager"/>
</tomcat-users>
```

Step 2.2.3.2 : In **setting.xml** of Maven, Server configuration needs to be done by adding few lines of xml tags definition as for eg:

```
server>

    <id>myserver</id>

    <username>test</username>
    <password>test</password>
</server>
```

Step 2.2.3.3 : Next I added a tomcat-maven plugin entry to the "webapp" project's pom.xml. In my case, "**FinDev\FinWeb\pom.xml**"

```
<plugin>
    <groupId>org.codehaus.mojo</groupId>
    <artifactId>tomcat-maven-plugin</artifactId>
    <version>1.0-alpha-1</version>
    <configuration>
        <url>http://localhost:8080/manager</url>
        <server>myserver</server>
        <path>/FinWeb</path>
    </configuration>
</plugin>
```

Step 2.2.4 : Start your Tomcat Server and login into Tomcat Manager using UserName and Password

Step 2.2.5 : To **clean, build and Deploy** in Tomcat Server from Eclipse run your Maven from Eclipse using the Maven command

Command : clean tomcat:undeploy tomcat:deploy

Step: 2.2.6 : Finally, we will be able to see the **WAR** File has deployed in Server.

MAVEN SETUP WITH ECLIPSE