**Webdriver:**

1. **Webdriver** is a tool which only help you to interact with browser. It will not help you to generate reports, build the test cases, batch running the test cases. For all these things you need **TestNG**.

**TestNG:**

Go to http://testng.org/doc/download.html & copy <http://beust.com/eclipse>

Goto Help>Install new Software>

Paste URL and hit Enter

Ignore errors and continue with installation.

1. Main function is not required in TestNG. There are annotations in TestNG.
2. throw new SkipException("Reason")
3. Assert / Soft Assert
4. Data provider annotation returns the two dimensional object array.

Windows> Preferences>

**ANT:** Ant is build and compile tool.

https://ant.apache.org/bindownload.cgi

Download the latest zip file

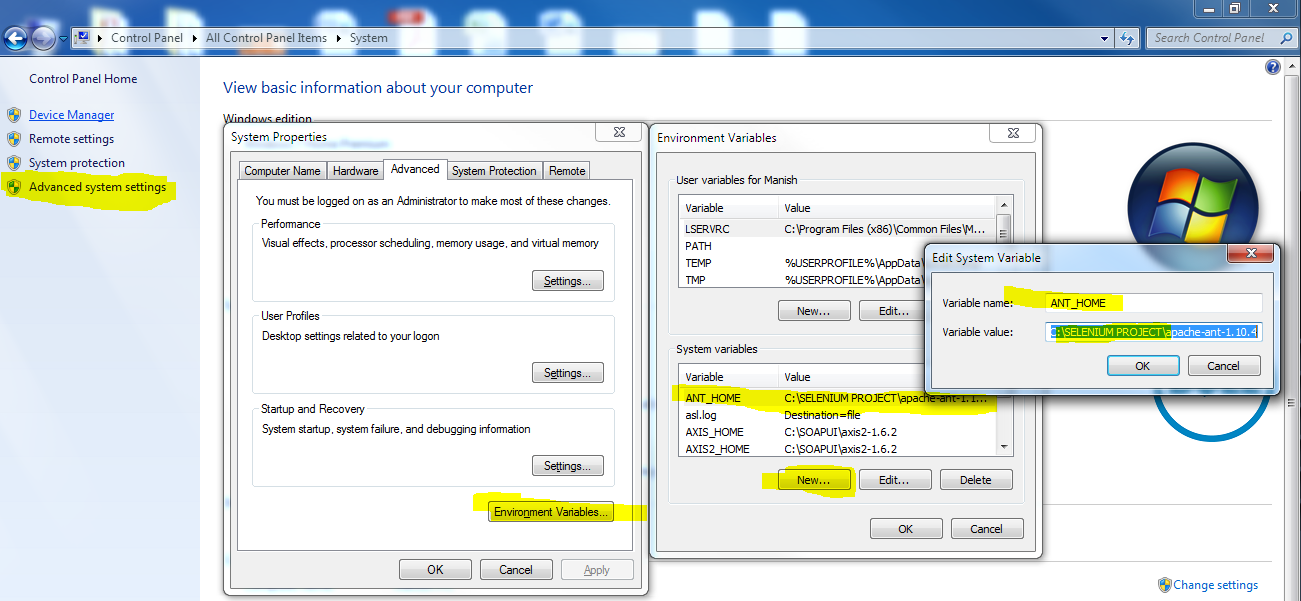
Extract the file.

Place the extracted folder to your project folder.

Copy the path of the folder

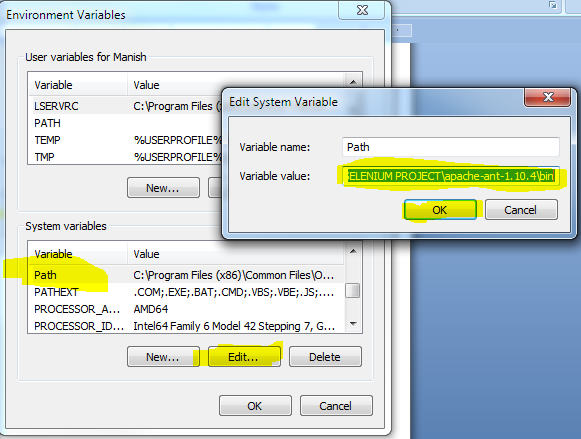
Go to Control Panel> System > Advance system settings> Environment Variables

Create an Environment Variable ANT\_HOME in System variables.

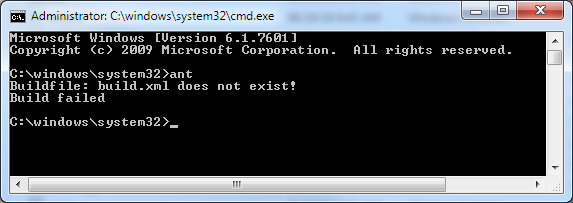


Scroll down to **Path** variable > Edit

Add a (**;**) at the end & put a path of bin folder of ant



Reopen the command prompt and run the command ant and if you get below message, it is successfully configured.



**NOTES:**

Build.xml is heart of ANT

ant compile

ant run

**Listeners:**

It looks at the code execution and if some event has to be fired after or before the code execution, that can be fired using Listener.

Prevent your code from stopping.

reporting multiple failures

**Maven with TestNG:**

Selenium creators gives the latest jar files to the community servers like (Apache). Apache deploys the jar files on their server and you can connect to that server to get the latest JAR files.

Basically All the open source distributors and manufacturers they can make the project and submit the jar files inside the MAVEN centralize repository and all the JAVA developers in this world they can get the required JAR files from that centralize repository by connecting to the server.

SVN and Maven both runs on a server.

SVN just used to store the JAVA code and MAVEN is used to store the JAR files.

Both are used to organize the stuff when working with a team.

Maven runs on community servers.(ex Apache, MVN etc.)

Maven is similar to SVN but not SVN.

Maven tool helps to download the JARs from Maven Repositories and configure it into eclipse project.

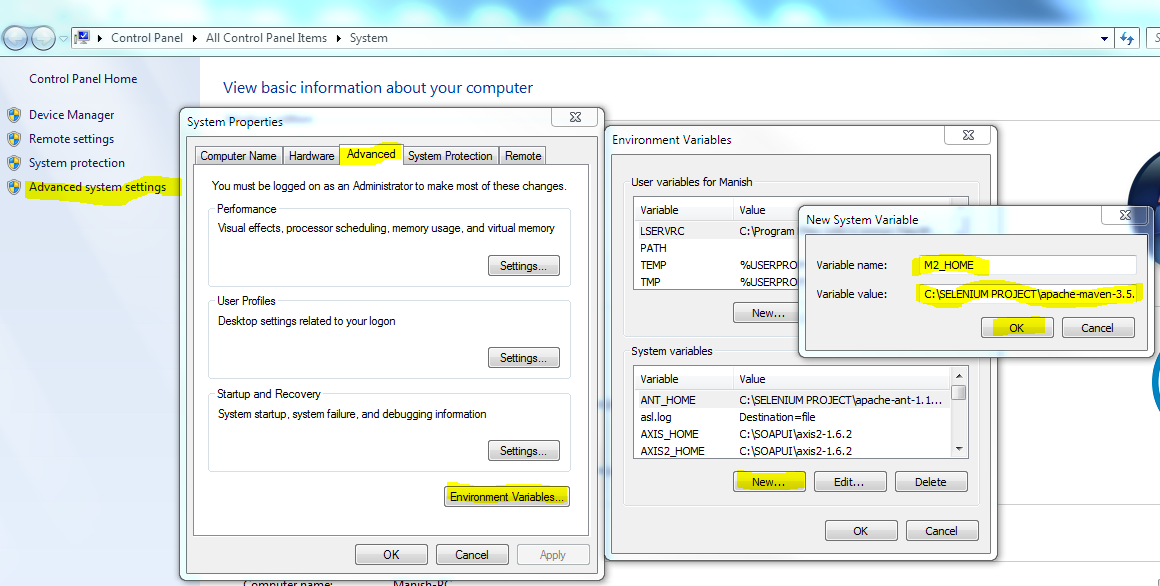
after reading POM.xml is like a document

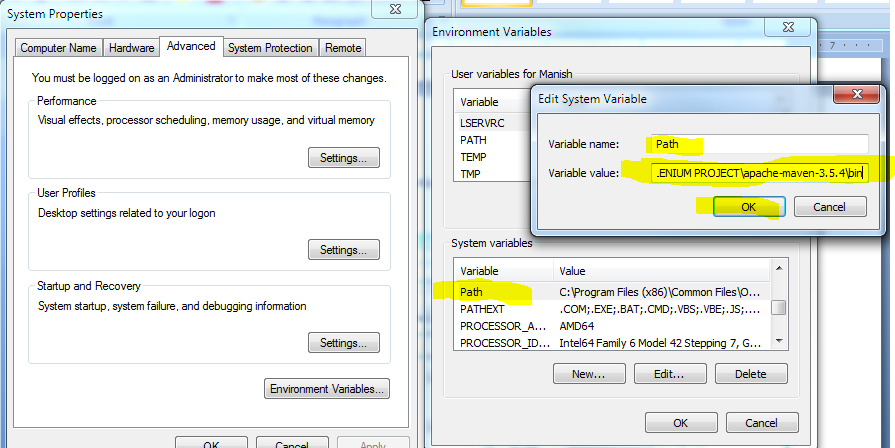
**Configuring MAVEN:**

Download MAVEN **.zip** file - https://maven.apache.org/download.cgi

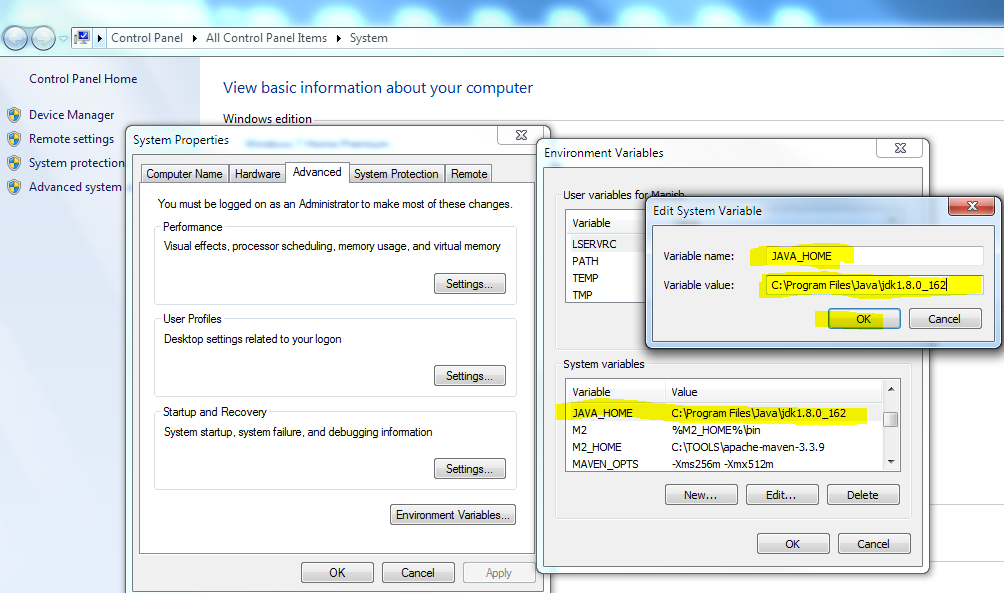
Extract and place wherever you want to.

Add environment variable **M2\_HOME**. - (MAVEN\_HOME)

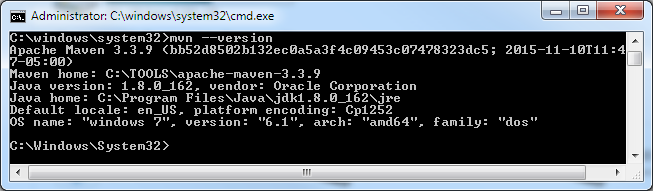


Add **bin** folder path

Add **JAVA\_HOME** variable and give the **path** of **JDK installation** if not already there.



Close all windows and open/reopen the command prompt and run the command **mvn --version**. If it returns like below, your MAVEN is configured successfully.



**mvn archetype:generate** - This command connects to the centralize repository of Apache and it will get all types of projects/Application which are associated with that repository and every project/Applications has a project/Application number & these are basically JAVA projects. these projects are known a arcchetypes.

groupId - packageName (Ex. com.testing)

artifactId - MyMavenProject - (Name of your Maven project)

Enter

Enter

Y

It will build the project

It will create the **MyMavenProject** at the dir from which you fired the command.

It will have the **pom.xml** which is the heart of the application. It specifies all the dependencies and JAR files on which your project is dependant.

You can import the maven project inside eclipse IDE. For that we have few commands.

Go to **MyMavenProject** dir then fire the below command

mvn eclipse:eclipse - This will make your project capable enough to be imported into the eclipse.

**.project** and **.classpath** files would be created. These files are required by eclipse to understand that this is a project.

Other:

mvnrepositories.com

Search for JUNIT/TestNG

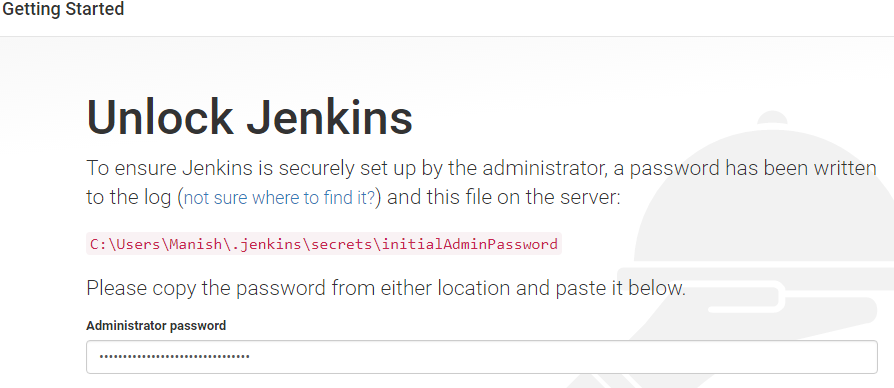
Refer to Module 37 - Part 2 for further instructions.

**TestNG dependencies:** http://testng.org/doc/maven.html

**MavenAntRun**

**Jenkins:**

1. Download **jenkins.war** file
2. Open command prompt and go to the path where the **jenkins.war** is placed
3. run command: **java -jar jenkins.war**
4. **In brwser enter url :** http://localhost:8080 & you will get below

****

Go to the path in screenshot and copy paste the password and hit continue.

