MAVEN

Build Management :

- Build Management refers to the process of converting raw source code into a distributable package after being tested and validated.
- Maven is one of Build Management tools
- Technically, Build refers to syntax check in source code.

Advantages of Build tool :

- Purpose – to build the code.
- Builds are done in SDLC to identify bugs at early stage of life cycle
- Eliminate human errors

Build tools :

- MSBuild
- Ant
- nAnt
- Maven
- SBT
- Gulp
- Gradle etc

Maven build tool :

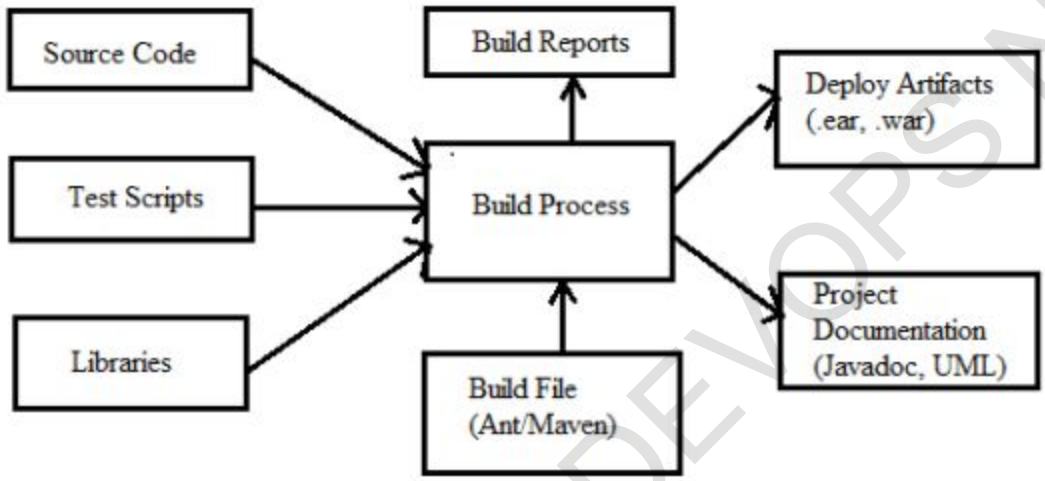
- Apache maven is an advanced project management tool for java software projects which is based on POM (project object model).
- It uses POM (project object model) file to manage project's build, dependency and documentation.
- The most powerful features of maven are to download the project dependency libraries automatically and to create the right project structure.
- Introduced in 2002 by Apache
- Open source
- Can build any java framework
- Important configuration file is : pom.xml

Features of Maven :

- Open source
- Generate document
- Generate reports

- Project management tool
- Follows POM Model
- Does builds and runs test cases

Architecture of maven :



Maven – Installation, Configuration, verification :

1. visit official site of Maven, web site : http://maven.apache.org

2. go to downloads, download binary zip - apache-maven-3.6.3-bin.zip

3. Unzip it to c:\programfiles\

4. Need to create Configure environment variables

   - go to system properties, go to advanced tab,

   - Under system variables. Please create below -

   1. MAVEN_HOME C:\Program Files\apache-maven-3.6.3\bin

   2. M2_HOME C:\Program Files\apache-maven-3.6.3\bin

6. Then open command prompt, and verify using below command

mvn --version

Follow below steps to configure Maven in Jenkins

1. Open Jenkins

2. open Manage Jenkins --> open " Global Tool configuration"

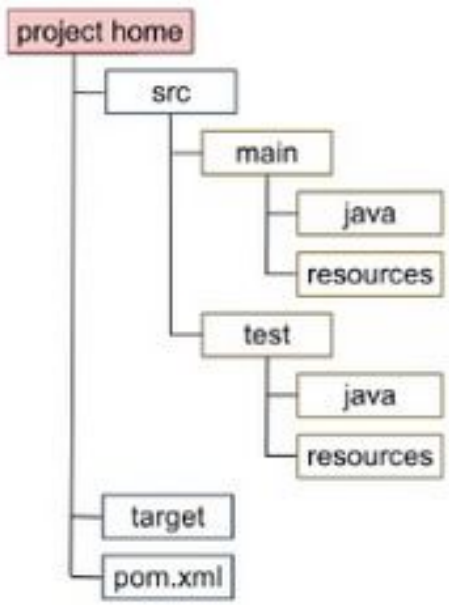
- need to configure maven

MAVEN_HOME

Maven Life-Cycle Phases :

Phase	Description
validate	Validate the project is correct and all necessary information is available.
compile	It compiles the source code of the project.
Test	Tests the compiled source code using a suitable testing framework.
package	This phase take the compiled code and creates the JAR/WAR package as mentioned in the packaging in POM.xml.
install	This phase installs the package in local maven repository.
Deploy	This phase copies the final package to the remote repository.

Maven Project Structure :



Maven repositories:

- Maven repositories are directories of packaged JAR files with extra meta-data.
- The meta-data is represented by POM files.
- A repository contains all the project jars, library jar, plugins and any other project specific artifacts.

Types of maven repository:

1. **Local Repository**
2. **Central Repository**
3. **Remote Repository**

1. Maven Local Repository:

- Maven local repository is a directory on the developer's machine.
- It gets created when we run any maven command for the first time.

- It contains all the dependencies (downloaded by maven) like library jars, plugin jars etc.
- Default location of maven local repository is user-home/.m2 directory.
- We can change the default location of maven local repository by changing the settings.xml file. It is located in MAVEN_HOME/conf/settings.xml.

```
<settings>
   <localRepository>
      //Set desired location
   </localRepository>
</settings>
```

2. Maven Central Repository:

- Maven central repository is created by the apache maven community itself.
- It contains a lot of commonly used libraries.
- By default Maven looks in this central repository for any dependencies needed but not found in your local repository.

Maven central repository path: http://repo1.maven.org/maven2/.

3. Maven Remote Repository:

- Maven remote repository is a repository on a web server.
- A remote repository can be located anywhere on the internet or inside a local network.
- We can configure a remote repository in the POM file.
- We have to put the following XML elements right after the element:

```
<repositories>
  <repository>
     <id>codesjava.code</id>
     <url>https://maven.codesjava.com/maven2/lib</url>
  </repository>
</repositories>
```

SAMPLE PROJECT TO BUILD USING MAVEN :

1. Open Jenkins

2. click on new item

3. give name to project and select Freestyle option

4. We have to now configure the project

   - code repo link

     https://github.com/jleetutorial/maven-project.git

   - setup scm

     select git, put URL of Repo, enter credentials, select branch

   - setup build trigger

   - choose the build

     "Invoke top-level Maven Targets",

     select the maven version,

     enter goals

5. finally save configurations

How to check results of build :

Method 1 : Using build history from dashboard menu

- it shows the data of build history, select the appropriate project and click on console output

Method 2: open the workspace of the project, select the build number from build history. and click on console output