**Maven**

**What is maven?**

It is a build tool, developed by Apache company.

It is an open source.

**What is build?**

Build is a version of a software the development team hands over to the testing team for testing purposes

Build is the process of creating the application program for a software release, by fetching the code from the source control repository. Compile the code and then creating a build artefact, such as binaries or executable program, etc.

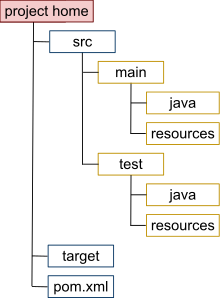
**Advantages of Maven**

* Maven can add all the dependencies required for project building, testing, and running code automatically by reading pom file.
* Using maven we can easily build the project to jar, war package etc.
* Maven downloads the required file automatically from the repository
* Using Maven, one can manage the entire life cycle of a test project
* Create reports
* Create documents

**Understanding the pom.xml file.**

* POM stands for project object model
* it will create automatically when we create a maven project. It manages all dependencies & plugins needed during the build lifecycle of a Maven project
* We can add our required dependencies (third-party libraries) and plug-ins(configurations) in this xml file.

**Maven project structure:**



This directory serves as the root of every Maven project.

Let's take a closer look at the standard files and subdirectories that are typically found at root:

* maven-project/src/main/java – Java source code for the artifact
* maven-project/src/main/resources – configuration files and others such as i18n files, per-environment configuration files, and XML configurations
* maven-project/src/test/java – Java source code for tests
* maven-project/src/test/resources – configuration files and others used by tests
* maven-project/target – The target folder is the maven default output folder. When a project is built or packaged, all the content of the sources, resources and web files will be put inside this folder.
* maven-project/pom.xml – defines dependencies and plugins needed during the build lifecycle of a Maven project
* maven-project/LICENSE.txt – licensing information of the project
* maven-project/README.txt – summary of the project
* maven-project/NOTICE.txt – information about third-party libraries used in the project.

**Maven Build Lifecycle:**

It consists of the following sequence of phases.

|  |  |  |
| --- | --- | --- |
| Phase | Handles | Description |
| prepare-resources | resource copying | Resource copying can be customized in this phase. |
| validate | Validating the information | Validates if the project is correct and if all necessary information is available. |
| compile | compilation | Source code compilation is done in this phase. |
| Test | Testing | Tests the compiled source code suitable for testing framework. |
| package | packaging | This phase creates the JAR/WAR package as mentioned in the packaging in POM.xml. |
| Install | installation | This phase installs the package in local/remote maven repository. |
| Deploy | Deploying | Copies the final package to the remote repository. |

**Maven Architecture:**

Diagram

Description automatically generated

* Create maven project
* Integrate selenium with testng, maven project

To integrate testng with maven we need to add maven surefire plugin in pom.xml file i.e.

Graphical user interface, text, application

Description automatically generated

* Run selenium test in eclipse using maven
* Maven reports
* Run maven project from command prompt

To run maven project from command prompt we need to install maven

**Maven Installation:**

Follow the below steps to download and set up Maven in the system:

1. Make sure that the system has Java installed (at least JDK 1.7) and JDK & JAVA\_HOME environment variables configured
2. Go to the [Maven Apache Download](https://maven.apache.org/download.cgi) page and [download the binary zip file](https://mirrors.estointernet.in/apache/maven/maven-3/3.6.1/binaries/apache-maven-3.6.1-bin.zip). Unzip it to the C drive, and the installation is complete
3. Configure the MAVEN\_HOME path as Environment Variable and give path till the bin folder
4. Open the Control panel and go to the System link
5. Click on the Advanced system settings tab at the left slide and navigate to the Advanced tab
6. Click on the environment variables and add the MAVEN\_HOME variable in system variables and give the path of the Maven bin folder and save it.
7. To set the Environmental Variable – Go to Start menu > Edit Environment Variables > System Properties > Environment Variables > In the System Variables set the MAVEN\_HOME path

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

1. Verify the installation by executing below command :

**mvn -version**

* Maven commands

1. mvn clean
2. mvn validate
3. mvn compile
4. mvn test
5. mvn package
6. mvn install
7. mvn deploy
8. mvn clean test install