Maven

Build Tools/Build Automation/build management tool.

It will collect the source code from git and compile and test the code and finally generate the package.

Same like maven tools:(build automation tools).

Grunt,

Gulp,

Gradle,

Ant,

Maven.

Diff b/w ant and maven:

Ant actions are defined need to write more script.

Sequence are define in ant

No default directory layout.

Ant follows you.

Compiling source code.

To run maven we need java

To install java go through the below url.

<https://www.oracle.com/java/technologies/javase/javase-jdk8-downloads.html>

download the macos 254 + mb

install on mac or windows.

Now we have to set java on environment variable.

In mac

Cd ~/

Type ls –al

If u see .bash\_profile means u won’t need to create a any file .otherwise touch .bash\_profile create it .

If u want open a hide files

Open –e .bash\_profile.

Give below command

export JAVA\_HOME=$(/usr/libexec/java\_home)

save and close it.

After u type below command means

echo $JAVA\_HOME

now it will show the total path of java install path.

Then download maven

Open u r terminal and u will type below commands means u check maven

mvn –version

go to this url : <https://maven.apache.org/download.cgi>

for mac:

download this file Binary tar.gz archive

after unzip in download

we have 2 move that folder from downloads to appications

mv apache-maven-3.6.3 /Applications/

now that file will move to application

cd /Applications/

cd apache-maven-3.5.4/

cd /Users/mac here mac is the username of user.

Now type ls –l

Open .bash\_profile

Open –e .bash\_profile

Type below command

Open –e .bash\_profile

Paste below code

export M2\_HOME=/Applications/apache-maven-3.6.3

export PATH=$PATH:$M2\_HOME/bin

note:here 3.6.3 is the maven version.

To reload or refresh a file

Source .bash\_profile

Now finally if u type mvn –version

**Apache Maven 3.6.3 (cecedd343002696d0abb50b32b541b8a6ba2883f)**

Maven home: /Applications/apache-maven-3.6.3

Java version: 1.8.0\_241, vendor: Oracle Corporation, runtime: /Library/Java/JavaVirtualMachines/jdk1.8.0\_241.jdk/Contents/Home/jre

Default locale: en\_US, platform encoding: UTF-8

OS name: "mac os x", version: "10.13.6", arch: "x86\_64", family: "mac"

If u see above code means maven successfully install on mac.

Maven variables

Local variables & system variables

System variables are (predefined).

Local variables are define by user.

Global variables access at anywhere.

How maven works:

It works on goals as plugin or jars files which has the future of when and what to do,

Ex:maven do test 🡪 then it call plugin to do testing.

Default life cycle or dependent lifecycle:

1.generate-source files.

2.compile all java files .into .class files

3.testing.(unit test a piece of code).

4.package

5.integration-test

6.install

--clean : it delete run time files.

--site :documentation(99% we will not use,very rare)

ex:mvn compile package.

Steps we cover 1,2,3,4

Note:source code we can customize.

Binary which can we buy directly.

Maven layout:

Standard direct layout:

Src->

main->java realated files.(source code will be there).

test.->test related files.

Gav:

How maven which plugin to use.select when we give a goal

G (groupid) – string name or company name

A(artifactid) –final output of u r product

V(versioned) –major or minor or patch .

Packages:

Jar -- java archive(default package).

War – web archieve – contain group of jar + config + xml(for web based projects)

Ear – enterprise application.

Real time start.

Go to git directory in compile and create a new folder maven using mkdir

Mkdir maven

To start maven we need one pom.xml and src entire arechetecture what we discuss in previously.

To start run below command

mvn archetype:generate

after typing this command it will start to download the files.

Interactive mode and next enter and next enter take default one version next it will ask

group id 🡪some name

artifactid->some id like demo  
packge -> press enter after accept y.

finally we will see build success message.

If we get that message means maven successfully installed on that project.

After we type ls meand we will see demo folder means we given artifact id.

Now we will go to inside of demo folder we will see pom.xml and src

If pom.xml and src there we will start maven.

Go to the exact location of pom.xml and src

Commands will start from compile.

mvn compile

it will create a target folder inside target folder all java files will converted into .class files.

Next

mvn test

again it will start from compile if that plugin not there in local it will go to online maven server .

next

package

the related output project files will store diff packages.

mvn package

again dependency will start from

compile and after test and after go to package.

Package also it will show finally success.

Package will create a zar or zip files it will store on targets here all java files will be available in class formats.

Clean:

mvn clean

will clear all the runtime data that will present on target folder.

Finally target folder also gone.

--------------------------------------------------------------------------

POM (project object model).

It’s a fundamental work of maven ,pom is xml file,it contain information about the project and configuration details used by maven to build the project.

Atleast one pom file will be present in maven

Describe a project

Name and versions.archetecture type ,source code locations,dependencies.

Plugins

Profiles(alternative build configuration)

It uses xml by default.

How maven identify the plugin?

When 2 run the plugins ?

What is the use of plugins?

What exactly we doing?

<project>

<build>

<plugins>

<plugin>

<groupId>maventrain</groupId>

<artifactId>demo</artifactId>

<version>1.0-SNAPSHOT</version>

<executions>

<execution>

<phase>clean</phase>

<goals>

<goal>run</goal>

</goals>

<configurations>

<tasks>

<echo>Helloworld</echo>

</tasks>

</configurations>

</execution>

</executions>

</plugin>

</plugins>

</build>

</project>

1.mvn pluginame <goal name>

2.mvn compile

3.ex:mvn antrun:run

above code is background original code.

Go to this below url

<https://maven.apache.org/>

under website plugins category we will see this one.

<https://maven.apache.org/plugins/index.html>

under compiler and go to usage page.

<https://maven.apache.org/plugins/maven-compiler-plugin/usage.html>

example take plugin page antrun.

<https://maven.apache.org/plugins/maven-antrun-plugin/usage.html>

copy ant gav:

<artifactId>maven-antrun-plugin</artifactId>

<version>1.8</version>

and paste below plugin

<?xml version="1.0" encoding="UTF-8"?>

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>maventrain</groupId>

<artifactId>demo</artifactId>

<version>1.0-SNAPSHOT</version>

<name>demo</name>

<!-- FIXME change it to the project's website -->

<url>http://www.example.com</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.7</maven.compiler.source>

<maven.compiler.target>1.7</maven.compiler.target>

</properties>

<build>

<plugins>

<plugin>

<artifactId>maven-antrun-plugin</artifactId>

<version>1.8</version>

<executions>

<execution>

<id>id.claen</id>

<phase>clean</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<tasks>

<echo>Helloworld</echo>

</tasks>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

and run the

note : under plugins plugin we need to give correct artifact id and version id from below url.

<https://maven.apache.org/plugins/maven-antrun-plugin/usage.html>

otherwise it will through an error.

mvn clean

now it will show build success.

It will print echo world.

Now run u r goal

<https://maven.apache.org/plugins/maven-antrun-plugin/>

mvn antrun:run

it will show build success.

Above way we can run specific plugin.

Now we are going exec plugin

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>exec-maven-plugin</artifactId>

<version>1.6.0</version>

<configuration>

<executable>maven</executable>

<arguments>

<argument>-X</argument>

<argument>myproject:dist</argument>

</arguments>

</configuration>

</plugin>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>exec-maven-plugin</artifactId>

<version>1.6.0</version>

<executions>

<execution>

<id>id.exec</id>

<phase>exec</phase>

<goals>

<goal>exec</goal>

</goals>

<configuration>

<tasks>

<echo>Helloworld exec</echo>

</tasks>

</configuration>

</execution>

</executions>

</plugin>

to execute this plugin.

mvn exec:exec

plugins concept:

to run particular plugin means we can go to profile concept.

Above build create profiles and create profile id below code like this

<profiles>

<profile>

<id>cleanprofile</id>

<build>

<plugins>

<plugin>

<artifactId>maven-antrun-plugin</artifactId>

<version>1.8</version>

<executions>

<execution>

<id>id.claen</id>

<phase>clean</phase>

<goals>

<goal>run</goal>

</goals>

<configuration>

<tasks>

<echo>Helloworld</echo>

</tasks>

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

</profile>

</profiles>

now type

mvn clean

means above run all code will execute.

To run specific plugin profile.

mvn –Pprofileid clean

note p should be in capital.

Now it will show build success.

Snapshot in maven:

<version>1.0-SNAPSHOT</version>

it will explain the product will be in developing mode or production mode.

If the version it will be there in 1.0 snapshot means its under development only.it s standard rule.

When it go to production it will change.

Multi Module & dependeces

If a maven project need to run means we need src and pem file.

So a project will divided multiple submodules so every module need a src and pom file.

Example take some sub modules:

Go to maven folder

Create some folder above src folder.

Create 2 new folders and paste src folder and pom file in 2 folders.

In pom file we can go for default we don’t need any plugin code now

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>maventrain</groupId>

<artifactId>demo</artifactId>

<version>1.0-SNAPSHOT</version>

<name>demo</name>

<!-- FIXME change it to the project's website -->

<url>http://www.example.com</url>

<properties>

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

<maven.compiler.source>1.7</maven.compiler.source>

<maven.compiler.target>1.7</maven.compiler.target>

</properties>

</profiles>

<dependencies>

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.11</version>

<scope>test</scope>

</dependency>

</dependencies>

</project>

goto folder 1 and change give same name to artifactid and name like ex:folder1.

Now cd go to folder1 and simply install maven packages (build+compile) all willdone in one command that is.

mvn package.

Now it will show build success message.and it will create a target folder

After success message do the same operation to the folder 2 also until show success message.

So above procedure 4 manually creating a submodules in a project.

Now we are creating modules automatically using pom file

Now go to parent maven directory.

Type mvn clean package

If u type this command means it will through an error.

To solve this one we need to mention what is parent pom file and what is the chilf 4 this one.

Go to parent pom file.

And add package name as the.

<packaging>pom</packaging>

for above code it will conside as the parent pom file.

Now it will come build success message.

In maven when we have plugins it automatically makes dependencies.

Now go to parent pom file and change the dependencies.

From modules under plugins

Make reverse

Test2

Test1

Previoulsly we have

Test1

Test2

Save it and run this command

Mvn clean package

After build success

It will show test2 and test1.

Now we have to give dependency means go to file2 pom file and create a dependence

<dependency>

<groupId>maventrain</groupId>

<artifactId>test1</artifactId>

<version>1.0-SNAPSHOT</version>

</dependency>

add dependency in dependencies.

In file2 pom page dependencies.

If u check mvn package means it will show error

Bcoz that test1 and test2 are the local plugins we need to tell the maven about our local plugins

Now we are including test1 gar file in test2

So go to cd test1

And type mvn install

Show build success message

Means it will load the data.

Now go to folders

Give mvn install

Go to parent type cd..

Mvn package

After build success we can see dependencies like

Test1,test2,demo

Bcoz we did local dependency concept.

If in real time we have a lot of modules not only two that time we have problem 4 this one we have a solution in maven.

Above parent and child realtion is partially.

Now we r going fully.

So maven take all modules automatically dependencies.

Go to parent pom file and copy the parent pom file gar and copy that paste in all local files.

In parent

<parent>

<groupId>maventrain</groupId>

<artifactId>demo</artifactId>

<version>1.0-SNAPSHOT</version>

</parent>

now type mvn clean package

now it will come in correct order like

demo

test1

test2.

From now maven is completed.