s

### Maven

is a build managment tool provided by apache

### Advantages of maven

It provides project structure

Provides pom.xml in which we can define dependencies and plugins(configurations)

Generate reports (in target folder🡪 index.html)

Documentation

Packaging ( convert into jar files)

### Maven build life cycle

Validate : validates the dependancies

Compile : compile the code

Test : performs validate, compiles and tests

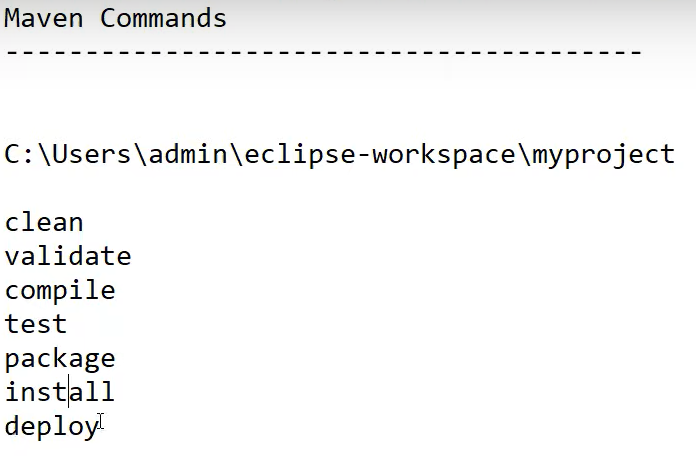
Package :

Integration test

Verify

Install : install in local repo

Deploy : install in remote repo to use for other project



### Maven commands

mvn clean

mvn validate

mvn compile

mvn test

mvn package

mvn install

mvn deploy

### Passing multiple commands

mvn clean test : cleans the project and runs the test

<https://www.javatpoint.com/maven-interview-questions> --start

Maven Interview Question

A list of top frequently asked **maven interview questions** and answers are given below.

1) What is Maven?

Maven is a project/build management tool. It is based on POM (Project Object Model). [More details.](https://www.javatpoint.com/maven-tutorial)

2) What aspects are managed by Maven?

Builds

Documentation

Reporting

SCMs

Releases

Distribution

[More details.](https://www.javatpoint.com/maven-tutorial)

3) What are the advantages of Maven?

No need to add jar file in each project

Creates right directory structure

Builds and deploys the project

AD

[More details.](https://www.javatpoint.com/maven-tutorial)

4) What is the command to check the maven version?

Type the following command on console to know the maven version.

mvn -version

[More details.](https://www.javatpoint.com/how-to-install-maven)

5) What does the build tool?

Generates source code (if the auto-generated code is used)

Generates documentation from source code

Compiles source code

Packages compiled code into a JAR or ZIP file

Installs the packaged code in the local repository, server repository, or central repository

[More details.](https://www.javatpoint.com/maven-tutorial)

6) What is the difference between Ant and Maven?

|  |  |
| --- | --- |
| **Ant** | **Maven** |
| It is **a toolbox**. | It is **a framework**. |
| It is **mainly a build tool**. | It is **mainly a project management tool**. |
| There is **no life cycle**. | There is a **life cycle**. |
| Ant **doesn't have formal conventions**. | Maven **has a convention** to place source code, compiled code etc. |
| Ant is **procedural**. | Maven is **declarative**. |
| The ant scripts are **not reusable**. | The Maven plugins are **reusable**. |
|  |  |

[More details.](https://www.javatpoint.com/difference-between-ant-and-maven)

7) Why is the use of the profile required in Maven?

For providing probability to projects, we use profiles.

8) What is the syntax for offline project creation?

The syntax for project creation is:

mvn o packg.

9) How is the propagation of plugins to child POMs stopped?

It can be done using the following syntax:

set**<inherited>** to false.

10) What is the use of the exclusion element?

The element is used to exclude dependencies.

11) Define SNAPSHOT in terms of maven.

The snapshot indicates the current development copy.

12) Define Archetype.

It is a Maven plugin which is designed for the creation of project structure.

7. What does central repository consist of?

It consists of a large number of libraries that are frequently used.

13) Give the command for installation of the JAR file in a local repository.

mvn install

14) Mention the phases of cleaning lifecycle.

The lifecycle of cleaning consist of:

pre-clean

clean

post-clean

15) What is the purpose of mvn clean command?

The command removes the target directory before the starting of a build process.

16) What is a MOJO?

A MOJO stands for Maven plain Old Java Object. Each MOJO is an executable goal in Maven, and a plugin is a distribution of one or more related MOJOs.

17) What is a repository?

A repository is a directory or place where all the jars and pom.xml file are stored. There are 3 types of a repository in Maven:

Local Repository

Central Repository

Remote Repository

[More details.](https://www.javatpoint.com/maven-repository)

18) What is a local repository?

Maven local repository is created by maven in your local system when you run any maven command. [More details.](https://www.javatpoint.com/maven-repository#local)

19) What is a central repository?

Maven community creates maven central repository on the web. [More details.](https://www.javatpoint.com/maven-repository#central)

20) What is a remote repository?

Maven remote repository is located on the web by different vendors. So you need to define the dependency in pom.xml file manually. It is important because most of the libraries are missing from the central repository. [More details.](https://www.javatpoint.com/maven-repository#remote)

21) What is POM?

POM stands for Project Object Model. The pom.xml file contains information of project and project configuration. [More details.](https://www.javatpoint.com/maven-pom-xml)

22) What are the build phases in Maven?

validate

compile

test

package

integration-test

verify

install

deploy

23) What is the command to package maven project?

mvn -package

24) What is the fully qualified artifact name of maven project?

**<groupId>**:**<artifactId>**:**<version>**

25) What is an archetype?

Archetype is the maven plugin. It creates the project structure.

<https://www.javatpoint.com/maven-interview-questions> --end

================================================================================

<https://www.softwaretestinghelp.com/maven-interview-questions/> --start

**Q #1) What do you mean by Maven?**

**Answer**: Maven is a project management tool (introduced by Apache Software Foundation) that provides an entire framework for the build cycle. It is open-source and is mainly used for the project developed in Java.

Maven is driven by a project object model popularly known as POM. It is the central repository for all the dependencies. It maintains the same folder convention across organizations and can be easily integrated with continuous integration tools like Jenkins.

Besides, it takes care of the reporting and documentation of the project. It is developed in Java language.

**Q #2) What are the features or advantages of Maven?**

**Answer**:

**The features or advantages of Maven are as follows:**

Not required to manually add jars for the project. The updates to the project dependencies and transitive dependencies are carried out automatically by Maven.

Maven maintains a uniform directory structure across the organization.

Both the deployment and build activities are taken care by Maven.

Maven is simple, easy to grasp, to set up and utilize in the projects as it is driven by the POM file.

Rapidly expanding repository of Maven contains a large number of libraries that can be used for multiple projects at a time.

Maven encourages the use of extensible code design.

**Q #3) Name the aspects that are managed by Maven.**

**Answer**: The aspects that are managed by Maven are documentation, builds, SCMs, releases, distribution, mailing lists, dependencies, and reporting.

**Q #4) What is meant by Maven build lifecycle? Name these cycles.**

**Answer**: A Maven build life cycle comprises of certain phases. It determines the order in which the Maven goals need to be performed. If a particular phase is to be accomplished, then all the previous phases prior to that particular phase need to be run successfully in sequence.

**Maven build life cycles are listed below as the followings:**

**Clean**: Removes the artifact produced from the preceding build processes.

**Default**: Deployment and build process is taken care of.

**Site**: Project documentation is carried out in this cycle.

**Q #5) What is a build tool responsible for?**

**Answer**: Build tools are primarily responsible for producing the source code (provided the auto-generated code is utilized). It creates project documentation from the source code.

It compiles and packages the code in the form of JAR/ZIP file. Finally, places that code in the local, central, or remote repository.

**Q #6) What do you mean by POM and what does it contain?**

**Answer:**Project Object Model or pom forms the elementary part of the working of Maven. While performing a Maven goal or a task, the pom file residing in the present directory is searched and referred to collect information on the project.

It is basically an xml file. It contains configuration details like plugins, goals, developers, dependencies, profiles, versions, and mailing lists.

**For example**, to add excel dependencies in Selenium, we need to add the dependencies pasted below.

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2020/05/pom.png)

**Q #7)** **What do you mean by a Maven Repository?**

**Answer:**Maven repository is the location of a directory where all the related project artifacts, jars, libraries, plugins are kept and can be utilized by Maven easily.

**Q #8) Explain the various types of Maven Repositories.**

**Answer: There are three types of Maven repositories.**

**They are:**

**Local Repository:**This is placed in our local machine generally in the **.m2** directory. This is generated once we are able to execute a Maven command successfully. All the project dependencies reside here. Once Maven scans the pom file, it first searches for its dependencies in the local repository.

**Central Repository:**This is supplied by Apache Maven. It contains most of the routinely used libraries. Once any of the dependencies are not found in the local repository, then Maven connects to a central repository.

**Remote Repository:**Sometimes companies develop their own custom repository comprising of their project artifacts, jars, and libraries. This is a type of repository that remains private for use inside that organization.

**Q #9) Explain Maven Artifact.**

**Answer:**Maven artifact is known as a file (generally a jar) that is extended to the Maven repository. Maven build generates multiple artifacts comprising of source jar and compiled jar. **GroupId, artifactId, and version** together form an artifact and all of the three combined solely identifies it.

**The below code snippet shows an example of the Maven artifact.**

<dependency>

<groupId> org.selenium </groupId>

<artifactId> seleniumJava </artifactId>

<version> 4.0.2 </version>

</dependency>

**Q #10) List down the various scopes of Maven Dependency.**

**Answer:**

**The various scopes of Maven dependency include:**

**Compile**: This scope is required to build, test, and run the project and is available by default.

**Provided:**This scope is needed to build and test the project and is available at runtime.

**Runtime:**This scope is not required for compilation but needed for execution.

**Test:**This scope is needed for compilation and running of the unit test cases.

**System:**This scope cannot be taken from the remote repository and need to be placed in the local project path.

**Import:**This scope is utilized when the dependencies are of pom type.

**Q #11) What are the differences between Maven and Ant?**

**Answer: The differences between Maven and Ant are listed below:**

| **Maven** | **Ant** |
| --- | --- |
| Maven is descriptive and most of the project information are defined in the pom file. | Ant is in form of procedure or method where all the instructions to be performed are given in order. |
| Maven follows a life cycle. | Ant does not follow any life cycle. |
| Maven is considered as a framework. | Ant is considered as a tool box. |
| Maven is a project management and build tool. | Ant is a tool used only for build process. |
| Maven plugins can be reused. | Ant scripts cannot be reused. |
| Maven follows a protocol for example, Maven projects follow a uniform directory structure across organization. | Ant does not follow any specific protocol. |

**Q #12) What do you mean by Maven Archetype?**

**Answer:**Maven archetype is basically a plugin assigned with the job of generating the project structure according to a particular skeleton or template.

**AD**

**The following command is used to create a project template:**

**mvn archetype: generate**

**Q #13) How to determine the version of Maven in our system?**

**Answer:**To determine the version of Maven we are using in our system, we need to enter the below command in the console.

**mvn –version**

**Q #14) What is known as SNAPSHOT in Maven?**

**Answer:**Maven snapshot is that version which is still unreleased. Prior to the first release is completed, there is a 1.0-SNAPSHOT version. In the future, this version will emerge to be 1.0 version. It is called as the present development copy. Maven looks for the new SNAPSHOT version in the remote repository.

**Q #15) How to mention profiles in Maven?**

**Answer:**The profiles are mentioned in Maven with the help of a subset of elements present in the pom file.

**Q #16)** **What are Maven Plugins?**

**Answer:**Maven plugins are the basic component of a Maven framework. Each of the plugins has a specific task to be performed.

**Maven generally performs the following functionalities:**

Generates jar files.

Generates war files.

Compiles the code.

Executes unit testing of code.

Generates documentation of the project.

Generates customized reports.

**Maven plugin gives a group of goals that can be run with the following command syntax:**

**mvn [plugin-name]:[goal-name]**

**Q #17) What are the different types of Maven Plugins?**

**Answer:**

**The different types of Maven plugins are listed below:**

**Building Plugins:**These plugins are used at the time of build and are defined in the building element of the pom file.

**Reporting Plugins:**These plugins are used at the time of site generation and are defined in the reporting element of the pom file.

**Q #18) What does goal in Maven mean?**

**Answer:**A collection of Maven goals constitute a phase. Each goal is defined for a particular job involved in project management and execution of the build.

**Q #19) Name the build phases in Maven Build Lifecycle.**

**Answer:**The build phases in Maven Build Lifecycle are listed down:

**Validate:**Checks if all the preconditions information to trigger the build is obtained.

**Compile:**Project source code is compiled.

**Test:**The Source code that is compiled is tested with the unit test framework. In this phase, the code is not deployed or packaged.

**Package:**Source code after compilation is packaged in the form of ZIP or JAR files.

**Integration- test:**After the package is deployed in an environment, the integration test cases are executed.

**Verify:**Examines to ensure that the package is correct and it meets all the required quality specifications.

**Install:**Installation of packages into the local repository.

**Deploy:**A specimen of the final package is made accessible to the remote repository for distribution among the other developers across projects.

**Q #20) What is the location where Maven dependencies are downloaded?**

**Answer:**The project artifacts, dependencies, and jars downloaded by Maven are placed in the local repository of Maven. The folder **.m2**is by default the location for the local repository. This default location can be changed from the settings.xml file.

**Q #21) Name the phases of the Maven Clean Lifecycle.**

**Answer:**

**The phases of Maven Clean Lifecycle are listed below:**

pre-clean

clean

post-clean

**Q #22) What is the purpose of command mvn clean in Maven?**

**Answer:**mvn clean aims to clean the project artifacts created by the previous Maven builds from the target directories. This is generally executed before initiating a new build process.

**Q #23) What do you mean by Build profiles in Maven?**

**Answer:**Build profile is a collection of configurations that helps to place a value or overrule the default values required for the build process in Maven.

We can design the build process for various environments like development, testing, and production. Profiles are described in the pom files with the help of <profiles> and is usually used to point to different environments.

**Q #24) List down the types of Maven Build profiles.**

**Answer:**

**The types of Maven Build profiles are listed below:**

**Per-User:** This is described in Maven settings.xml file.

**Per Project:** This is described in pom.xml of the project.

**Global**: This is described in the global Maven settings.xml file.

**Q #25) What do you mean by Maven’s External Dependencies?**

**Answer:**Once Maven reads through the pom file, it gets to know the list of dependencies for the project. It searches for these dependencies in Local, Central and Remote repositories. In case any of the dependency not found in any of the repositories, then Maven utilizes the external dependency.

**Q #26) What are the parameters required to define External Dependency in Maven?**

**Answer:**Maven external dependencies are described in the pom xml file just like other dependencies. The parameters required to define the external dependencies are **groupid, artifactId, scope set to system and system path as per the project location** (relative path).

**Q #27) What is meant by Mojo in the terminology of Maven?**

**Answer:**Mojo is referred to as **Maven plain Old Java Object**. A mojo is basically an executable goal to increase the features of Maven. A plugin is a distribution of one or multiple connected MOJOs.

**Q #28) What is the job of the exclusion element in Maven?**

**Answer:**In order to keep out any of the dependencies, the exclusion element is used in Maven.

**Q #29) Why is it recommended to keep the external dependencies in the local repository rather than a remote repository in Maven?**

**Answer:**It is recommended to keep the external dependencies in the local repository instead of remote in Maven because local repository consumes less space, are easily accessible and there is no need to take care of versioning for jars.

**Q #30) How to perform a force update in Maven?**

**Answer:**A forced update in Maven can be done with the following commands:

**mvn clean install –U**

**-U** is used for forcing a Maven update. However, the dependencies that are release based cannot be performed in this manner.

**Q #31) Explain the error “You cannot have two plugin executions with the same ( or missing) elements” in Maven.**

**Answer:**This error message comes in the situations where we have run a single plugin more than one time with the same id. We need to give a unique id for each execution.

<https://www.softwaretestinghelp.com/maven-interview-questions/> -- end

==========================================================================

--start

<https://www.guru99.com/maven-interview-questions.html>

1) Explain what is Maven? How does it work?

Maven is a project management tool. It provides the developer a complete build lifecycle framework. On executing Maven commands, it will look for POM file in Maven; it will run the command on the resources described in the POM.

2) List out what are the aspects does Maven Manages?

Maven handles following activities of a developer

Build

Documentation

Reporting

Dependencies

SCMs

Releases

Distribution

Mailing list

3) Mention the three build lifecycle of Maven?

**Clean:** Cleans up artifacts that are created by prior builds

**Default (build):** Used to create the application

**Site:** For the project generates site documentation

4) Explain what is POM?

In [Maven](https://www.guru99.com/maven-tutorial.html), POM (Project Object Model) is the fundamental unit of work. It is an XML file which holds the information about the project and configuration details used to build a project by Maven.

5) Explain what is Maven artifact?

Usually an artifact is a JAR file which gets arrayed to a Maven repository. One or more artifacts a maven build produces such as compiled JAR and a sources JAR.

Each artifact includes a group ID, an artifact ID and a version string.

6) Explain what is Maven Repository? What are their types?

A Maven repository is a location where all the project jars, library jars, plugins or any other particular project related artifacts are stored and can be easily used by Maven.

Their types are local, central and remote

7) Why Maven Plugins are used?

Maven plugins are used to

Create a jar file

Create war file

Compile code files

Unit testing of code

Documenting projects

Reporting

8) List out the dependency scope in Maven?

The various dependency scope used in Maven are:

**Compile:** It is the default scope, and it indicates what dependency is available in the classpath of the project

**Provided:** It indicates that the dependency is provided by JDK or web server or container at runtime

**Runtime:** This tells that the dependency is not needed for compilation but is required during execution

**Test:** It says dependency is available only for the test compilation and execution phases

**System:** It indicates you have to provide the system path

**Import:** This indicates that the identified or specified POM should be replaced with the dependencies in that POM’s section

9) Mention how profiles are specified in Maven?

Profiles are specified in Maven by using a subset of the elements existing in the POM itself.

10) Explain how you can exclude dependency?

By using the exclusion element, dependency can be excluded

11) Mention the difference between Apache Ant and Maven?

Apache Ant Maven

Ant is a toolbox – Maven is a framework

Ant does not have formal conventions like project directory structure – Maven has conventions

Ant is procedural; you have to tell to compile, copy and compress – Maven is declarative ( information on what to make & how to build)

Ant does not have lifecycle; you have to add sequence of tasks manually – Maven has a lifecycle

Ant scripts are not reusable – Maven plugins are reusable

12) In Maven what are the two setting files called and what are their location?

In Maven, the setting files are called settings.xml, and the two setting files are located at

Maven installation directory: $M2\_Home/conf/settings.xml

User’s home directory: ${ user.home }/ .m2 / settings.xml

13) List out what are the build phases in Maven?

Build phases in Maven are

Validate

Compile

Test

Package

Install

Deploy

14) List out the build, source and test source directory for POM in Maven?

Build = Target

Source = src/main/java

Test = src/main/test

15) Where do you find the class files when you compile a Maven project?

You will find the class files ${basedir}/target/classes/.

16) Explain what would the “jar: jar” goal do?

jar: jar will not recompile sources; it will imply just create a JAR from the target/classes directory considering that everything else has been done

17) List out what are the Maven’s order of inheritance?

The maven’s order of inheritance is

Parent Pom

Project Pom

Settings

CLI parameters

18) For POM what are the minimum required elements?

The minimum required elements for POM are project root, modelVersion, groupID, artifactID and version.

19) Explain how you can produce execution debug output or error messages?

To produce execution debug output you could call Maven with X parameter or e parameter.

20) Explain how to run test classes in Maven?

To run test classes in Maven, you need surefire plugin, check and configure your settings in setting.xml and pom.xml for a property named “test.”

These interview questions will also help in your viva(orals)

<https://www.guru99.com/maven-interview-questions.html>

--end