

Maven

Maven is a open-source software project management build tool developed by the Apache Group based on Project Object Model (POM). The tool is typically used for Java-based projects. Every Java project requires certain dependencies, which are automatically downloaded when running a Maven build. This simplifies everyday tasks for Java developers and helps them with their projects. In other hand maven automates the entire process of building and developing these applications

During the process, Maven takes care of the following elements:

- Builds
- Dependencies
- Reports
- Distribution
- Releases
- Mailing list

POM

Project Object Model (POM) refers to the XML files with all the information regarding project and configuration details. It contains the project description, as well as details regarding the versioning and configuration management of the project

Dependencies

Dependencies refer to the Java libraries required for the project. Repositories refer to the directories of packaged JAR files.

Repositories

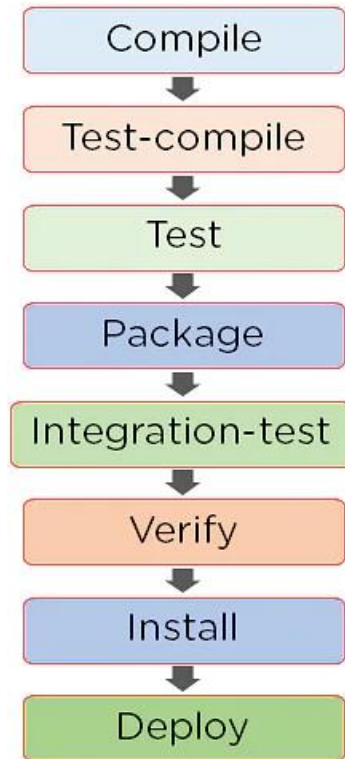
Maven repositories refer to the directories of packaged JAR files that contain metadata. The metadata refers to the POM files relevant to each project. This metadata is what enables Maven to download dependencies

There are three types of repositories:

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

Maven Build Life Cycle

The Maven build life cycle is a sequence of steps that need to be followed in order to build a project, which eventually helps to execute goals.



Why maven?

- The tool simplifies the way of project building by increasing the performance of the project and the building process
- The task of downloading dependencies and JAR files is automated
- Maven provides easy access to all necessary information
- Maven makes it simple for the developer to build a project in different environments without worrying about the dependencies, processes, etc.
- In Maven, it is simple to add new dependencies—you only have to write the dependency code in the POM file

Maven vs gradel

Advantages of Gradle:

1. Its highly customizable property. The tool can be modified under various technologies for diverse projects.
2. The performance of Gradle is very fast and efficient. It is around 2x in speed to that of Maven.
3. Gradle is a tool used for creating plug-ins and is a flexible instrument.
4. It provides a wide variety of IDEs for an enhanced user experience.

Disadvantages of Gradle:

1. Great technical expertise is required to build tasks with Gradle.
2. It does not come with inbuilt ant project structure.
3. Documentation of Gradle is somewhat extensive.
4. Ant build scripts are to be drafted with the help of XML. Moreover , to automate a difficult project , a lot of logic need to be written in XML files.

Advantages of Maven:

1. The process of project building is simplified and well organized.
2. Maven automatically executes the task of downloading Jar files and the other dependencies.
3. Maven can easily incorporate new dependencies by formulating the dependency code in the POM file.
4. It facilitates easy access to all the essential information.
5. It's extensible and plug-ins can be easily written using scripting languages or Java.

Disadvantages of Maven:

1. Installation in the working system is needed.
2. We can not implement a dependency using Maven if the Maven code for existing dependency is not found.
3. In terms of execution of project , Maven is quite slow,

Basis	Gradle	Maven
Based on	Gradle is based on developing domain-specific language projects.	Maven is based on developing pure Java language-based software.
Configuration	It uses a Groovy-based Domain-specific language(DSL) for creating project structure.	It uses Extensible Markup Language(XML) for creating project structure.
Focuses on	Developing applications by adding new features to them.	Developing applications in a given time limit.
Performance	It performs better than maven as it optimized for tracking only current running task.	It does not create local temporary files during software creation, and is hence – slower.
Java Compilation	It avoids compilation.	It is necessary to compile.
Usability	It is a new tool, which requires users to spend a lot of time to get used to it.	This tool is a known tool for many users and is easily available.
Customization	This tool is highly customizable as it supports a variety of IDE's.	This tool serves a limited amount of developers and is not that customizable.
Languages supported	It supports software development in Java, C, C++, and Groovy.	It supports software development in Java, Scala, C#, and Ruby.

Resources:

<https://www.simplilearn.com/tutorials/maven-tutorial/introduction-to-maven>

<https://www.simplilearn.com/tutorials/maven-tutorial/what-is-maven>

<https://maven.apache.org/what-is-maven.html>

<https://www.simplilearn.com/tutorials/project-management-tutorial/project-management-tools>

<https://gradle.org/maven-vs-gradle/>

<https://www.geeksforgeeks.org/difference-between-gradle-and-maven/>