Gradle vs. Maven

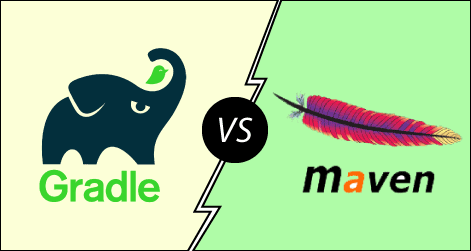
Gradle is one of the several build tools available for Java, but it's not the only build automation tool to consider. Maven is an older and generally used alternative, but which build system is best for our project with other Java frameworks, such as Spring, Hibernate. It is tough to decide which tool is better to use when both tools are coupled with an increasing number of integrations. Let's discuss both tools to understand which tool is better according to our needs.

**Gradle** is a **build automation** tool that is an open-source and builds based on the concepts of Apache Maven and Apache Ant. It is capable of building almost any type of software. It is designed for the multi-project build, which can be quite large. It introduces a **Java and Groovy**-based **DSL** **(Domain Specific Language)** instead of XML (Extensible Markup Language) for declaring the project configuration. It uses a DAG (Directed Acyclic Graph) to define the order of executing the task. Gradle offers an elastic model that can help the development lifecycle from compiling and packaging code for web and mobile applications.

Gradle provides support for the **building**, **testing**, and **deploying software** on different platforms. It has been developed for building automation on many languages and platforms, including Java, Scala, Android, C / C ++, and Groovy. It is the official build tool for Android. Gradle provides integration with several development tools and servers, including **Eclipse**, **IntelliJ**, **Jenkins**, and **Android Studio**.

Some significant benefits of Gradle are as following:

* Gradle allows us to write the build script with Java programing language.
* It is easy to use and maintain.
* It supports dependency management
* It provides high performance and scalable builds.
* Gradle integration process is quite easier.
* It supports a multi-project structure.
* It is easy to migrate to Gradle from Maven or other build tools.



**Maven** is an **open-source software project management** tool that is primarily used for Java projects. It can also be used for other programming projects such as C#, Ruby, Scala, and more. The Apache software foundation manages maven projects. Maven addresses the two main aspects of software development; **dependency**, and how software is built. In maven, An **XML file describes the building process of a project**, its dependencies, components, and other external modules. There are predefined targets for tasks like packaging and compiling.

Note: Gradle is built to overcome the drawbacks of Maven.

The structure of the project can be created automatically in a short duration as Maven uses the standard directory layout and a default build lifecycle that helps to build a project in a concise duration. When there are multiple development teams, Maven can operate in a standard format over a short duration.

Some significant benefits of Maven are as following:

* Maven has enhanced dependency management.
* In Maven, there is no need to store the binary libraries (third party) within the source control.
* It efficiently manages the hierarchical dependency tree.
* It makes the debugging process more straightforward.
* It provides better co-operation among the source code, plugin, libraries, and the IDE.
* It reduces the duplication within the project.

Gradle vs. Maven

Let's discuss some key differences between Gradle and Maven:

|  |  |
| --- | --- |
| **Gradle** | **Maven** |
| It is a build automation system that uses a Groovy-based DSL (domain-specific language ) | It is a software project management system that is primarily used for java projects. |
| It does not use an XML file for declaring the project configuration. | It uses an XML file for declaring the project, its dependencies, the build order, and its required plugin. |
| It is based on a graph of task dependencies that do the work. | It is based on the phases of the fixed and linear model. |
| In Gradle, the main goal is to add functionality to the project. | In maven, the main goal is related to the project phase. |
| It avoids the work by tracking input and output tasks and only runs the tasks that have been changed. Therefore it gives a faster performance.(incremental build) | It does not use the build cache; thus, its build time is slower than Gradle. |
| Gradle is highly customizable; it provides a wide range of IDE support custom builds. | Maven has a limited number of parameters and requirements, so customization is a bit complicated. |
| Gradle avoids the compilation of Java. | The compilation is mandatory in Maven. |