

To Update dependencies:
mvn versions:update-properties

What is Maven?

Maven is a popular open-source build tool developed by the Apache Group to build, publish, and deploy several projects at once for better [project management](#). The tool provides allows developers to build and document the lifecycle framework.

Maven is written in [Java](#) and is used to build projects written in [C#](#), Scala, [Ruby](#), etc. Based on the Project Object Model (POM), this tool has made the lives of [Java developers](#) easier while developing reports, checks build and testing automation setups.

Maven focuses on the simplification and standardization of the building process, taking care of the following:

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

What is Maven: Evolution

Maven was created to simplify Jakarta Tribune project building processes. Many of the projects had slightly different ANT files, so Apache developed Maven to handle building multiple projects together, including publishing project information, facilitating team collaboration, deploying projects, and sharing JARS among several projects.

What is Maven: Objective

Maven's purpose is to provide developers with:

- A comprehensive, maintainable, reusable, and simple model for projects.
- A set of tools and plug-ins that can interact with the declarative model.

What is Maven: Features

Maven is loaded with many valuable and useful features, which goes a long way towards explaining its popularity. Here are some of Maven's more noteworthy features:

- A huge, continuously growing repository of user libraries
- The ability to set up projects easily, using best practices
- Dependency management, featuring automatic updating
- Backwards compatible with previous versions
- Strong error and integrity reporting
- Automatic parent versioning
- Ensures consistent usage across all projects
- It's extensible, and you can easily write plug-ins using scripting languages or Java.

Build Tools

Build tools are the tools or programs that help create an executable application from the source code. As the name suggests, it's essential for building or scripting a wide variety of tasks.

The build tool is needed for the following processes:

- Generating source code
- Generating documentation from the source code
- Compiling source code
- Packaging the compiled codes into JAR files
- Installing the packaged code in the local repository, server, or central repository

Project Object Model (POM)

Maven is so useful thanks to the Project Object Model (POM), which is an [XML file](#) that has all the information regarding project and configuration details. The POM has the description of the project, details regarding the versioning, and configuration management of the project.

The XML file is located in the project home directory. When you execute a task, Maven searches for the POM in the current directory.

The Need for Maven

Maven is chiefly used for Java-based projects, helping to download dependencies, which refers to the libraries or JAR files. The tool helps get the right JAR files for each project as there may be different versions of separate packages.

After Maven, downloading dependencies doesn't require visiting the official websites of different software. You can visit [mvnrepository](#) to find libraries in different languages. The tool also helps to create the right project structure in struts, servlets, etc., which is essential for execution.

How to Use Maven

When you're ready to start [using Maven](#), keep these three things in mind:

1. Configure Maven in Java, using Project Object Model (POM) found in a pom.xml file.
2. All Maven-related configuration settings are found in the POM. You can edit and configure plug-ins in the <plugins> tag of a pom.xml file.
3. Maven provides default settings for configurations, so you don't have to add every configuration into the pom.xml file.

Steps/Process Involved in Building a Project

Here are the steps to follow when building a [Maven project](#):

- Add or write the code to create the application creation, and process it into the source code repository.
- Edit any necessary configuration / pom.XML / plugin details.
- Build the actual application.
- Save your build process output as either a WAR or EAR file to a local server or other location.
- Access the file from the local location or server and deploy it to the production or client site.
- Update the application document by changing the date and updated application version number, if necessary.
- Create and generate a report as requested for the application or the requirement.

Advantages of Maven

- Helps manage all the processes, such as building, documentation, releasing, and distribution in project management
- Simplifies the process of project building
- Increases the performance of the project and the building process
- The task of downloading Jar files and other dependencies is done automatically
- Provides easy access to all the required information
- Makes it easy for the developer to build a project in different environments without worrying about the dependencies, processes, etc.
- In Maven, it's easy to add new dependencies by writing the dependency code in the pom file

Conversely, Maven has a few drawbacks.

- Maven requires installation in the working system and the Maven plug-in for the IDE
- If the Maven code for an existing dependency is unavailable, you cannot add that dependency using Maven itself
- Some [sources](#) claim that Maven is slow