

# Maven Lite

---

- Voir la [documentation Française](#)
- See the [PDF documentation](#)
- Version: 2.0.0
- Author: Floris Robart
- Translated by: Chat GPT-3.5

## Table of Contents

- [Maven Lite](#)
  - [Table of Contents](#)
  - [Compatibility](#)
  - [Description](#)
  - [Maven Lite Installation and Dependencies](#)
  - [Using Maven Lite](#)

## Compatibility

- Windows 10 (Not tested)
- Windows 11 (Tested on Windows 11 22H2)
- Linux with [bash](#) (Tested on Ubuntu 23.10 Mantic Minotaur)
- MacOS Mojave 10.14.6 and lower (Not tested)
- MacOS Catalina 10.15 and higher with [bash](#) installed (Not tested)

## Description

Maven Lite is a Java project management system that allows for the simple and fast compilation and execution of Java projects.

The goal is to be able to compile and/or run a Java project with a single short command, like Maven but simpler.

Maven Lite is simpler than Maven because it does not handle dependencies outside the local environment, plugins, build phases, automatic deployments, etc. It is therefore suitable for simple projects that do not require managing external dependencies. However, Maven Lite handles local dependencies in the form of JAR files. For more complex projects, it is recommended to use Maven.

Maven Lite allows you to compile the project into a target folder, manage local dependencies (JAR files), and execute the project.

The main advantage of Maven Lite in managing dependencies is to provide a folder that contains all dependencies; the script will automatically add all dependencies to the classpath.

This second version of Maven Lite may contain bugs. If you find any, feel free to report them.

A configuration file is available to configure Maven Lite options. It is possible to do without a configuration file and put the options in the command line, but it is less convenient.

The order of the options does not matter.

Each option can be used multiple times.

Each argument must be preceded by its option; you cannot put an argument without an option or put all the options at the beginning and all the arguments at the end. For example, `mvnl --source src/main/java --target target` is valid, but `mvnl --source --target src/main/java target` is not.

**It is recommended to leave the default options because Maven Lite is based on simplified Maven conventions, which are naming and directory conventions known to many Java developers.**

## Maven Lite Installation and Dependencies

[See the documentation on Maven Lite installation](#)

## Using Maven Lite

[See the documentation on using Maven Lite](#)