



Java Build Tools

V.V.S.K Chaitanya

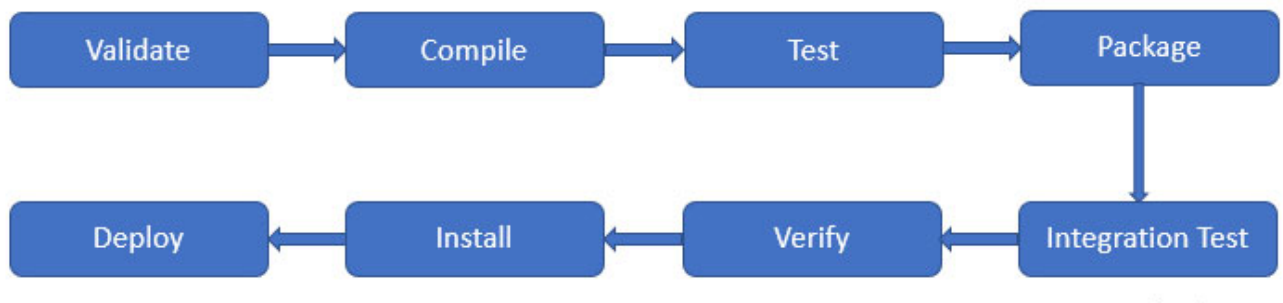
Build Tool?

Build Tool is a software utility tool used for building software projects also referred as **Project Management Tool**. Mainly helps for automation of build process for our software application, reduces a lot of manual effort and helps to quickly convert our application code into executable / working mode.

Features

1. **Project Management:** To define structure, layout and lifecycle of a project.
2. **Dependency Management:** Declare and control the dependencies required
3. **Plugin Management:** Include extra plugins to further automate build process
4. **Version Control:** Efficiently track the version changes for application code.
5. **Task Configuration:** Include tasks for cleanup, testing, packing, deploying.
6. **Extensibility:** Extend for creating custom tasks, plugins and scripts.

Common Build Process



Outcome

The outcome of the build tool is a **well-packaged software product** (application) that can be installed / executed on target machine (server) (or) a usable library that can be delivered to clients.



Top Java Build Tools:



Apache Maven is a software project management and comprehension tool. Based on the concept of a **project object model (POM)**, Maven can manage a project's build, reporting and documentation from a central piece of information.



Gradle is a build automation tool for multi-language software development. It is based on **Groovy and Kotlin**. It controls the development process in the tasks of compilation and packaging to testing, deployment, and publishing.



Apache Ant (**A**nother **N**eat **T**ool) is an open source project started by Apache Software Foundation. Ant is a Java library and a software tool used for automating software build processes such as compile, run, test and assemble Java applications.

Hands On:

1. Install Maven Build Tool

- Download and configure Maven using <https://maven.apache.org/download.cgi>

2. Generate Spring Project

- Open <https://start.spring.io/>
- Select project = Maven and language = Java and add Dependency Spring Web as shown below

The screenshot shows the Spring Initializr web application. The browser address bar displays 'start.spring.io'. The page features the 'spring initializr' logo. On the left, under 'Project', 'Gradle - Groovy' is selected. Under 'Language', 'Java' is selected. Under 'Spring Boot', '3.2.2 (SNAPSHOT)' is selected. The 'Project Metadata' section has 'Group' as 'com.example', 'Artifact' as 'demo', and 'Name' as 'demo'. On the right, under 'Dependencies', 'Spring Web' is added with a 'WEB' tag. A button 'ADD DEPENDENCIES... CTRL + B' is visible. At the bottom, there are three buttons: 'GENERATE CTRL + G', 'EXPLORE CTRL + SPACE', and 'SHARE...'.

- Click “Generate” to download the base spring boot web project setup with required spring boot dependencies.

○

3. Run Maven

- Unzip the downloaded project and open CMD inside project directory
- Make sure the directory got **pom.xml** file
- From CMD, run “**mvn -version**” to show downloaded maven version
- Now run “**mvn clean install**” to run build on our downloaded project.
- Wait for maven to download project dependencies and finally to display the message “**Build Successful**”.
- **Congrats!** You have completed simple steps of running Maven Build Tool.