

Build tools

What is a build tool?

A **build tool** is a program that automates the process of creating executable applications from source code.

The build process includes compiling sources and linking and packaging the code into a usable or executable form.

In small projects (like projects for learning), developers can manually invoke the build process.

However, this approach is not efficient for larger projects, when it is pretty hard to keep track of what needs to be built. Automating the build process minimises the risk of human error.

Additionally, an automated build tool typically runs faster than someone performing the same steps manually. As a consequence, an automated build process improves the quality of the product and saves time and money.

What can build tools do?

Modern build tools can perform a wide variety of tasks that software developers do in their day-to-day activities:

1. **Downloading and adding dependencies.** This is especially convenient when your project depends on a large number of libraries.
2. **Compiling source code into bytecode.** Build tools will invoke the compiler for all the files in your project.
3. **Packaging compiled code.** You will have a production-ready application archive like JAR, APK, or some other.
4. **Running tests.** For example, testing the application archive every time to check if it works correctly. It allows programmers to avoid bugs after modifying the application.
5. **Deploying** to a production environment.

This list of tasks is not complete and may be different depending on the particular build tool used. Some additional features might be available; for example, you can use some tools to generate documentation after the build.

Build tools for Java-based projects

There are three main build tools for Java-based projects: **Apache Ant**, **Apache Maven**, and **Gradle**.

Apache Ant was released in 2000. It is the oldest of these three build tools. Coders rarely use **Ant** in new projects but it still occurs in practice. You can use

this tool together with **Apache Ivy** to manage dependencies.

Apache Maven was released in 2004, and now it is one of the most popular choices for Java developers (especially for server-side development). Many projects, both old and new, use Maven as a build tool because of its powerful dependency management possibilities.

Maven follows the *Convention Over Configuration* concept which means that a developer needs to specify only unconventional aspects of the application, and all standard aspects work by default.

Gradle is a new tool compared to Ant and Maven. It was released in 2007 and is now standard for Android applications. Also, developers use it for server and desktop development.

Gradle aims to *“combine the power and flexibility of Ant with the dependency management and conventions of Maven into a more effective way to build.”*

All of these build tools are free and can be used in any operating system.

Note: **Apache Maven** and **Gradle** are more than simply build tools. They manage almost the entire lifecycle of an application.

There is also another build tool called **sbt** (*Scala Build Tool*). It is primarily used for Scala projects but you can use it for Java or Kotlin as well.