[Dependency management](https://docs.gradle.org/current/userguide/java_plugin.html" \l "sec:java_plugin_and_dependency_management)

The Java plugin adds a number of [dependency configurations](https://docs.gradle.org/current/userguide/declaring_dependencies.html#sec:what-are-dependency-configurations) to your project, as shown below. Tasks such as compileJava and test then use one or more of those configurations to get the corresponding files and use them, for example by placing them on a compilation or runtime classpath.

[Dependency configurations](https://docs.gradle.org/current/userguide/java_plugin.html#tab:configurations)

|  |  |
| --- | --- |
|  | To find information on the api configuration, please consult the [Java Library Plugin](https://docs.gradle.org/current/userguide/java_library_plugin.html#sec:java_library_separation) reference documentation and [Dependency Management for Java Projects](https://docs.gradle.org/current/userguide/dependency_management_for_java_projects.html#dependency_management_for_java_projects). |

**~~compile~~(Deprecated)**

Compile time dependencies. Superseded by implementation.

**implementation extends compile**

Implementation only dependencies.

**compileOnly**

Compile time only dependencies, not used at runtime.

**compileClasspath extends compile, compileOnly, implementation**

Compile classpath, used when compiling source. Used by task compileJava.

**~~runtime~~(Deprecated) extends compile**

Runtime dependencies. Superseded by runtimeOnly.

**runtimeOnly**

Runtime only dependencies.

**runtimeClasspath extends runtimeOnly, runtime, implementation**

Runtime classpath contains elements of the implementation, as well as runtime only elements.

**~~testCompile~~(Deprecated) extends compile**

Additional dependencies for compiling tests. Superseded by testImplementation.

**testImplementation extends testCompile, implementation**

Implementation only dependencies for tests.

**testCompileOnly**

Additional dependencies only for compiling tests, not used at runtime.

**testCompileClasspath extends testCompile, testCompileOnly, testImplementation**

Test compile classpath, used when compiling test sources. Used by task compileTestJava.

**~~testRuntime~~(Deprecated) extends runtime, testCompile**

Additional dependencies for running tests only. Superseded by testRuntimeOnly.

**testRuntimeOnly extends runtimeOnly**

Runtime only dependencies for running tests.

**testRuntimeClasspath extends testRuntimeOnly, testRuntime, testImplementation**

Runtime classpath for running tests. Used by task test.

**archives**

Artifacts (e.g. jars) produced by this project. Used by task uploadArchives.

**default extends runtimeClasspath**

The default configuration used by a project dependency on this project. Contains the artifacts and dependencies required by this project at runtime.

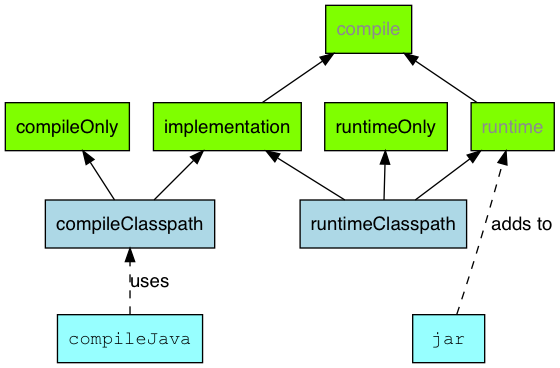
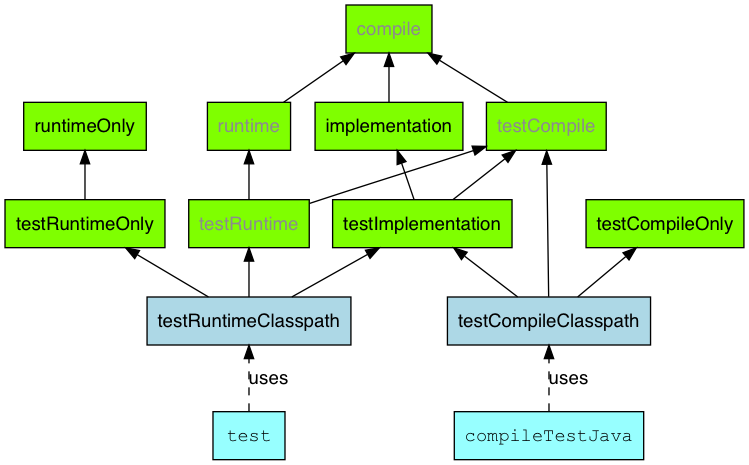
The following diagrams show the dependency configurations for the *main* and *test* source sets respectively. You can use this legend to interpret the colors:

**Gray text — the configuration is *deprecated*.**

**Green background — you can declare dependencies against the configuration.**

**Blue-gray background — the configuration is for consumption by tasks, not for you to declare dependencies.**

**Light blue background with monospace text — a task.**

‑‑‑*Figure 2. Java plugin - main source set dependency configurations*

*Figure 3. Java plugin - test source set dependency configurations*