Docs

[User Manual](http://docs.google.com/userguide/userguide.html)

[Guides and Tutorials](https://guides.gradle.org)

[DSL Reference](http://docs.google.com/dsl/)

[Javadoc](http://docs.google.com/javadoc/)

[Release Notes](http://docs.google.com/release-notes.html)

[Forums](https://discuss.gradle.org/)

[Training](https://gradle.org/training/)

[Try Gradle Enterprise](https://gradle.com/enterprise)

[PDF](http://docs.google.com/userguide/userguide.pdf)

* [User Manual Home](http://docs.google.com/userguide/userguide.html)
* [Release Notes](http://docs.google.com/release-notes.html)
* [Installing Gradle](http://docs.google.com/userguide/installation.html)
* [Tutorials](https://guides.gradle.org/)

### Reference

* [Groovy DSL Reference](http://docs.google.com/dsl/)
* [Gradle API Javadoc](http://docs.google.com/javadoc/)
* [Core Plugins](http://docs.google.com/userguide/plugin_reference.html)
* [Gradle & Third-party Tools](http://docs.google.com/userguide/third_party_integration.html)

### Getting Started

* [Creating New Gradle Builds](https://guides.gradle.org/creating-new-gradle-builds/)
* [Creating Build Scans](https://guides.gradle.org/creating-build-scans/)
* [Migrating From Maven](https://guides.gradle.org/migrating-from-maven/)

### Running Gradle Builds

* [Command-Line Interface](http://docs.google.com/userguide/command_line_interface.html)
* [Customizing Execution](#gjdgxs)
  + [Configuring the Build Environment](http://docs.google.com/userguide/build_environment.html)
  + [Configuring the Gradle Daemon](http://docs.google.com/userguide/gradle_daemon.html)
  + [Initialization Scripts](http://docs.google.com/userguide/init_scripts.html)
* [Directory Layout](http://docs.google.com/userguide/directory_layout.html)
* [Executing Multi-Project Builds](http://docs.google.com/userguide/intro_multi_project_builds.html)
* [Gradle Wrapper](http://docs.google.com/userguide/gradle_wrapper.html)
* [Troubleshooting](http://docs.google.com/userguide/troubleshooting.html)
* [Using Build Scans](https://docs.gradle.com/build-scan-plugin)
* [Enabling and Configuring the Build Cache](http://docs.google.com/userguide/build_cache.html)
* [Integrating Separate Gradle Builds (Composite Builds)](http://docs.google.com/userguide/composite_builds.html)

### Authoring Gradle Builds

* [Fundamentals](#30j0zll)
  + [Introducing the Basics of Build Scripts](http://docs.google.com/userguide/tutorial_using_tasks.html)
  + [Working with Tasks](http://docs.google.com/userguide/more_about_tasks.html)
  + [Learning More About Build Scripts](http://docs.google.com/userguide/writing_build_scripts.html)
  + [Working with Files](http://docs.google.com/userguide/working_with_files.html)
  + [Creating Custom Task Types](http://docs.google.com/userguide/custom_tasks.html)
  + [Using Gradle Plugins](http://docs.google.com/userguide/plugins.html)
  + [The Standard Gradle Plugins](http://docs.google.com/userguide/standard_plugins.html)
  + [Understanding the Build Lifecycle](http://docs.google.com/userguide/build_lifecycle.html)
  + [Working with Logging](http://docs.google.com/userguide/logging.html)
  + [Configuring Multi-Project Builds](http://docs.google.com/userguide/multi_project_builds.html)
* [Best Practices](#1fob9te)
  + [Authoring Maintainable Build Scripts](http://docs.google.com/userguide/authoring_maintainable_build_scripts.html)
  + [Organizing Gradle Projects](http://docs.google.com/userguide/organizing_gradle_projects.html)
  + [Optimizing Build Performance](https://guides.gradle.org/performance/)
  + [Using the Build Cache](https://guides.gradle.org/using-build-cache/)
* [Dependency Management](#3znysh7)
  + [Introduction to Dependency Management](http://docs.google.com/userguide/introduction_dependency_management.html)
  + [Dependency Management Terminology](http://docs.google.com/userguide/dependency_management_terminology.html)
  + [Dependency Types](http://docs.google.com/userguide/dependency_types.html)
  + [Repository Types](http://docs.google.com/userguide/repository_types.html)
  + [Declaring Dependencies](http://docs.google.com/userguide/declaring_dependencies.html)
  + [Declaring Repositories](http://docs.google.com/userguide/declaring_repositories.html)
  + [Inspecting Dependencies](http://docs.google.com/userguide/inspecting_dependencies.html)
  + [Managing Dependency Configurations](http://docs.google.com/userguide/managing_dependency_configurations.html)
  + [Managing Transitive Dependencies](http://docs.google.com/userguide/managing_transitive_dependencies.html)
  + [Dependency Locking](http://docs.google.com/userguide/dependency_locking.html)
  + [Troubleshooting Dependency Resolution](http://docs.google.com/userguide/troubleshooting_dependency_resolution.html)
  + [Customizing Dependency Resolution Behavior](http://docs.google.com/userguide/customizing_dependency_resolution_behavior.html)
  + [Dependency Cache Internals](http://docs.google.com/userguide/dependency_cache.html)
  + [Working with Dependencies](http://docs.google.com/userguide/working_with_dependencies.html)
* [Publishing Artifacts](http://docs.google.com/userguide/artifact_management.html)
* [C++ Projects](#2et92p0)
  + [Building Native Software](http://docs.google.com/userguide/native_software.html)
  + [Software Model Concepts](http://docs.google.com/userguide/software_model_concepts.html)
  + [Rule-based Model Configuration](http://docs.google.com/userguide/software_model.html)
  + [Implementing Model Rules in a Plugin](http://docs.google.com/userguide/rule_source.html)
  + [Extending the Software Model](http://docs.google.com/userguide/software_model_extend.html)
* [Java Projects](#tyjcwt)
  + [Building Java & JVM projects](http://docs.google.com/userguide/building_java_projects.html)
  + [Testing Java & JVM projects](http://docs.google.com/userguide/java_testing.html)
* [Advanced Techniques](#3dy6vkm)
  + [Configuring Tasks Lazily](http://docs.google.com/userguide/lazy_configuration.html)
  + [Developing Parallel Tasks](https://guides.gradle.org/using-the-worker-api/)
  + [Testing Your Build with TestKit](http://docs.google.com/userguide/test_kit.html)
  + [Using Ant from Gradle](http://docs.google.com/userguide/ant.html)
* [Sample Gradle builds](#1t3h5sf)
  + [Groovy DSL Samples](https://github.com/gradle/gradle/tree/master/subprojects/docs/src/samples)
  + [Kotlin DSL Samples](https://github.com/gradle/kotlin-dsl/tree/master/samples)

### Extending Gradle

* [Writing Custom Plugins](http://docs.google.com/userguide/custom_plugins.html)
* [Plugin Development Guides](https://gradle.org/guides/?q=Plugin+Development)

[Edit this page](https://github.com/gradle/gradle/edit/master/subprojects/docs/src/docs/userguide/)

# Dependency Types

Contents

[Module dependencies](#4d34og8)

[File dependencies](#2s8eyo1)

[Project dependencies](#17dp8vu)

[Gradle distribution-specific dependencies](#3rdcrjn)

[Module dependencies](#4d34og8)

Module dependencies are the most common dependencies. They refer to a module in a repository.

[Example: Module dependencies](#26in1rg)

**build.gradle**

dependencies {  
 runtime group: 'org.springframework', name: 'spring-core', version: '2.5'  
 runtime 'org.springframework:spring-core:2.5',  
 'org.springframework:spring-aop:2.5'  
 runtime(  
 [group: 'org.springframework', name: 'spring-core', version: '2.5'],  
 [group: 'org.springframework', name: 'spring-aop', version: '2.5']  
 )  
 runtime('org.hibernate:hibernate:3.0.5') {  
 transitive = true  
 }  
 runtime group: 'org.hibernate', name: 'hibernate', version: '3.0.5', transitive: true  
 runtime(group: 'org.hibernate', name: 'hibernate', version: '3.0.5') {  
 transitive = true  
 }  
}

See the [DependencyHandler](http://docs.google.com/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html) class in the API documentation for more examples and a complete reference.

Gradle provides different notations for module dependencies. There is a string notation and a map notation. A module dependency has an API which allows further configuration. Have a look at [ExternalModuleDependency](http://docs.google.com/javadoc/org/gradle/api/artifacts/ExternalModuleDependency.html) to learn all about the API. This API provides properties and configuration methods. Via the string notation you can define a subset of the properties. With the map notation you can define all properties. To have access to the complete API, either with the map or with the string notation, you can assign a single dependency to a configuration together with a closure.

| **✨** | If you declare a module dependency, Gradle looks for a module metadata file (.module, .pom or ivy.xml) in the repositories. If such a module metadata file exists, it is parsed and the artifacts of this module (e.g. hibernate-3.0.5.jar) as well as its dependencies (e.g. cglib) are downloaded. If no such module metadata file exists, Gradle may look, depending on the [metadata sources definitions](http://docs.google.com/repository_types.html#sub:supported_metadata_sources), for an artifact file called hibernate-3.0.5.jar directly. In Maven, a module can have one and only one artifact. In Gradle and Ivy, a module can have multiple artifacts. Each artifact can have a different set of dependencies. |
| --- | --- |

[File dependencies](#2s8eyo1)

File dependencies allow you to directly add a set of files to a configuration, without first adding them to a repository. This can be useful if you cannot, or do not want to, place certain files in a repository. Or if you do not want to use any repositories at all for storing your dependencies.

To add some files as a dependency for a configuration, you simply pass a [file collection](http://docs.google.com/working_with_files.html#sec:file_collections) as a dependency:

[Example: File dependencies](#lnxbz9)

**build.gradle**

dependencies {  
 runtime files('libs/a.jar', 'libs/b.jar')  
 runtime fileTree(dir: 'libs', include: '\*.jar')  
}

File dependencies are not included in the published dependency descriptor for your project. However, file dependencies are included in transitive project dependencies within the same build. This means they cannot be used outside the current build, but they can be used with the same build.

You can declare which tasks produce the files for a file dependency. You might do this when, for example, the files are generated by the build.

[Example: Generated file dependencies](#35nkun2)

**build.gradle**

dependencies {  
 compile files("$buildDir/classes") {  
 builtBy 'compile'  
 }  
}  
  
task compile {  
 doLast {  
 println 'compiling classes'  
 }  
}  
  
task list(dependsOn: configurations.compile) {  
 doLast {  
 println "classpath = ${configurations.compile.collect { File file -> file.name }}"  
 }  
}

**Output of** gradle -q list

> gradle -q list  
compiling classes  
classpath = [classes]

[Project dependencies](#17dp8vu)

Gradle distinguishes between external dependencies and dependencies on projects which are part of the same multi-project build. For the latter you can declare *project dependencies*.

[Example: Project dependencies](#1ksv4uv)

**build.gradle**

dependencies {  
 compile project(':shared')  
}

For more information see the API documentation for [ProjectDependency](http://docs.google.com/javadoc/org/gradle/api/artifacts/ProjectDependency.html).

Multi-project builds are discussed in [this chapter](http://docs.google.com/multi_project_builds.html#multi_project_builds).

[Gradle distribution-specific dependencies](#3rdcrjn)

[Gradle API dependency](#44sinio)

You can declare a dependency on the API of the current version of Gradle by using the [DependencyHandler.gradleApi()](http://docs.google.com/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:gradleApi()) method. This is useful when you are developing custom Gradle tasks or plugins.

[Example: Gradle API dependencies](#2jxsxqh)

**build.gradle**

dependencies {  
 compile gradleApi()  
}

[Gradle TestKit dependency](#z337ya)

You can declare a dependency on the TestKit API of the current version of Gradle by using the [DependencyHandler.gradleTestKit()](http://docs.google.com/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:gradleTestKit()) method. This is useful for writing and executing functional tests for Gradle plugins and build scripts.

[Example: Gradle TestKit dependencies](#3j2qqm3)

**build.gradle**

dependencies {  
 testCompile gradleTestKit()  
}

[The TestKit chapter](http://docs.google.com/test_kit.html#test_kit) explains the use of TestKit by example.

[Local Groovy dependency](#1y810tw)

You can declare a dependency on the Groovy that is distributed with Gradle by using the [DependencyHandler.localGroovy()](http://docs.google.com/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:localGroovy()) method. This is useful when you are developing custom Gradle tasks or plugins in Groovy.

[Example: Gradle’s Groovy dependencies](#4i7ojhp)

**build.gradle**

dependencies {  
 compile localGroovy()  
}

Docs

* [User Manual](http://docs.google.com/userguide/userguide.html)
* [DSL Reference](http://docs.google.com/dsl/)
* [Release Notes](http://docs.google.com/release-notes.html)
* [Javadoc](http://docs.google.com/javadoc/)

News

* [Blog](https://blog.gradle.org/)
* [Newsletter](https://newsletter.gradle.com/)
* [Twitter](https://twitter.com/gradle)

Products

* [Build Scans](https://gradle.com/build-scans)
* [Build Cache](https://gradle.com/build-cache)
* [Enterprise Docs](https://gradle.com/enterprise/resources)

Get Help

* [Forums](https://discuss.gradle.org/c/help-discuss)
* [GitHub](https://github.com/gradle/)
* [Training](https://gradle.org/training/)
* [Services](https://gradle.org/services/)

Subscribe for important Gradle updates and news

Subscribe

By entering your email, you agree to our [Terms](https://gradle.org/terms/) and [Privacy Policy](https://gradle.org/privacy/), including receipt of emails. You can unsubscribe at any time.

© [Gradle Inc.](https://gradle.com) 2018 All rights reserved.

[Careers](https://gradle.com/careers) | [Privacy](https://gradle.org/privacy) | [Terms of Service](https://gradle.org/terms) | [Contact](https://gradle.org/contact/)