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/)

# The Groovy Plugin

Contents

[Usage](#4d34og8)

[Tasks](#2s8eyo1)

[Project layout](#17dp8vu)

[Dependency management](#3rdcrjn)

[Automatic configuration of groovyClasspath](#26in1rg)

[Convention properties](#lnxbz9)

[Source set properties](#35nkun2)

[GroovyCompile](#1ksv4uv)

[Compiling and testing for Java 6 or Java 7](#44sinio)

The Groovy plugin extends the Java plugin to add support for Groovy projects. It can deal with Groovy code, mixed Groovy and Java code, and even pure Java code (although we don’t necessarily recommend to use it for the latter). The plugin supports *joint compilation*, which allows you to freely mix and match Groovy and Java code, with dependencies in both directions. For example, a Groovy class can extend a Java class that in turn extends a Groovy class. This makes it possible to use the best language for the job, and to rewrite any class in the other language if needed.

[Usage](#4d34og8)

To use the Groovy plugin, include the following in your build script:

[Example: Using the Groovy plugin](#2jxsxqh)

**build.gradle**

apply plugin: 'groovy'

[Tasks](#2s8eyo1)

The Groovy plugin adds the following tasks to the project.

compileGroovy — [GroovyCompile](http://docs.google.com/dsl/org.gradle.api.tasks.compile.GroovyCompile.html)

*Depends on*: compileJava

Compiles production Groovy source files.

compileTestGroovy — [GroovyCompile](http://docs.google.com/dsl/org.gradle.api.tasks.compile.GroovyCompile.html)

*Depends on*: compileTestJava

Compiles test Groovy source files.

compile*SourceSet*Groovy — [GroovyCompile](http://docs.google.com/dsl/org.gradle.api.tasks.compile.GroovyCompile.html)

*Depends on*: compile*SourceSet*Java

Compiles the given source set’s Groovy source files.

groovydoc — [Groovydoc](http://docs.google.com/dsl/org.gradle.api.tasks.javadoc.Groovydoc.html)

Generates API documentation for the production Groovy source files.

The Groovy plugin adds the following dependencies to tasks added by the Java plugin.

Table 1. Groovy plugin - additional task dependencies

| **Task name** | **Depends on** |
| --- | --- |
| classes | compileGroovy |
| testClasses | compileTestGroovy |
| *sourceSet*Classes | compile*SourceSet*Groovy |



*Figure 1. Groovy plugin - tasks*

[Project layout](#17dp8vu)

The Groovy plugin assumes the project layout shown in [Groovy Layout](#z337ya). All the Groovy source directories can contain Groovy *and* Java code. The Java source directories may only contain Java source code.[[1](#3j2qqm3)] None of these directories need to exist or have anything in them; the Groovy plugin will simply compile whatever it finds.

src/main/java

Production Java source.

src/main/resources

Production resources, such as XML and properties files.

src/main/groovy

Production Groovy source. May also contain Java source files for joint compilation.

src/test/java

Test Java source.

src/test/resources

Test resources.

src/test/groovy

Test Groovy source. May also contain Java source files for joint compilation.

src/*sourceSet*/java

Java source for the source set named *sourceSet*.

src/*sourceSet*/resources

Resources for the source set named *sourceSet*.

src/*sourceSet*/groovy

Groovy source files for the given source set. May also contain Java source files for joint compilation.

[Changing the project layout](#1y810tw)

Just like the Java plugin, the Groovy plugin allows you to configure custom locations for Groovy production and test source files.

[Example: Custom Groovy source layout](#4i7ojhp)

**build.gradle**

sourceSets {  
 main {  
 groovy {  
 srcDirs = ['src/groovy']  
 }  
 }  
  
 test {  
 groovy {  
 srcDirs = ['test/groovy']  
 }  
 }  
}

[Dependency management](#3rdcrjn)

Because Gradle’s build language is based on Groovy, and parts of Gradle are implemented in Groovy, Gradle already ships with a Groovy library. Nevertheless, Groovy projects need to explicitly declare a Groovy dependency. This dependency will then be used on compile and runtime class paths. It will also be used to get hold of the Groovy compiler and Groovydoc tool, respectively.

If Groovy is used for production code, the Groovy dependency should be added to the compile configuration:

[Example: Configuration of Groovy dependency](#2xcytpi)

**build.gradle**

repositories {  
 mavenCentral()  
}  
  
dependencies {  
 compile 'org.codehaus.groovy:groovy-all:2.4.15'  
}

If Groovy is only used for test code, the Groovy dependency should be added to the testCompile configuration:

[Example: Configuration of Groovy test dependency](#1ci93xb)

**build.gradle**

dependencies {  
 testCompile 'org.codehaus.groovy:groovy-all:2.4.15'  
}

To use the Groovy library that ships with Gradle, declare a localGroovy() dependency. Note that different Gradle versions ship with different Groovy versions; as such, using localGroovy() is less safe then declaring a regular Groovy dependency.

[Example: Configuration of bundled Groovy dependency](#3whwml4)

**build.gradle**

dependencies {  
 compile localGroovy()  
}

The Groovy library doesn’t necessarily have to come from a remote repository. It could also come from a local lib directory, perhaps checked in to source control:

[Example: Configuration of Groovy file dependency](#2bn6wsx)

**build.gradle**

repositories {  
 flatDir { dirs 'lib' }  
}  
  
dependencies {  
 compile module('org.codehaus.groovy:groovy:2.4.15') {  
 dependency('org.ow2.asm:asm-all:5.0.3')  
 dependency('antlr:antlr:2.7.7')  
 dependency('commons-cli:commons-cli:1.2')  
 module('org.apache.ant:ant:1.9.4') {  
 dependencies('org.apache.ant:ant-junit:1.9.4@jar',  
 'org.apache.ant:ant-launcher:1.9.4')  
 }  
 }  
}

[Automatic configuration of groovyClasspath](#26in1rg)

The GroovyCompile and Groovydoc tasks consume Groovy code in two ways: on their classpath, and on their groovyClasspath. The former is used to locate classes referenced by the source code, and will typically contain the Groovy library along with other libraries. The latter is used to load and execute the Groovy compiler and Groovydoc tool, respectively, and should only contain the Groovy library and its dependencies.

Unless a task’s groovyClasspath is configured explicitly, the Groovy (base) plugin will try to infer it from the task’s classpath. This is done as follows:

* If a groovy-all(-indy) Jar is found on classpath, that jar will be added to groovyClasspath.
* If a groovy(-indy) jar is found on classpath, and the project has at least one repository declared, a corresponding groovy(-indy) repository dependency will be added to groovyClasspath.
* Otherwise, execution of the task will fail with a message saying that groovyClasspath could not be inferred.

Note that the “-indy” variation of each jar refers to the version with invokedynamic support.

[Convention properties](#lnxbz9)

The Groovy plugin does not add any convention properties to the project.

[Source set properties](#35nkun2)

The Groovy plugin adds the following convention properties to each source set in the project. You can use these properties in your build script as though they were properties of the source set object.

[Groovy Plugin — source set properties](#qsh70q)

groovy — [SourceDirectorySet](http://docs.google.com/dsl/org.gradle.api.file.SourceDirectorySet.html) (read-only)

*Default value*: Not null

The Groovy source files of this source set. Contains all .groovy and .java files found in the Groovy source directories, and excludes all other types of files.

groovy.srcDirs — Set<File>

*Default value*: [*projectDir*/src/*name*/groovy]

The source directories containing the Groovy source files of this source set. May also contain Java source files for joint compilation. Can set using anything described in [Specifying Multiple Files](http://docs.google.com/working_with_files.html#sec:specifying_multiple_files).

allGroovy — [FileTree](http://docs.google.com/javadoc/org/gradle/api/file/FileTree.html) (read-only)

*Default value*: Not null

All Groovy source files of this source set. Contains only the .groovy files found in the Groovy source directories.

These properties are provided by a convention object of type [GroovySourceSet](http://docs.google.com/dsl/org.gradle.api.tasks.GroovySourceSet.html).

The Groovy plugin also modifies some source set properties:

[Groovy Plugin - modified source set properties](#3as4poj)

| **Property name** | **Change** |
| --- | --- |
| allJava | Adds all .java files found in the Groovy source directories. |
| allSource | Adds all source files found in the Groovy source directories. |

[GroovyCompile](#1ksv4uv)

The Groovy plugin adds a [GroovyCompile](http://docs.google.com/dsl/org.gradle.api.tasks.compile.GroovyCompile.html) task for each source set in the project. The task type extends the JavaCompile task (see [the relevant Java Plugin section](http://docs.google.com/building_java_projects.html#sec:compile)). The GroovyCompile task supports most configuration options of the official Groovy compiler.

Table 2. Groovy plugin - GroovyCompile properties

| **Task Property** | **Type** | **Default Value** |
| --- | --- | --- |
| classpath | [FileCollection](http://docs.google.com/javadoc/org/gradle/api/file/FileCollection.html) | *sourceSet*.compileClasspath |
| source | [FileTree](http://docs.google.com/javadoc/org/gradle/api/file/FileTree.html). Can set using anything described in [Specifying Multiple Files](http://docs.google.com/working_with_files.html#sec:specifying_multiple_files). | *sourceSet*.groovy |
| destinationDir | File. | *sourceSet*.groovy.outputDir |
| groovyClasspath | [FileCollection](http://docs.google.com/javadoc/org/gradle/api/file/FileCollection.html) | groovy configuration if non-empty; Groovy library found on classpath otherwise |

[Compiling and testing for Java 6 or Java 7](#44sinio)

The Groovy compiler will always be executed with the same version of Java that was used to start Gradle. You should set sourceCompatibility and targetCompatibility to 1.6 or 1.7. If you also have Java source files, you can follow the same steps as for the [Java plugin](http://docs.google.com/building_java_projects.html#sec:java_cross_compilation) to ensure the correct Java compiler is used.

[Example: Configure Java 6 build for Groovy](#1pxezwc)

**gradle.properties**

# in $HOME/.gradle/gradle.properties  
java6Home=/Library/Java/JavaVirtualMachines/1.6.0.jdk/Contents/Home

**build.gradle**

sourceCompatibility = 1.6  
targetCompatibility = 1.6  
  
assert hasProperty('java6Home') : "Set the property 'java6Home' in your your gradle.properties pointing to a Java 6 installation"  
def javaExecutablesPath = new File(java6Home, 'bin')  
def javaExecutables = [:].withDefault { execName ->  
 def executable = new File(javaExecutablesPath, execName)  
 assert executable.exists() : "There is no ${execName} executable in ${javaExecutablesPath}"  
 executable  
}  
tasks.withType(AbstractCompile) {  
 options.with {  
 fork = true  
 forkOptions.javaHome = file(java6Home)  
 }  
}  
tasks.withType(Javadoc) {  
 executable = javaExecutables.javadoc  
}  
tasks.withType(Test) {  
 executable = javaExecutables.java  
}  
tasks.withType(JavaExec) {  
 executable = javaExecutables.java  
}

[1](#49x2ik5). Gradle uses the same conventions as introduced by Russel Winder’s [Gant tool](https://gant.github.io/).

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/)