4.10.1

* 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)
* [User Manual Home](http://docs.google.com/userguide/userguide.html)
* [DSL Reference Home](http://docs.google.com/dsl/)
* [Release Notes](http://docs.google.com/release-notes.html)
  + [Properties](http://docs.google.com/org.gradle.api.initialization.Settings.html#N18CDB)
  + [Methods](http://docs.google.com/org.gradle.api.initialization.Settings.html#N18D3F)

### Build script blocks

* [allprojects { }](http://docs.google.com/org.gradle.api.Project.html#org.gradle.api.Project:allprojects(groovy.lang.Closure))
* [artifacts { }](http://docs.google.com/org.gradle.api.Project.html#org.gradle.api.Project:artifacts(groovy.lang.Closure))
* [buildscript { }](http://docs.google.com/org.gradle.api.Project.html#org.gradle.api.Project:buildscript(groovy.lang.Closure))
* [configurations { }](http://docs.google.com/org.gradle.api.Project.html#org.gradle.api.Project:configurations(groovy.lang.Closure))
* [dependencies { }](http://docs.google.com/org.gradle.api.Project.html#org.gradle.api.Project:dependencies(groovy.lang.Closure))
* [repositories { }](http://docs.google.com/org.gradle.api.Project.html#org.gradle.api.Project:repositories(groovy.lang.Closure))
* [sourceSets { }](http://docs.google.com/org.gradle.api.Project.html#org.gradle.api.Project:sourceSets(groovy.lang.Closure))
* [subprojects { }](http://docs.google.com/org.gradle.api.Project.html#org.gradle.api.Project:subprojects(groovy.lang.Closure))
* [publishing { }](http://docs.google.com/org.gradle.api.Project.html#org.gradle.api.Project:publishing(groovy.lang.Closure))

### Core types

* [Project](http://docs.google.com/org.gradle.api.Project.html)
* [Task](http://docs.google.com/org.gradle.api.Task.html)
* [Gradle](http://docs.google.com/org.gradle.api.invocation.Gradle.html)
* [Settings](http://docs.google.com/org.gradle.api.initialization.Settings.html)
* [IncludedBuild](http://docs.google.com/org.gradle.api.initialization.IncludedBuild.html)
* [Script](http://docs.google.com/org.gradle.api.Script.html)
* [JavaToolChain](http://docs.google.com/org.gradle.jvm.toolchain.JavaToolChain.html)
* [SourceSet](http://docs.google.com/org.gradle.api.tasks.SourceSet.html)
* [SourceSetOutput](http://docs.google.com/org.gradle.api.tasks.SourceSetOutput.html)
* [SourceDirectorySet](http://docs.google.com/org.gradle.api.file.SourceDirectorySet.html)
* [IncrementalTaskInputs](http://docs.google.com/org.gradle.api.tasks.incremental.IncrementalTaskInputs.html)
* [Configuration](http://docs.google.com/org.gradle.api.artifacts.Configuration.html)
* [ResolutionStrategy](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html)
* [ArtifactResolutionQuery](http://docs.google.com/org.gradle.api.artifacts.query.ArtifactResolutionQuery.html)
* [ComponentSelection](http://docs.google.com/org.gradle.api.artifacts.ComponentSelection.html)
* [ComponentSelectionRules](http://docs.google.com/org.gradle.api.artifacts.ComponentSelectionRules.html)
* [ExtensionAware](http://docs.google.com/org.gradle.api.plugins.ExtensionAware.html)
* [ExtraPropertiesExtension](http://docs.google.com/org.gradle.api.plugins.ExtraPropertiesExtension.html)
* [PluginDependenciesSpec](http://docs.google.com/org.gradle.plugin.use.PluginDependenciesSpec.html)
* [PluginDependencySpec](http://docs.google.com/org.gradle.plugin.use.PluginDependencySpec.html)
* [PluginManagementSpec](http://docs.google.com/org.gradle.plugin.management.PluginManagementSpec.html)
* [ResourceHandler](http://docs.google.com/org.gradle.api.resources.ResourceHandler.html)
* [TextResourceFactory](http://docs.google.com/org.gradle.api.resources.TextResourceFactory.html)

### Publishing types

* [PublishingExtension](http://docs.google.com/org.gradle.api.publish.PublishingExtension.html)
* [IvyPublication](http://docs.google.com/org.gradle.api.publish.ivy.IvyPublication.html)
* [IvyArtifact](http://docs.google.com/org.gradle.api.publish.ivy.IvyArtifact.html)
* [IvyArtifactSet](http://docs.google.com/org.gradle.api.publish.ivy.IvyArtifactSet.html)
* [IvyModuleDescriptorSpec](http://docs.google.com/org.gradle.api.publish.ivy.IvyModuleDescriptorSpec.html)
* [IvyModuleDescriptorAuthor](http://docs.google.com/org.gradle.api.publish.ivy.IvyModuleDescriptorAuthor.html)
* [IvyModuleDescriptorLicense](http://docs.google.com/org.gradle.api.publish.ivy.IvyModuleDescriptorLicense.html)
* [IvyModuleDescriptorDescription](http://docs.google.com/org.gradle.api.publish.ivy.IvyModuleDescriptorDescription.html)
* [MavenPublication](http://docs.google.com/org.gradle.api.publish.maven.MavenPublication.html)
* [MavenArtifact](http://docs.google.com/org.gradle.api.publish.maven.MavenArtifact.html)
* [MavenArtifactSet](http://docs.google.com/org.gradle.api.publish.maven.MavenArtifactSet.html)
* [MavenPom](http://docs.google.com/org.gradle.api.publish.maven.MavenPom.html)
* [MavenPomCiManagement](http://docs.google.com/org.gradle.api.publish.maven.MavenPomCiManagement.html)
* [MavenPomContributor](http://docs.google.com/org.gradle.api.publish.maven.MavenPomContributor.html)
* [MavenPomContributorSpec](http://docs.google.com/org.gradle.api.publish.maven.MavenPomContributorSpec.html)
* [MavenPomDeveloper](http://docs.google.com/org.gradle.api.publish.maven.MavenPomDeveloper.html)
* [MavenPomDeveloperSpec](http://docs.google.com/org.gradle.api.publish.maven.MavenPomDeveloperSpec.html)
* [MavenPomDistributionManagement](http://docs.google.com/org.gradle.api.publish.maven.MavenPomDistributionManagement.html)
* [MavenPomIssueManagement](http://docs.google.com/org.gradle.api.publish.maven.MavenPomIssueManagement.html)
* [MavenPomLicense](http://docs.google.com/org.gradle.api.publish.maven.MavenPomLicense.html)
* [MavenPomLicenseSpec](http://docs.google.com/org.gradle.api.publish.maven.MavenPomLicenseSpec.html)
* [MavenPomMailingList](http://docs.google.com/org.gradle.api.publish.maven.MavenPomMailingList.html)
* [MavenPomMailingListSpec](http://docs.google.com/org.gradle.api.publish.maven.MavenPomMailingListSpec.html)
* [MavenPomOrganization](http://docs.google.com/org.gradle.api.publish.maven.MavenPomOrganization.html)
* [MavenPomRelocation](http://docs.google.com/org.gradle.api.publish.maven.MavenPomRelocation.html)
* [MavenPomScm](http://docs.google.com/org.gradle.api.publish.maven.MavenPomScm.html)

### Container types

* [TaskContainer](http://docs.google.com/org.gradle.api.tasks.TaskContainer.html)
* [ConfigurationContainer](http://docs.google.com/org.gradle.api.artifacts.ConfigurationContainer.html)
* [RepositoryHandler](http://docs.google.com/org.gradle.api.artifacts.dsl.RepositoryHandler.html)
* [DependencyHandler](http://docs.google.com/org.gradle.api.artifacts.dsl.DependencyHandler.html)
* [ArtifactHandler](http://docs.google.com/org.gradle.api.artifacts.dsl.ArtifactHandler.html)

### Build Cache types

* [BuildCacheConfiguration](http://docs.google.com/org.gradle.caching.configuration.BuildCacheConfiguration.html)
* [DirectoryBuildCache](http://docs.google.com/org.gradle.caching.local.DirectoryBuildCache.html)
* [HttpBuildCache](http://docs.google.com/org.gradle.caching.http.HttpBuildCache.html)

### Input Normalization types

* [InputNormalizationHandler](http://docs.google.com/org.gradle.normalization.InputNormalizationHandler.html)
* [InputNormalization](http://docs.google.com/org.gradle.normalization.InputNormalization.html)
* [RuntimeClasspathNormalization](http://docs.google.com/org.gradle.normalization.RuntimeClasspathNormalization.html)

### Help Task types

* [TaskReportTask](http://docs.google.com/org.gradle.api.tasks.diagnostics.TaskReportTask.html)
* [ProjectReportTask](http://docs.google.com/org.gradle.api.tasks.diagnostics.ProjectReportTask.html)
* [DependencyReportTask](http://docs.google.com/org.gradle.api.tasks.diagnostics.DependencyReportTask.html)
* [DependencyInsightReportTask](http://docs.google.com/org.gradle.api.tasks.diagnostics.DependencyInsightReportTask.html)
* [PropertyReportTask](http://docs.google.com/org.gradle.api.tasks.diagnostics.PropertyReportTask.html)
* [ComponentReport](http://docs.google.com/org.gradle.api.reporting.components.ComponentReport.html)
* [DependentComponentsReport](http://docs.google.com/org.gradle.api.reporting.dependents.DependentComponentsReport.html)
* [ModelReport](http://docs.google.com/org.gradle.api.reporting.model.ModelReport.html)

### Task types

* [AntlrTask](http://docs.google.com/org.gradle.api.plugins.antlr.AntlrTask.html)
* [BuildEnvironmentReportTask](http://docs.google.com/org.gradle.api.tasks.diagnostics.BuildEnvironmentReportTask.html)
* [Checkstyle](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html)
* [CodeNarc](http://docs.google.com/org.gradle.api.plugins.quality.CodeNarc.html)
* [CompareGradleBuilds](http://docs.google.com/org.gradle.api.plugins.buildcomparison.gradle.CompareGradleBuilds.html)
* [Copy](http://docs.google.com/org.gradle.api.tasks.Copy.html)
* [CreateStartScripts](http://docs.google.com/org.gradle.jvm.application.tasks.CreateStartScripts.html)
* [Delete](http://docs.google.com/org.gradle.api.tasks.Delete.html)
* [Ear](http://docs.google.com/org.gradle.plugins.ear.Ear.html)
* [Exec](http://docs.google.com/org.gradle.api.tasks.Exec.html)
* [FindBugs](http://docs.google.com/org.gradle.api.plugins.quality.FindBugs.html)
* [GenerateIvyDescriptor](http://docs.google.com/org.gradle.api.publish.ivy.tasks.GenerateIvyDescriptor.html)
* [GenerateMavenPom](http://docs.google.com/org.gradle.api.publish.maven.tasks.GenerateMavenPom.html)
* [GenerateBuildDashboard](http://docs.google.com/org.gradle.api.reporting.GenerateBuildDashboard.html)
* [GradleBuild](http://docs.google.com/org.gradle.api.tasks.GradleBuild.html)
* [GroovyCompile](http://docs.google.com/org.gradle.api.tasks.compile.GroovyCompile.html)
* [Groovydoc](http://docs.google.com/org.gradle.api.tasks.javadoc.Groovydoc.html)
* [HtmlDependencyReportTask](http://docs.google.com/org.gradle.api.reporting.dependencies.HtmlDependencyReportTask.html)
* [JacocoReport](http://docs.google.com/org.gradle.testing.jacoco.tasks.JacocoReport.html)
* [JacocoMerge](http://docs.google.com/org.gradle.testing.jacoco.tasks.JacocoMerge.html)
* [JacocoCoverageVerification](http://docs.google.com/org.gradle.testing.jacoco.tasks.JacocoCoverageVerification.html)
* [Jar](http://docs.google.com/org.gradle.api.tasks.bundling.Jar.html)
* [JavaCompile](http://docs.google.com/org.gradle.api.tasks.compile.JavaCompile.html)
* [Javadoc](http://docs.google.com/org.gradle.api.tasks.javadoc.Javadoc.html)
* [JavaExec](http://docs.google.com/org.gradle.api.tasks.JavaExec.html)
* [JDepend](http://docs.google.com/org.gradle.api.plugins.quality.JDepend.html)
* [Pmd](http://docs.google.com/org.gradle.api.plugins.quality.Pmd.html)
* [PublishToIvyRepository](http://docs.google.com/org.gradle.api.publish.ivy.tasks.PublishToIvyRepository.html)
* [PublishToMavenRepository](http://docs.google.com/org.gradle.api.publish.maven.tasks.PublishToMavenRepository.html)
* [ScalaCompile](http://docs.google.com/org.gradle.api.tasks.scala.ScalaCompile.html)
* [ScalaDoc](http://docs.google.com/org.gradle.api.tasks.scala.ScalaDoc.html)
* [InitBuild](http://docs.google.com/org.gradle.buildinit.tasks.InitBuild.html)
* [Sign](http://docs.google.com/org.gradle.plugins.signing.Sign.html)
* [Sync](http://docs.google.com/org.gradle.api.tasks.Sync.html)
* [Tar](http://docs.google.com/org.gradle.api.tasks.bundling.Tar.html)
* [AbstractTestTask](http://docs.google.com/org.gradle.api.tasks.testing.AbstractTestTask.html)
* [Test](http://docs.google.com/org.gradle.api.tasks.testing.Test.html)
* [TestReport](http://docs.google.com/org.gradle.api.tasks.testing.TestReport.html)
* [Upload](http://docs.google.com/org.gradle.api.tasks.Upload.html)
* [War](http://docs.google.com/org.gradle.api.tasks.bundling.War.html)
* [Wrapper](http://docs.google.com/org.gradle.api.tasks.wrapper.Wrapper.html)
* [WriteProperties](http://docs.google.com/org.gradle.api.tasks.WriteProperties.html)
* [Zip](http://docs.google.com/org.gradle.api.tasks.bundling.Zip.html)

### Reporting types

* [CustomizableHtmlReport](http://docs.google.com/org.gradle.api.reporting.CustomizableHtmlReport.html)
* [SingleFileReport](http://docs.google.com/org.gradle.api.reporting.SingleFileReport.html)
* [DirectoryReport](http://docs.google.com/org.gradle.api.reporting.DirectoryReport.html)
* [FindBugsXmlReport](http://docs.google.com/org.gradle.api.plugins.quality.FindBugsXmlReport.html)
* [Report](http://docs.google.com/org.gradle.api.reporting.Report.html)
* [Reporting](http://docs.google.com/org.gradle.api.reporting.Reporting.html)
* [ReportContainer](http://docs.google.com/org.gradle.api.reporting.ReportContainer.html)
* [ReportingExtension](http://docs.google.com/org.gradle.api.reporting.ReportingExtension.html)

### Eclipse/IDEA model types

* [EclipseModel](http://docs.google.com/org.gradle.plugins.ide.eclipse.model.EclipseModel.html)
* [EclipseProject](http://docs.google.com/org.gradle.plugins.ide.eclipse.model.EclipseProject.html)
* [EclipseClasspath](http://docs.google.com/org.gradle.plugins.ide.eclipse.model.EclipseClasspath.html)
* [EclipseJdt](http://docs.google.com/org.gradle.plugins.ide.eclipse.model.EclipseJdt.html)
* [EclipseWtp](http://docs.google.com/org.gradle.plugins.ide.eclipse.model.EclipseWtp.html)
* [EclipseWtpComponent](http://docs.google.com/org.gradle.plugins.ide.eclipse.model.EclipseWtpComponent.html)
* [EclipseWtpFacet](http://docs.google.com/org.gradle.plugins.ide.eclipse.model.EclipseWtpFacet.html)
* [IdeaModel](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModel.html)
* [IdeaProject](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaProject.html)
* [IdeaModule](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html)
* [IdeaWorkspace](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaWorkspace.html)
* [XmlFileContentMerger](http://docs.google.com/org.gradle.plugins.ide.api.XmlFileContentMerger.html)
* [FileContentMerger](http://docs.google.com/org.gradle.plugins.ide.api.FileContentMerger.html)

### Eclipse/IDEA task types

* [GenerateEclipseProject](http://docs.google.com/org.gradle.plugins.ide.eclipse.GenerateEclipseProject.html)
* [GenerateEclipseClasspath](http://docs.google.com/org.gradle.plugins.ide.eclipse.GenerateEclipseClasspath.html)
* [GenerateEclipseJdt](http://docs.google.com/org.gradle.plugins.ide.eclipse.GenerateEclipseJdt.html)
* [GenerateEclipseWtpComponent](http://docs.google.com/org.gradle.plugins.ide.eclipse.GenerateEclipseWtpComponent.html)
* [GenerateEclipseWtpFacet](http://docs.google.com/org.gradle.plugins.ide.eclipse.GenerateEclipseWtpFacet.html)
* [GenerateIdeaModule](http://docs.google.com/org.gradle.plugins.ide.idea.GenerateIdeaModule.html)
* [GenerateIdeaProject](http://docs.google.com/org.gradle.plugins.ide.idea.GenerateIdeaProject.html)
* [GenerateIdeaWorkspace](http://docs.google.com/org.gradle.plugins.ide.idea.GenerateIdeaWorkspace.html)

### Native software types

* [PrebuiltLibrary](http://docs.google.com/org.gradle.nativeplatform.PrebuiltLibrary.html)
* [PrebuiltSharedLibraryBinary](http://docs.google.com/org.gradle.nativeplatform.PrebuiltSharedLibraryBinary.html)
* [PrebuiltStaticLibraryBinary](http://docs.google.com/org.gradle.nativeplatform.PrebuiltStaticLibraryBinary.html)
* [NativeComponentSpec](http://docs.google.com/org.gradle.nativeplatform.NativeComponentSpec.html)
* [NativeExecutableSpec](http://docs.google.com/org.gradle.nativeplatform.NativeExecutableSpec.html)
* [NativeLibrarySpec](http://docs.google.com/org.gradle.nativeplatform.NativeLibrarySpec.html)
* [NativeTestSuiteSpec](http://docs.google.com/org.gradle.nativeplatform.test.NativeTestSuiteSpec.html)
* [CUnitTestSuiteSpec](http://docs.google.com/org.gradle.nativeplatform.test.cunit.CUnitTestSuiteSpec.html)
* [GoogleTestTestSuiteSpec](http://docs.google.com/org.gradle.nativeplatform.test.googletest.GoogleTestTestSuiteSpec.html)
* [NativeBinarySpec](http://docs.google.com/org.gradle.nativeplatform.NativeBinarySpec.html)
* [NativeExecutableBinarySpec](http://docs.google.com/org.gradle.nativeplatform.NativeExecutableBinarySpec.html)
* [NativeLibraryBinarySpec](http://docs.google.com/org.gradle.nativeplatform.NativeLibraryBinarySpec.html)
* [SharedLibraryBinarySpec](http://docs.google.com/org.gradle.nativeplatform.SharedLibraryBinarySpec.html)
* [StaticLibraryBinarySpec](http://docs.google.com/org.gradle.nativeplatform.StaticLibraryBinarySpec.html)
* [NativeTestSuiteBinarySpec](http://docs.google.com/org.gradle.nativeplatform.test.NativeTestSuiteBinarySpec.html)
* [CUnitTestSuiteBinarySpec](http://docs.google.com/org.gradle.nativeplatform.test.cunit.CUnitTestSuiteBinarySpec.html)
* [GoogleTestTestSuiteBinarySpec](http://docs.google.com/org.gradle.nativeplatform.test.googletest.GoogleTestTestSuiteBinarySpec.html)
* [NativePlatform](http://docs.google.com/org.gradle.nativeplatform.platform.NativePlatform.html)
* [BuildType](http://docs.google.com/org.gradle.nativeplatform.BuildType.html)
* [Flavor](http://docs.google.com/org.gradle.nativeplatform.Flavor.html)
* [Gcc](http://docs.google.com/org.gradle.nativeplatform.toolchain.Gcc.html)
* [Clang](http://docs.google.com/org.gradle.nativeplatform.toolchain.Clang.html)
* [VisualCpp](http://docs.google.com/org.gradle.nativeplatform.toolchain.VisualCpp.html)
* [AssemblerSourceSet](http://docs.google.com/org.gradle.language.assembler.AssemblerSourceSet.html)
* [CSourceSet](http://docs.google.com/org.gradle.language.c.CSourceSet.html)
* [CppSourceSet](http://docs.google.com/org.gradle.language.cpp.CppSourceSet.html)
* [ObjectiveCSourceSet](http://docs.google.com/org.gradle.language.objectivec.ObjectiveCSourceSet.html)
* [ObjectiveCppSourceSet](http://docs.google.com/org.gradle.language.objectivecpp.ObjectiveCppSourceSet.html)
* [WindowsResourceSet](http://docs.google.com/org.gradle.language.rc.WindowsResourceSet.html)
* [VisualStudioProject](http://docs.google.com/org.gradle.ide.visualstudio.VisualStudioProject.html)
* [VisualStudioSolution](http://docs.google.com/org.gradle.ide.visualstudio.VisualStudioSolution.html)
* [NativeExecutable](http://docs.google.com/org.gradle.nativeplatform.NativeExecutable.html)
* [NativeLibrary](http://docs.google.com/org.gradle.nativeplatform.NativeLibrary.html)
* [NativeBinary](http://docs.google.com/org.gradle.nativeplatform.NativeBinary.html)
* [NativeExecutableBinary](http://docs.google.com/org.gradle.nativeplatform.NativeExecutableBinary.html)
* [SharedLibraryBinary](http://docs.google.com/org.gradle.nativeplatform.SharedLibraryBinary.html)
* [StaticLibraryBinary](http://docs.google.com/org.gradle.nativeplatform.StaticLibraryBinary.html)

### Native component task types

* [CppCompile](http://docs.google.com/org.gradle.language.cpp.tasks.CppCompile.html)
* [CCompile](http://docs.google.com/org.gradle.language.c.tasks.CCompile.html)
* [Assemble](http://docs.google.com/org.gradle.language.assembler.tasks.Assemble.html)
* [ObjectiveCCompile](http://docs.google.com/org.gradle.language.objectivec.tasks.ObjectiveCCompile.html)
* [ObjectiveCppCompile](http://docs.google.com/org.gradle.language.objectivecpp.tasks.ObjectiveCppCompile.html)
* [WindowsResourceCompile](http://docs.google.com/org.gradle.language.rc.tasks.WindowsResourceCompile.html)
* [LinkExecutable](http://docs.google.com/org.gradle.nativeplatform.tasks.LinkExecutable.html)
* [LinkSharedLibrary](http://docs.google.com/org.gradle.nativeplatform.tasks.LinkSharedLibrary.html)
* [CreateStaticLibrary](http://docs.google.com/org.gradle.nativeplatform.tasks.CreateStaticLibrary.html)
* [InstallExecutable](http://docs.google.com/org.gradle.nativeplatform.tasks.InstallExecutable.html)
* [RunTestExecutable](http://docs.google.com/org.gradle.nativeplatform.test.tasks.RunTestExecutable.html)

Settings

**Table of Contents**

[Assembling a Multi-Project Build](http://docs.google.com/org.gradle.api.initialization.Settings.html#N18CA9)[Using Settings in a Settings File](http://docs.google.com/org.gradle.api.initialization.Settings.html#N18CC0)[Properties](http://docs.google.com/org.gradle.api.initialization.Settings.html#N18CDB)[Methods](http://docs.google.com/org.gradle.api.initialization.Settings.html#N18D3F)[Script blocks](http://docs.google.com/org.gradle.api.initialization.Settings.html#N18DC7)[Property details](http://docs.google.com/org.gradle.api.initialization.Settings.html#N18DCC)[Method details](http://docs.google.com/org.gradle.api.initialization.Settings.html#N18E6E)

| API Documentation: | [Settings](http://docs.google.com/javadoc/org/gradle/api/initialization/Settings.html) |
| --- | --- |

Declares the configuration required to instantiate and configure the hierarchy of [Project](http://docs.google.com/dsl/org.gradle.api.Project.html) instances which are to participate in a build.

There is a one-to-one correspondence between a Settings instance and a settings.gradle settings file. Before Gradle assembles the projects for a build, it creates a Settings instance and executes the settings file against it.

Assembling a Multi-Project Build

One of the purposes of the Settings object is to allow you to declare the projects which are to be included in the build. You add projects to the build using the [Settings.include(java.lang.String[])](http://docs.google.com/dsl/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:include(java.lang.String%5B%5D)) method. There is always a root project included in a build. It is added automatically when the Settings object is created. The root project's name defaults to the name of the directory containing the settings file. The root project's project directory defaults to the directory containing the settings file.

When a project is included in the build, a [ProjectDescriptor](http://docs.google.com/javadoc/org/gradle/api/initialization/ProjectDescriptor.html) is created. You can use this descriptor to change the default values for several properties of the project.

Using Settings in a Settings File

Dynamic Properties

In addition to the properties of this interface, the Settings object makes some additional read-only properties available to the settings script. This includes properties from the following sources:

* Defined in the gradle.properties file located in the settings directory of the build.
* Defined the gradle.properties file located in the user's .gradle directory.
* Provided on the command-line using the -P option.

Properties

| Property | Description |
| --- | --- |
| [buildCache](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:buildCache) | The build cache configuration. |
| [gradle](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:gradle) | The [Gradle](http://docs.google.com/dsl/org.gradle.api.invocation.Gradle.html) instance for the current build. |
| [pluginManager](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:pluginManager) | Incubating  The plugin manager for this plugin aware object. |
| [plugins](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:plugins) | The container of plugins that have been applied to this object. |
| [rootDir](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:rootDir) | The root directory of the build. The root directory is the project directory of the root project. |
| [rootProject](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:rootProject) | The root project of the build. |
| [settings](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:settings) | Returns this settings object. |
| [settingsDir](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:settingsDir) | The settings directory of the build. The settings directory is the directory containing the settings file. |
| [startParameter](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:startParameter) | The set of parameters used to invoke this instance of Gradle. |

Methods

| Method | Description |
| --- | --- |
| [apply](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:apply(groovy.lang.Closure))(closure) | Applies zero or more plugins or scripts. |
| [apply](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:apply(java.util.Map))(options) | Applies a plugin or script, using the given options provided as a map. Does nothing if the plugin has already been applied. |
| [apply](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:apply(org.gradle.api.Action))(action) | Applies zero or more plugins or scripts. |
| [buildCache](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:buildCache(org.gradle.api.Action))(action) | Configures build cache. |
| [findProject](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:findProject(java.io.File))(projectDir) | Returns the project with the given project directory. |
| [findProject](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:findProject(java.lang.String))(path) | Returns the project with the given path. |
| [include](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:include(java.lang.String%5B%5D))(projectPaths) | Adds the given projects to the build. Each path in the supplied list is treated as the path of a project to add to the build. Note that these path are not file paths, but instead specify the location of the new project in the project hierarchy. As such, the supplied paths must use the ':' character as separator (and NOT '/'). |
| [includeBuild](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:includeBuild(java.lang.Object))(rootProject) | Incubating  Includes a build at the specified path to the composite build. |
| [includeBuild](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:includeBuild(java.lang.Object,%20org.gradle.api.Action))(rootProject, configuration) | Incubating  Includes a build at the specified path to the composite build, with the supplied configuration. |
| [includeFlat](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:includeFlat(java.lang.String%5B%5D))(projectNames) | Adds the given projects to the build. Each name in the supplied list is treated as the name of a project to add to the build. |
| [project](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:project(java.io.File))(projectDir) | Returns the project with the given project directory. |
| [project](http://docs.google.com/org.gradle.api.initialization.Settings.html#org.gradle.api.initialization.Settings:project(java.lang.String))(path) | Returns the project with the given path. |

Script blocks

No script blocks

Property details

[**BuildCacheConfiguration**](http://docs.google.com/dsl/org.gradle.caching.configuration.BuildCacheConfiguration.html) buildCache (read-only)

The build cache configuration.

[**Gradle**](http://docs.google.com/dsl/org.gradle.api.invocation.Gradle.html) gradle (read-only)

The [Gradle](http://docs.google.com/dsl/org.gradle.api.invocation.Gradle.html) instance for the current build.

[**PluginManager**](http://docs.google.com/dsl/org.gradle.api.plugins.PluginManager.html) pluginManager (read-only)

Note: This property is [incubating](http://docs.google.com/userguide/feature_lifecycle.html) and may change in a future version of Gradle.

The plugin manager for this plugin aware object.

[**PluginContainer**](http://docs.google.com/javadoc/org/gradle/api/plugins/PluginContainer.html) plugins (read-only)

The container of plugins that have been applied to this object.

While not deprecated, it is preferred to use the methods of this interface or the [PluginAware.getPluginManager()](http://docs.google.com/dsl/org.gradle.api.plugins.PluginAware.html#org.gradle.api.plugins.PluginAware:pluginManager) than use the plugin container.

Use one of the 'apply' methods on this interface or on the [PluginAware.getPluginManager()](http://docs.google.com/dsl/org.gradle.api.plugins.PluginAware.html#org.gradle.api.plugins.PluginAware:pluginManager) to apply plugins instead of applying via the plugin container.

Use [PluginManager.hasPlugin(java.lang.String)](http://docs.google.com/dsl/org.gradle.api.plugins.PluginManager.html#org.gradle.api.plugins.PluginManager:hasPlugin(java.lang.String)) or similar to query for the application of plugins instead of doing so via the plugin container.

[**File**](http://download.oracle.com/javase/8/docs/api/java/io/File.html) rootDir (read-only)

The root directory of the build. The root directory is the project directory of the root project.

[**ProjectDescriptor**](http://docs.google.com/javadoc/org/gradle/api/initialization/ProjectDescriptor.html) rootProject (read-only)

The root project of the build.

[**Settings**](http://docs.google.com/dsl/org.gradle.api.initialization.Settings.html) settings (read-only)

Returns this settings object.

[**File**](http://download.oracle.com/javase/8/docs/api/java/io/File.html) settingsDir (read-only)

The settings directory of the build. The settings directory is the directory containing the settings file.

[**StartParameter**](http://docs.google.com/javadoc/org/gradle/StartParameter.html) startParameter (read-only)

The set of parameters used to invoke this instance of Gradle.

Method details

void apply([**Closure**](http://docs.groovy-lang.org/2.4.15/html/gapi/groovy/lang/Closure.html) closure)

Applies zero or more plugins or scripts.

The given closure is used to configure an [ObjectConfigurationAction](http://docs.google.com/javadoc/org/gradle/api/plugins/ObjectConfigurationAction.html), which “builds” the plugin application.

This method differs from [PluginAware.apply(java.util.Map)](http://docs.google.com/dsl/org.gradle.api.plugins.PluginAware.html#org.gradle.api.plugins.PluginAware:apply(java.util.Map)) in that it allows methods of the configuration action to be invoked more than once.

void apply([**Map**](http://download.oracle.com/javase/8/docs/api/java/util/Map.html)<[**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html), ?> options)

Applies a plugin or script, using the given options provided as a map. Does nothing if the plugin has already been applied.

The given map is applied as a series of method calls to a newly created [ObjectConfigurationAction](http://docs.google.com/javadoc/org/gradle/api/plugins/ObjectConfigurationAction.html). That is, each key in the map is expected to be the name of a method [ObjectConfigurationAction](http://docs.google.com/javadoc/org/gradle/api/plugins/ObjectConfigurationAction.html) and the value to be compatible arguments to that method.

The following options are available:

* from: A script to apply. Accepts any path supported by [Project.uri(java.lang.Object)](http://docs.google.com/dsl/org.gradle.api.Project.html#org.gradle.api.Project:uri(java.lang.Object)).
* plugin: The id or implementation class of the plugin to apply.
* to: The target delegate object or objects. The default is this plugin aware object. Use this to configure objects other than this object.

void apply([**Action**](http://docs.google.com/javadoc/org/gradle/api/Action.html)<? super [**ObjectConfigurationAction**](http://docs.google.com/javadoc/org/gradle/api/plugins/ObjectConfigurationAction.html)> action)

Applies zero or more plugins or scripts.

The given closure is used to configure an [ObjectConfigurationAction](http://docs.google.com/javadoc/org/gradle/api/plugins/ObjectConfigurationAction.html), which “builds” the plugin application.

This method differs from [PluginAware.apply(java.util.Map)](http://docs.google.com/dsl/org.gradle.api.plugins.PluginAware.html#org.gradle.api.plugins.PluginAware:apply(java.util.Map)) in that it allows methods of the configuration action to be invoked more than once.

void buildCache([**Action**](http://docs.google.com/javadoc/org/gradle/api/Action.html)<? super [**BuildCacheConfiguration**](http://docs.google.com/dsl/org.gradle.caching.configuration.BuildCacheConfiguration.html)> action)

Configures build cache.

[**ProjectDescriptor**](http://docs.google.com/javadoc/org/gradle/api/initialization/ProjectDescriptor.html) findProject([**File**](http://download.oracle.com/javase/8/docs/api/java/io/File.html) projectDir)

Returns the project with the given project directory.

[**ProjectDescriptor**](http://docs.google.com/javadoc/org/gradle/api/initialization/ProjectDescriptor.html) findProject([**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html) path)

Returns the project with the given path.

void include([**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html)... projectPaths)

Adds the given projects to the build. Each path in the supplied list is treated as the path of a project to add to the build. Note that these path are not file paths, but instead specify the location of the new project in the project hierarchy. As such, the supplied paths must use the ':' character as separator (and NOT '/').

The last element of the supplied path is used as the project name. The supplied path is converted to a project directory relative to the root project directory. The project directory can be altered by changing the 'projectDir' property after the project has been included (see [ProjectDescriptor.setProjectDir(java.io.File)](http://docs.google.com/javadoc/org/gradle/api/initialization/ProjectDescriptor.html#setProjectDir-java.io.File-))

As an example, the path a:b adds a project with path :a:b, name b and project directory $rootDir/a/b. It also adds the a project with path :a, name a and project directory $rootDir/a, if it does not exist already.

Some common examples of using the project path are:

// include two projects, 'foo' and 'foo:bar'  
// directories are inferred by replacing ':' with '/'  
include 'foo:bar'  
  
// include one project whose project dir does not match the logical project path  
include 'baz'  
project(':baz').projectDir = file('foo/baz')  
  
// include many projects whose project dirs do not match the logical project paths  
file('subprojects').eachDir { dir ->  
 include dir.name  
 project(":${dir.name}").projectDir = dir  
}

void includeBuild([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html) rootProject)

Note: This method is [incubating](http://docs.google.com/userguide/feature_lifecycle.html) and may change in a future version of Gradle.

Includes a build at the specified path to the composite build.

void includeBuild([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html) rootProject, [**Action**](http://docs.google.com/javadoc/org/gradle/api/Action.html)<[**ConfigurableIncludedBuild**](http://docs.google.com/javadoc/org/gradle/api/initialization/ConfigurableIncludedBuild.html)> configuration)

Note: This method is [incubating](http://docs.google.com/userguide/feature_lifecycle.html) and may change in a future version of Gradle.

Includes a build at the specified path to the composite build, with the supplied configuration.

void includeFlat([**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html)... projectNames)

Adds the given projects to the build. Each name in the supplied list is treated as the name of a project to add to the build.

The supplied name is converted to a project directory relative to the *parent* directory of the root project directory.

As an example, the name a add a project with path :a, name a and project directory $rootDir/../a.

[**ProjectDescriptor**](http://docs.google.com/javadoc/org/gradle/api/initialization/ProjectDescriptor.html) project([**File**](http://download.oracle.com/javase/8/docs/api/java/io/File.html) projectDir)

Returns the project with the given project directory.

[**ProjectDescriptor**](http://docs.google.com/javadoc/org/gradle/api/initialization/ProjectDescriptor.html) project([**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html) path)

Returns the project with the given path.

**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/)