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.Script.html#N19203)
  + [Methods](http://docs.google.com/org.gradle.api.Script.html#N19238)
  + [Script blocks](http://docs.google.com/org.gradle.api.Script.html#N19377)

### 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)

Script

**Table of Contents**

[Properties](http://docs.google.com/org.gradle.api.Script.html#N19203)[Methods](http://docs.google.com/org.gradle.api.Script.html#N19238)[Script blocks](http://docs.google.com/org.gradle.api.Script.html#N19377)[Property details](http://docs.google.com/org.gradle.api.Script.html#N1938C)[Method details](http://docs.google.com/org.gradle.api.Script.html#N193CC)[Script block details](http://docs.google.com/org.gradle.api.Script.html#N1963B)

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

This interface is implemented by all Gradle scripts to add in some Gradle-specific methods. As your compiled script class will implement this interface, you can use the methods and properties declared by this interface directly in your script.

Generally, a Script object will have a delegate object attached to it. For example, a build script will have a [Project](http://docs.google.com/dsl/org.gradle.api.Project.html) instance attached to it, and an initialization script will have a [Gradle](http://docs.google.com/dsl/org.gradle.api.invocation.Gradle.html) instance attached to it. Any property reference or method call which is not found on this Script object is forwarded to the delegate object.

Properties

| Property | Description |
| --- | --- |
| [buildscript](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:buildscript) | The script handler for this script. You can use this handler to manage the classpath used to compile and execute this script. |
| [logger](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:logger) | The logger for this script. You can use this in your script to write log messages. |
| [logging](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:logging) | The [LoggingManager](http://docs.google.com/javadoc/org/gradle/api/logging/LoggingManager.html) which can be used to receive logging and to control the standard output/error capture for this script. By default, System.out is redirected to the Gradle logging system at the QUIET log level, and System.err is redirected at the ERROR log level. |
| [resources](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:resources) | Provides access to resource-specific utility methods, for example factory methods that create various resources. |

Methods

| Method | Description |
| --- | --- |
| [apply](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:apply(groovy.lang.Closure))(closure) | Configures the delegate object for this script using plugins or scripts. |
| [apply](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:apply(java.util.Map))(options) | Configures the delegate object for this script using plugins or scripts. |
| [copy](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:copy(groovy.lang.Closure))(closure) | Copy the specified files. The given closure is used to configure a [CopySpec](http://docs.google.com/javadoc/org/gradle/api/file/CopySpec.html), which is then used to copy the files. Example: |
| [copySpec](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:copySpec(groovy.lang.Closure))(closure) | Creates a [CopySpec](http://docs.google.com/javadoc/org/gradle/api/file/CopySpec.html) which can later be used to copy files or create an archive. The given closure is used to configure the [CopySpec](http://docs.google.com/javadoc/org/gradle/api/file/CopySpec.html) before it is returned by this method. |
| [delete](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:delete(java.lang.Object%5B%5D))(paths) | Deletes files and directories. |
| [exec](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:exec(groovy.lang.Closure))(closure) | Executes an external command. The closure configures a [ExecSpec](http://docs.google.com/javadoc/org/gradle/process/ExecSpec.html). |
| [exec](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:exec(org.gradle.api.Action))(action) | Executes an external command. |
| [file](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:file(java.lang.Object))(path) | Resolves a file path relative to the directory containing this script. This works as described for [Project.file(java.lang.Object)](http://docs.google.com/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object)) |
| [file](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:file(java.lang.Object,%20org.gradle.api.PathValidation))(path, validation) | Resolves a file path relative to the directory containing this script and validates it using the given scheme. See [PathValidation](http://docs.google.com/javadoc/org/gradle/api/PathValidation.html) for the list of possible validations. |
| [fileTree](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:fileTree(java.lang.Object))(baseDir) | Creates a new ConfigurableFileTree using the given base directory. The given baseDir path is evaluated as per [Script.file(java.lang.Object)](http://docs.google.com/dsl/org.gradle.api.Script.html#org.gradle.api.Script:file(java.lang.Object)). |
| [fileTree](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:fileTree(java.lang.Object,%20groovy.lang.Closure))(baseDir, configureClosure) | Creates a new ConfigurableFileTree using the given base directory. The given baseDir path is evaluated as per [Script.file(java.lang.Object)](http://docs.google.com/dsl/org.gradle.api.Script.html#org.gradle.api.Script:file(java.lang.Object)). The closure will be used to configure the new file tree. The file tree is passed to the closure as its delegate. Example: |
| [fileTree](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:fileTree(java.util.Map))(args) | Creates a new ConfigurableFileTree using the provided map of arguments. The map will be applied as properties on the new file tree. Example: |
| [files](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:files(java.lang.Object,%20groovy.lang.Closure))(paths, configureClosure) | Creates a new ConfigurableFileCollection using the given paths. The file collection is configured using the given closure. This method works as described for [Project.files(java.lang.Object, groovy.lang.Closure)](http://docs.google.com/dsl/org.gradle.api.Project.html#org.gradle.api.Project:files(java.lang.Object,%20groovy.lang.Closure)). Relative paths are resolved relative to the directory containing this script. |
| [files](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:files(java.lang.Object%5B%5D))(paths) | Returns a [ConfigurableFileCollection](http://docs.google.com/javadoc/org/gradle/api/file/ConfigurableFileCollection.html) containing the given files. This works as described for [Project.files(java.lang.Object[])](http://docs.google.com/dsl/org.gradle.api.Project.html#org.gradle.api.Project:files(java.lang.Object%5B%5D)). Relative paths are resolved relative to the directory containing this script. |
| [javaexec](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:javaexec(groovy.lang.Closure))(closure) | Executes a Java main class. The closure configures a [JavaExecSpec](http://docs.google.com/javadoc/org/gradle/process/JavaExecSpec.html). |
| [javaexec](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:javaexec(org.gradle.api.Action))(action) | Executes a Java main class. |
| [mkdir](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:mkdir(java.lang.Object))(path) | Creates a directory and returns a file pointing to it. |
| [relativePath](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:relativePath(java.lang.Object))(path) | Returns the relative path from the directory containing this script to the given path. The given path object is (logically) resolved as described for [Script.file(java.lang.Object)](http://docs.google.com/dsl/org.gradle.api.Script.html#org.gradle.api.Script:file(java.lang.Object)), from which a relative path is calculated. |
| [tarTree](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:tarTree(java.lang.Object))(tarPath) | Creates a new FileTree which contains the contents of the given TAR file. The given tarPath path can be: |
| [uri](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:uri(java.lang.Object))(path) | Resolves a file path to a URI, relative to the directory containing this script. Evaluates the provided path object as described for [Script.file(java.lang.Object)](http://docs.google.com/dsl/org.gradle.api.Script.html#org.gradle.api.Script:file(java.lang.Object)), with the exception that any URI scheme is supported, not just 'file:' URIs. |
| [zipTree](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:zipTree(java.lang.Object))(zipPath) | Creates a new FileTree which contains the contents of the given ZIP file. The given zipPath path is evaluated as per [Script.file(java.lang.Object)](http://docs.google.com/dsl/org.gradle.api.Script.html#org.gradle.api.Script:file(java.lang.Object)). You can combine this method with the [Script.copy(groovy.lang.Closure)](http://docs.google.com/dsl/org.gradle.api.Script.html#org.gradle.api.Script:copy(groovy.lang.Closure)) method to unzip a ZIP file. |

Script blocks

| Block | Description |
| --- | --- |
| [buildscript](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:buildscript(groovy.lang.Closure)) | Configures the classpath for this script. |

Property details

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

The script handler for this script. You can use this handler to manage the classpath used to compile and execute this script.

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

The logger for this script. You can use this in your script to write log messages.

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

The [LoggingManager](http://docs.google.com/javadoc/org/gradle/api/logging/LoggingManager.html) which can be used to receive logging and to control the standard output/error capture for this script. By default, System.out is redirected to the Gradle logging system at the QUIET log level, and System.err is redirected at the ERROR log level.

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

Provides access to resource-specific utility methods, for example factory methods that create various resources.

Method details

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

Configures the delegate object for this script using plugins or scripts.

The given closure is used to configure an [ObjectConfigurationAction](http://docs.google.com/javadoc/org/gradle/api/plugins/ObjectConfigurationAction.html) which is then used to configure the delegate object.

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)

Configures the delegate object for this script using plugins or scripts.

The following options are available:

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

For more detail, see [ObjectConfigurationAction](http://docs.google.com/javadoc/org/gradle/api/plugins/ObjectConfigurationAction.html).

[**WorkResult**](http://docs.google.com/javadoc/org/gradle/api/tasks/WorkResult.html) copy([**Closure**](http://docs.groovy-lang.org/2.4.15/html/gapi/groovy/lang/Closure.html) closure)

Copy the specified files. The given closure is used to configure a [CopySpec](http://docs.google.com/javadoc/org/gradle/api/file/CopySpec.html), which is then used to copy the files. Example:

copy {  
 from configurations.runtime  
 into 'build/deploy/lib'  
}

Note that CopySpecs can be nested:

copy {  
 into 'build/webroot'  
 exclude '\*\*/.svn/\*\*'  
 from('src/main/webapp') {  
 include '\*\*/\*.jsp'  
 filter(ReplaceTokens, tokens:[copyright:'2009', version:'2.3.1'])  
 }  
 from('src/main/js') {  
 include '\*\*/\*.js'  
 }  
}

[**CopySpec**](http://docs.google.com/javadoc/org/gradle/api/file/CopySpec.html) copySpec([**Closure**](http://docs.groovy-lang.org/2.4.15/html/gapi/groovy/lang/Closure.html) closure)

Creates a [CopySpec](http://docs.google.com/javadoc/org/gradle/api/file/CopySpec.html) which can later be used to copy files or create an archive. The given closure is used to configure the [CopySpec](http://docs.google.com/javadoc/org/gradle/api/file/CopySpec.html) before it is returned by this method.

boolean delete([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html)... paths)

Deletes files and directories.

[**ExecResult**](http://docs.google.com/javadoc/org/gradle/process/ExecResult.html) exec([**Closure**](http://docs.groovy-lang.org/2.4.15/html/gapi/groovy/lang/Closure.html) closure)

Executes an external command. The closure configures a [ExecSpec](http://docs.google.com/javadoc/org/gradle/process/ExecSpec.html).

[**ExecResult**](http://docs.google.com/javadoc/org/gradle/process/ExecResult.html) exec([**Action**](http://docs.google.com/javadoc/org/gradle/api/Action.html)<? super [**ExecSpec**](http://docs.google.com/javadoc/org/gradle/process/ExecSpec.html)> action)

Executes an external command.

[**File**](http://download.oracle.com/javase/8/docs/api/java/io/File.html) file([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html) path)

Resolves a file path relative to the directory containing this script. This works as described for [Project.file(java.lang.Object)](http://docs.google.com/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object))

[**File**](http://download.oracle.com/javase/8/docs/api/java/io/File.html) file([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html) path, [**PathValidation**](http://docs.google.com/javadoc/org/gradle/api/PathValidation.html) validation)

Resolves a file path relative to the directory containing this script and validates it using the given scheme. See [PathValidation](http://docs.google.com/javadoc/org/gradle/api/PathValidation.html) for the list of possible validations.

[**ConfigurableFileTree**](http://docs.google.com/javadoc/org/gradle/api/file/ConfigurableFileTree.html) fileTree([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html) baseDir)

Creates a new ConfigurableFileTree using the given base directory. The given baseDir path is evaluated as per [Script.file(java.lang.Object)](http://docs.google.com/dsl/org.gradle.api.Script.html#org.gradle.api.Script:file(java.lang.Object)).

The returned file tree is lazy, so that it scans for files only when the contents of the file tree are queried. The file tree is also live, so that it scans for files each time the contents of the file tree are queried.

[**ConfigurableFileTree**](http://docs.google.com/javadoc/org/gradle/api/file/ConfigurableFileTree.html) fileTree([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html) baseDir, [**Closure**](http://docs.groovy-lang.org/2.4.15/html/gapi/groovy/lang/Closure.html) configureClosure)

Creates a new ConfigurableFileTree using the given base directory. The given baseDir path is evaluated as per [Script.file(java.lang.Object)](http://docs.google.com/dsl/org.gradle.api.Script.html#org.gradle.api.Script:file(java.lang.Object)). The closure will be used to configure the new file tree. The file tree is passed to the closure as its delegate. Example:

fileTree('src') {  
 exclude '\*\*/.svn/\*\*'  
}.copy { into 'dest'}

The returned file tree is lazy, so that it scans for files only when the contents of the file tree are queried. The file tree is also live, so that it scans for files each time the contents of the file tree are queried.

[**ConfigurableFileTree**](http://docs.google.com/javadoc/org/gradle/api/file/ConfigurableFileTree.html) fileTree([**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), ?> args)

Creates a new ConfigurableFileTree using the provided map of arguments. The map will be applied as properties on the new file tree. Example:

fileTree(dir:'src', excludes:['\*\*/ignore/\*\*','\*\*/.svn/\*\*'])

The returned file tree is lazy, so that it scans for files only when the contents of the file tree are queried. The file tree is also live, so that it scans for files each time the contents of the file tree are queried.

[**ConfigurableFileCollection**](http://docs.google.com/javadoc/org/gradle/api/file/ConfigurableFileCollection.html) files([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html) paths, [**Closure**](http://docs.groovy-lang.org/2.4.15/html/gapi/groovy/lang/Closure.html) configureClosure)

Creates a new ConfigurableFileCollection using the given paths. The file collection is configured using the given closure. This method works as described for [Project.files(java.lang.Object, groovy.lang.Closure)](http://docs.google.com/dsl/org.gradle.api.Project.html#org.gradle.api.Project:files(java.lang.Object,%20groovy.lang.Closure)). Relative paths are resolved relative to the directory containing this script.

[**ConfigurableFileCollection**](http://docs.google.com/javadoc/org/gradle/api/file/ConfigurableFileCollection.html) files([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html)... paths)

Returns a [ConfigurableFileCollection](http://docs.google.com/javadoc/org/gradle/api/file/ConfigurableFileCollection.html) containing the given files. This works as described for [Project.files(java.lang.Object[])](http://docs.google.com/dsl/org.gradle.api.Project.html#org.gradle.api.Project:files(java.lang.Object%5B%5D)). Relative paths are resolved relative to the directory containing this script.

[**ExecResult**](http://docs.google.com/javadoc/org/gradle/process/ExecResult.html) javaexec([**Closure**](http://docs.groovy-lang.org/2.4.15/html/gapi/groovy/lang/Closure.html) closure)

Executes a Java main class. The closure configures a [JavaExecSpec](http://docs.google.com/javadoc/org/gradle/process/JavaExecSpec.html).

[**ExecResult**](http://docs.google.com/javadoc/org/gradle/process/ExecResult.html) javaexec([**Action**](http://docs.google.com/javadoc/org/gradle/api/Action.html)<? super [**JavaExecSpec**](http://docs.google.com/javadoc/org/gradle/process/JavaExecSpec.html)> action)

Executes a Java main class.

[**File**](http://download.oracle.com/javase/8/docs/api/java/io/File.html) mkdir([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html) path)

Creates a directory and returns a file pointing to it.

[**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html) relativePath([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html) path)

Returns the relative path from the directory containing this script to the given path. The given path object is (logically) resolved as described for [Script.file(java.lang.Object)](http://docs.google.com/dsl/org.gradle.api.Script.html#org.gradle.api.Script:file(java.lang.Object)), from which a relative path is calculated.

[**FileTree**](http://docs.google.com/javadoc/org/gradle/api/file/FileTree.html) tarTree([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html) tarPath)

Creates a new FileTree which contains the contents of the given TAR file. The given tarPath path can be:

* an instance of [Resource](http://docs.google.com/javadoc/org/gradle/api/resources/Resource.html)
* any other object is evaluated as per [Script.file(java.lang.Object)](http://docs.google.com/dsl/org.gradle.api.Script.html#org.gradle.api.Script:file(java.lang.Object))

The returned file tree is lazy, so that it scans for files only when the contents of the file tree are queried. The file tree is also live, so that it scans for files each time the contents of the file tree are queried.

Unless custom implementation of resources is passed, the tar tree attempts to guess the compression based on the file extension.

You can combine this method with the [Script.copy(groovy.lang.Closure)](http://docs.google.com/dsl/org.gradle.api.Script.html#org.gradle.api.Script:copy(groovy.lang.Closure)) method to untar a TAR file:

task untar(type: Copy) {  
 from tarTree('someCompressedTar.gzip')  
  
 //tar tree attempts to guess the compression based on the file extension  
 //however if you must specify the compression explicitly you can:  
 from tarTree(resources.gzip('someTar.ext'))  
  
 //in case you work with unconventionally compressed tars  
 //you can provide your own implementation of a ReadableResource:  
 //from tarTree(yourOwnResource as ReadableResource)  
  
 into 'dest'  
}

[**URI**](http://download.oracle.com/javase/8/docs/api/java/net/URI.html) uri([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html) path)

Resolves a file path to a URI, relative to the directory containing this script. Evaluates the provided path object as described for [Script.file(java.lang.Object)](http://docs.google.com/dsl/org.gradle.api.Script.html#org.gradle.api.Script:file(java.lang.Object)), with the exception that any URI scheme is supported, not just 'file:' URIs.

[**FileTree**](http://docs.google.com/javadoc/org/gradle/api/file/FileTree.html) zipTree([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html) zipPath)

Creates a new FileTree which contains the contents of the given ZIP file. The given zipPath path is evaluated as per [Script.file(java.lang.Object)](http://docs.google.com/dsl/org.gradle.api.Script.html#org.gradle.api.Script:file(java.lang.Object)). You can combine this method with the [Script.copy(groovy.lang.Closure)](http://docs.google.com/dsl/org.gradle.api.Script.html#org.gradle.api.Script:copy(groovy.lang.Closure)) method to unzip a ZIP file.

The returned file tree is lazy, so that it scans for files only when the contents of the file tree are queried. The file tree is also live, so that it scans for files each time the contents of the file tree are queried.

Script block details

buildscript { }

Configures the classpath for this script.

The given closure is executed against this script's [ScriptHandler](http://docs.google.com/javadoc/org/gradle/api/initialization/dsl/ScriptHandler.html). The [ScriptHandler](http://docs.google.com/javadoc/org/gradle/api/initialization/dsl/ScriptHandler.html) is passed to the closure as the closure's delegate.

Delegates to:[ScriptHandler](http://docs.google.com/javadoc/org/gradle/api/initialization/dsl/ScriptHandler.html) from [buildscript](http://docs.google.com/org.gradle.api.Script.html#org.gradle.api.Script:buildscript)

**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/)