# Index

* **Home**
  + [Introduction](https://docs.gradle.org/4.1/dsl/index.html#N1000C) [local-Introduction](#_Introduction)
  + [Some basics](https://docs.gradle.org/4.1/dsl/index.html#N10011)
  + [Build script structure](https://docs.gradle.org/4.1/dsl/index.html#N10060)
  + [Core types](https://docs.gradle.org/4.1/dsl/index.html#N100CA)
  + [Container types](https://docs.gradle.org/4.1/dsl/index.html#N102A4)
  + [Build Cache types](https://docs.gradle.org/4.1/dsl/index.html#N102EA)
  + [Input Normalization types](https://docs.gradle.org/4.1/dsl/index.html#N10318)
  + [Help Task types](https://docs.gradle.org/4.1/dsl/index.html#N10342)
  + [Task types](https://docs.gradle.org/4.1/dsl/index.html#N103A8)
  + [Reporting types](https://docs.gradle.org/4.1/dsl/index.html#N1055B)
  + [Eclipse/IDEA model types](https://docs.gradle.org/4.1/dsl/index.html#N105B7)
  + [Eclipse/IDEA task types](https://docs.gradle.org/4.1/dsl/index.html#N10640)
  + [Native software model types](https://docs.gradle.org/4.1/dsl/index.html#N1069D)
  + [Native binary task types](https://docs.gradle.org/4.1/dsl/index.html#N1082E)
* **Build script blocks -------------------------------** [local](#_Script_block_details)
* [allprojects { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:allprojects(groovy.lang.Closure)) local
* [artifacts { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:artifacts(groovy.lang.Closure))
* [buildscript { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:buildscript(groovy.lang.Closure))
* [configurations { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:configurations(groovy.lang.Closure))
* [dependencies { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:dependencies(groovy.lang.Closure))
* [repositories { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:repositories(groovy.lang.Closure))
* [sourceSets { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:sourceSets(groovy.lang.Closure))
* [subprojects { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:subprojects(groovy.lang.Closure))
* [publishing { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:publishing(groovy.lang.Closure))
* **Core types**
* [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html)
* [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html) ------------------------------------------------ [local](#_Task)
* [Gradle](https://docs.gradle.org/4.1/dsl/org.gradle.api.invocation.Gradle.html)
* [Settings](https://docs.gradle.org/4.1/dsl/org.gradle.api.initialization.Settings.html)
* [IncludedBuild](https://docs.gradle.org/4.1/dsl/org.gradle.api.initialization.IncludedBuild.html)
* [Script](https://docs.gradle.org/4.1/dsl/org.gradle.api.Script.html)
* [JavaToolChain](https://docs.gradle.org/4.1/dsl/org.gradle.jvm.toolchain.JavaToolChain.html)
* [SourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.SourceSet.html)
* [SourceSetOutput](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.SourceSetOutput.html)
* [SourceDirectorySet](https://docs.gradle.org/4.1/dsl/org.gradle.api.file.SourceDirectorySet.html)
* [IncrementalTaskInputs](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.incremental.IncrementalTaskInputs.html)
* [Configuration](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.Configuration.html)
* [ResolutionStrategy](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ResolutionStrategy.html)
* [ArtifactResolutionQuery](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.query.ArtifactResolutionQuery.html)
* [ComponentSelection](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ComponentSelection.html)
* [ComponentSelectionRules](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ComponentSelectionRules.html)
* [GradlePluginPortal](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.GradlePluginPortal.html)
* [ExtensionAware](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.ExtensionAware.html)
* [ExtraPropertiesExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.ExtraPropertiesExtension.html)
* [PublishingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.PublishingExtension.html)
* [IvyPublication](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyPublication.html)
* [IvyArtifact](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyArtifact.html)
* [IvyArtifactSet](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyArtifactSet.html)
* [IvyModuleDescriptorSpec](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyModuleDescriptorSpec.html)
* [IvyPluginRepository](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.IvyPluginRepository.html)
* [MavenPublication](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenPublication.html)
* [MavenArtifact](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenArtifact.html)
* [MavenArtifactSet](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenArtifactSet.html)
* [MavenPom](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenPom.html)
* [MavenPluginRepository](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.MavenPluginRepository.html)
* [PluginDependenciesSpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.use.PluginDependenciesSpec.html)
* [PluginDependencySpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.use.PluginDependencySpec.html)
* [PluginManagementSpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.management.PluginManagementSpec.html)
* [PluginRepositoriesSpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.PluginRepositoriesSpec.html)
* [PluginRepository](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.PluginRepository.html)
* [ResourceHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.resources.ResourceHandler.html)
* [TextResourceFactory](https://docs.gradle.org/4.1/dsl/org.gradle.api.resources.TextResourceFactory.html)
* **Container types**
* [ConfigurationContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ConfigurationContainer.html)
* [RepositoryHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.RepositoryHandler.html)
* [DependencyHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html)
* [ArtifactHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.ArtifactHandler.html)
* **Build Cache types**
* [BuildCacheConfiguration](https://docs.gradle.org/4.1/dsl/org.gradle.caching.configuration.BuildCacheConfiguration.html)
* [DirectoryBuildCache](https://docs.gradle.org/4.1/dsl/org.gradle.caching.local.DirectoryBuildCache.html)
* [HttpBuildCache](https://docs.gradle.org/4.1/dsl/org.gradle.caching.http.HttpBuildCache.html)
* **Input Normalization types**
* [InputNormalizationHandler](https://docs.gradle.org/4.1/dsl/org.gradle.normalization.InputNormalizationHandler.html)
* [InputNormalization](https://docs.gradle.org/4.1/dsl/org.gradle.normalization.InputNormalization.html)
* [RuntimeClasspathNormalization](https://docs.gradle.org/4.1/dsl/org.gradle.normalization.RuntimeClasspathNormalization.html)
* **Help Task types**
* [TaskReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.TaskReportTask.html)
* [ProjectReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.ProjectReportTask.html)
* [DependencyReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.DependencyReportTask.html)
* [DependencyInsightReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.DependencyInsightReportTask.html)
* [PropertyReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.PropertyReportTask.html)
* [ComponentReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.components.ComponentReport.html)
* [DependentComponentsReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.dependents.DependentComponentsReport.html)
* [ModelReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.model.ModelReport.html)
* **Task types**
* [AntlrTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.antlr.AntlrTask.html)
* [BuildEnvironmentReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.BuildEnvironmentReportTask.html)
* [Checkstyle](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.Checkstyle.html)
* [CodeNarc](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.CodeNarc.html)
* [CompareGradleBuilds](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.buildcomparison.gradle.CompareGradleBuilds.html)
* [Copy](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.Copy.html)
* [CreateStartScripts](https://docs.gradle.org/4.1/dsl/org.gradle.jvm.application.tasks.CreateStartScripts.html)
* [Delete](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.Delete.html)
* [Ear](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ear.Ear.html)
* [Exec](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.Exec.html)
* [FindBugs](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.FindBugs.html)
* [GenerateIvyDescriptor](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.tasks.GenerateIvyDescriptor.html)
* [GenerateMavenPom](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.tasks.GenerateMavenPom.html)
* [GenerateBuildDashboard](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.GenerateBuildDashboard.html)
* [GradleBuild](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.GradleBuild.html)
* [GroovyCompile](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.compile.GroovyCompile.html)
* [Groovydoc](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.javadoc.Groovydoc.html)
* [HtmlDependencyReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.dependencies.HtmlDependencyReportTask.html)
* [JacocoReport](https://docs.gradle.org/4.1/dsl/org.gradle.testing.jacoco.tasks.JacocoReport.html)
* [JacocoMerge](https://docs.gradle.org/4.1/dsl/org.gradle.testing.jacoco.tasks.JacocoMerge.html)
* [JacocoCoverageVerification](https://docs.gradle.org/4.1/dsl/org.gradle.testing.jacoco.tasks.JacocoCoverageVerification.html)
* [Jar](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.bundling.Jar.html)
* [JavaCompile](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.compile.JavaCompile.html)
* [Javadoc](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.javadoc.Javadoc.html)
* [JavaExec](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.JavaExec.html)
* [JDepend](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.JDepend.html)
* [Pmd](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.Pmd.html)
* [PublishToIvyRepository](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.tasks.PublishToIvyRepository.html)
* [PublishToMavenRepository](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.tasks.PublishToMavenRepository.html)
* [ScalaCompile](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.scala.ScalaCompile.html)
* [ScalaDoc](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.scala.ScalaDoc.html)
* [InitBuild](https://docs.gradle.org/4.1/dsl/org.gradle.buildinit.tasks.InitBuild.html)
* [Sign](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.signing.Sign.html)
* [Sync](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.Sync.html)
* [Tar](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.bundling.Tar.html)
* [Test](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.testing.Test.html)
* [TestReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.testing.TestReport.html)
* [Upload](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.Upload.html)
* [War](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.bundling.War.html)
* [Wrapper](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.wrapper.Wrapper.html)
* [WriteProperties](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.WriteProperties.html)
* [Zip](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.bundling.Zip.html)
* **Reporting types**
* [CustomizableHtmlReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.CustomizableHtmlReport.html)
* [SingleFileReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.SingleFileReport.html)
* [DirectoryReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.DirectoryReport.html)
* [FindBugsXmlReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.FindBugsXmlReport.html)
* [Report](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.Report.html)
* [Reporting](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.Reporting.html)
* [ReportContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.ReportContainer.html)
* [ReportingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.ReportingExtension.html)
* **Eclipse/IDEA model types**
* [EclipseModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseModel.html)
* [EclipseProject](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseProject.html)
* [EclipseClasspath](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseClasspath.html)
* [EclipseJdt](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseJdt.html)
* [EclipseWtp](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseWtp.html)
* [EclipseWtpComponent](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseWtpComponent.html)
* [EclipseWtpFacet](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseWtpFacet.html)
* [IdeaModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaModel.html)
* [IdeaProject](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaProject.html)
* [IdeaModule](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)
* [IdeaWorkspace](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaWorkspace.html)
* [XmlFileContentMerger](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.api.XmlFileContentMerger.html)
* [FileContentMerger](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.api.FileContentMerger.html)
* **Eclipse/IDEA task types**
* [GenerateEclipseProject](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseProject.html)
* [GenerateEclipseClasspath](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseClasspath.html)
* [GenerateEclipseJdt](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseJdt.html)
* [GenerateEclipseWtpComponent](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseWtpComponent.html)
* [GenerateEclipseWtpFacet](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseWtpFacet.html)
* [GenerateIdeaModule](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.GenerateIdeaModule.html)
* [GenerateIdeaProject](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.GenerateIdeaProject.html)
* [GenerateIdeaWorkspace](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.GenerateIdeaWorkspace.html)
* **Native software types**
* [PrebuiltLibrary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.PrebuiltLibrary.html)
* [PrebuiltSharedLibraryBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.PrebuiltSharedLibraryBinary.html)
* [PrebuiltStaticLibraryBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.PrebuiltStaticLibraryBinary.html)
* [NativeComponentSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeComponentSpec.html)
* [NativeExecutableSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeExecutableSpec.html)
* [NativeLibrarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeLibrarySpec.html)
* [NativeTestSuiteSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.NativeTestSuiteSpec.html)
* [CUnitTestSuiteSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.cunit.CUnitTestSuiteSpec.html)
* [GoogleTestTestSuiteSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.googletest.GoogleTestTestSuiteSpec.html)
* [NativeBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeBinarySpec.html)
* [NativeExecutableBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeExecutableBinarySpec.html)
* [NativeLibraryBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeLibraryBinarySpec.html)
* [SharedLibraryBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.SharedLibraryBinarySpec.html)
* [StaticLibraryBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.StaticLibraryBinarySpec.html)
* [NativeTestSuiteBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.NativeTestSuiteBinarySpec.html)
* [CUnitTestSuiteBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.cunit.CUnitTestSuiteBinarySpec.html)
* [GoogleTestTestSuiteBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.googletest.GoogleTestTestSuiteBinarySpec.html)
* [NativePlatform](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.platform.NativePlatform.html)
* [BuildType](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.BuildType.html)
* [Flavor](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.Flavor.html)
* [Gcc](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.toolchain.Gcc.html)
* [Clang](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.toolchain.Clang.html)
* [VisualCpp](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.toolchain.VisualCpp.html)
* [AssemblerSourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.assembler.AssemblerSourceSet.html)
* [CSourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.c.CSourceSet.html)
* [CppSourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.cpp.CppSourceSet.html)
* [ObjectiveCSourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.objectivec.ObjectiveCSourceSet.html)
* [ObjectiveCppSourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.objectivecpp.ObjectiveCppSourceSet.html)
* [WindowsResourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.rc.WindowsResourceSet.html)
* [VisualStudioProject](https://docs.gradle.org/4.1/dsl/org.gradle.ide.visualstudio.VisualStudioProject.html)
* [VisualStudioSolution](https://docs.gradle.org/4.1/dsl/org.gradle.ide.visualstudio.VisualStudioSolution.html)
* [NativeExecutable](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeExecutable.html)
* [NativeLibrary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeLibrary.html)
* [NativeBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeBinary.html)
* [NativeExecutableBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeExecutableBinary.html)
* [SharedLibraryBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.SharedLibraryBinary.html)
* [StaticLibraryBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.StaticLibraryBinary.html)
* **Native component task types**
* [CppCompile](https://docs.gradle.org/4.1/dsl/org.gradle.language.cpp.tasks.CppCompile.html)
* [CCompile](https://docs.gradle.org/4.1/dsl/org.gradle.language.c.tasks.CCompile.html)
* [Assemble](https://docs.gradle.org/4.1/dsl/org.gradle.language.assembler.tasks.Assemble.html)
* [ObjectiveCCompile](https://docs.gradle.org/4.1/dsl/org.gradle.language.objectivec.tasks.ObjectiveCCompile.html)
* [ObjectiveCppCompile](https://docs.gradle.org/4.1/dsl/org.gradle.language.objectivecpp.tasks.ObjectiveCppCompile.html)
* [WindowsResourceCompile](https://docs.gradle.org/4.1/dsl/org.gradle.language.rc.tasks.WindowsResourceCompile.html)
* [LinkExecutable](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.tasks.LinkExecutable.html)
* [LinkSharedLibrary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.tasks.LinkSharedLibrary.html)
* [CreateStaticLibrary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.tasks.CreateStaticLibrary.html)
* [InstallExecutable](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.tasks.InstallExecutable.html)
* [RunTestExecutable](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.tasks.RunTestExecutable.html)

# Gradle Build Language Reference

## Introduction

This reference guide describes the various types which make up the Gradle build language, or DSL(领域特定语言).

## Some basics

There are a few basic concepts that you should understand, which will help you write Gradle scripts.

First, Gradle scripts are configuration scripts（配置脚本）. As the script executes, it configures an object of a particular type. For example, as a build script executes, it configures an object of type [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html)（它配置特定的工程类型）. This object is called the delegate object （委托对象）of the script. The following table shows the delegate for each type of Gradle script.

| **Type of script** | **Delegates to instance of** |
| --- | --- |
| Build script | [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html) #build脚本 |
| Init script | [Gradle](https://docs.gradle.org/4.1/dsl/org.gradle.api.invocation.Gradle.html) #初始化脚本 |
| Settings script | [Settings](https://docs.gradle.org/4.1/dsl/org.gradle.api.initialization.Settings.html) #settings脚本 |

The properties and methods(属性和方法) of the delegate object（委托对象） are available for you to use in the script.（me：这里应该是说，gradle内置了配置project gradle settings的脚本接口，我们可以在相应的脚本中使用这些接口定义的属性和方法，如在build.gradle中使用apply plugin/repositories/dependenties）

Second, each Gradle script implements the [Script](https://docs.gradle.org/4.1/dsl/org.gradle.api.Script.html) interface(由此也可看出gradle有脚本接口，每个定义的脚本都自动的实现这些接口). This interface defines a number of properties and methods（这些接口定义了属性和方法） which you can use in the script.

## Build script structure

A build script is made up of zero or more statements and script blocks（陈述语句和块脚本）. Statements can include method calls, property assignments, and local variable definitions（陈述语句中可以包含方法调用、属性赋值和本地变量定义）. A script block is a method call which takes a closure as a parameter（块脚本是一个方法调用，这个语句块将一个闭包作为参数）. The closure is treated as a configuration closure（配置闭包） which configures some delegate object as it executes（当方法执行的时候会配置一些委托对象）. The top level script blocks（顶级配置块脚本） are listed below.

| **Block** | **Description** |
| --- | --- |
| [allprojects { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:allprojects(groovy.lang.Closure)) | Configures this project and each of its sub-projects. |
| [artifacts { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:artifacts(groovy.lang.Closure)) | Configures the published artifacts for this project. |
| [buildscript { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:buildscript(groovy.lang.Closure)) | Configures the build script classpath for this project. |
| [configurations { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:configurations(groovy.lang.Closure)) | Configures the dependency configurations for this project. |
| [dependencies { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:dependencies(groovy.lang.Closure)) | Configures the dependencies for this project. |
| [repositories { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:repositories(groovy.lang.Closure)) | Configures the repositories for this project. |
| [sourceSets { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:sourceSets(groovy.lang.Closure)) | Configures the source sets of this project. |
| [subprojects { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:subprojects(groovy.lang.Closure)) | Configures the sub-projects of this project. |
| [publishing { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:publishing(groovy.lang.Closure)) | Configures the [PublishingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.PublishingExtension.html) added by the publishing plugin. |

说明：

|  |  |
| --- | --- |
| Block | 说明 |
| allprojects | 配置本工程和它的所有子项目 |
| artifacts | 配置本工程发布的包 |
| buildscript | build脚本的classpath配置 |
| configurations | 配置依赖的相关配置 |
| dependencies | 工程依赖配置 |
| repositories | 工程仓库配置 |
| sourceSets | 工程的资源集合配置 |
| subprojects | 工程的子工程配置 |
| publishing | 被发布插件添加的[PublishingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.PublishingExtension.html)的配置 |

A build script is also a Groovy script（gradle的build脚本的编写语言是Groovy）, and so can contain those elements allowed in a Groovy script, such as method definitions and class definitions(所以可以在build脚本中项Groovy脚本那样定义类和方法).

## Core types

Listed below are some of the central types which are used in Gradle scripts:

| **Type Type** | | **Description** |
| --- | --- | --- |
| [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html) | This interface is the main API you use to interact with Gradle from your build file. From a Project, you have programmatic access to all of Gradle's features. | |
| [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html) | A Task represents a single atomic piece of work for a build, such as compiling classes or generating javadoc. | |
| [Gradle](https://docs.gradle.org/4.1/dsl/org.gradle.api.invocation.Gradle.html) | Represents an invocation of Gradle. | |
| [Settings](https://docs.gradle.org/4.1/dsl/org.gradle.api.initialization.Settings.html) | Declares the configuration required to instantiate and configure the hierarchy of [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html)instances which are to participate in a build. | |
| [IncludedBuild](https://docs.gradle.org/4.1/dsl/org.gradle.api.initialization.IncludedBuild.html) | A build that is included in the composite. | |
| [Script](https://docs.gradle.org/4.1/dsl/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. | |
| [JavaToolChain](https://docs.gradle.org/4.1/dsl/org.gradle.jvm.toolchain.JavaToolChain.html) | A set of tools for building from Java source. | |
| [SourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.SourceSet.html) | A SourceSet represents a logical group of Java source and resources. | |
| [SourceSetOutput](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.SourceSetOutput.html) | A collection of all output directories (compiled classes, processed resources, etc.) - notice that [SourceSetOutput](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.SourceSetOutput.html) extends [FileCollection](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/FileCollection.html). | |
| [SourceDirectorySet](https://docs.gradle.org/4.1/dsl/org.gradle.api.file.SourceDirectorySet.html) | A SourceDirectorySet represents a set of source files composed from a set of source directories, along with associated include and exclude patterns. | |
| [IncrementalTaskInputs](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.incremental.IncrementalTaskInputs.html) | Provides access to any input files that need to be processed by an incremental task. | |
| [Configuration](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.Configuration.html) | A Configuration represents a group of artifacts and their dependencies. Find more information about declaring dependencies to a configuration or about managing configurations in docs for [ConfigurationContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ConfigurationContainer.html) | |
| [ResolutionStrategy](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ResolutionStrategy.html) | Defines the strategies around dependency resolution. For example, forcing certain dependency versions, substitutions, conflict resolutions or snapshot timeouts. | |
| [ArtifactResolutionQuery](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.query.ArtifactResolutionQuery.html) | A builder to construct a query that can resolve selected software artifacts of the specified components. | |
| [ComponentSelection](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ComponentSelection.html) | Represents a tuple of the component selector of a module and a candidate version to be evaluated in a component selection rule. | |
| [ComponentSelectionRules](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ComponentSelectionRules.html) | Represents a container for component selection rules. Rules can be applied as part of the resolutionStrategy of a configuration and individual components can be explicitly accepted or rejected by rule. Components that are neither accepted or rejected will be subject to the default version matching strategies. | |
| [GradlePluginPortal](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.GradlePluginPortal.html) | The Gradle Plugin Portal which can be used to resolve plugins. | |
| [ExtensionAware](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.ExtensionAware.html) | Objects that can be extended at runtime with other objects. | |
| [ExtraPropertiesExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.ExtraPropertiesExtension.html) | Additional, ad-hoc, properties for Gradle domain objects. | |
| [PublishingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.PublishingExtension.html) | The configuration of how to “publish” the different components of a project. | |
| [IvyPublication](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyPublication.html) | A IvyPublication is the representation/configuration of how Gradle should publish something in Ivy format, to an Ivy repository. You directly add a named Ivy Publication the project's publishing.publications container by providing [IvyPublication](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyPublication.html) as the type. | |
| [IvyArtifact](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyArtifact.html) | An artifact published as part of a [IvyPublication](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyPublication.html). | |
| [IvyArtifactSet](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyArtifactSet.html) | A Collection of [IvyArtifact](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyArtifact.html)s to be included in an [IvyPublication](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyPublication.html). Being a [DomainObjectSet](https://docs.gradle.org/4.1/javadoc/org/gradle/api/DomainObjectSet.html), a IvyArtifactSet provides convenient methods for querying, filtering, and applying actions to the set of [IvyArtifact](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyArtifact.html)s. | |
| [IvyModuleDescriptorSpec](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyModuleDescriptorSpec.html) | The descriptor of any Ivy publication. | |
| [IvyPluginRepository](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.IvyPluginRepository.html) | Represents an Ivy repository which contains Gradle plugins. | |
| [MavenPublication](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenPublication.html) | A MavenPublication is the representation/configuration of how Gradle should publish something in Maven format. You directly add a named Maven Publication the project's publishing.publications container by providing [MavenPublication](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenPublication.html) as the type. | |
| [MavenArtifact](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenArtifact.html) | An artifact published as part of a [MavenPublication](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenPublication.html). | |
| [MavenArtifactSet](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenArtifactSet.html) | A Collection of [MavenArtifact](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenArtifact.html)s to be included in a [MavenPublication](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenPublication.html). Being a [DomainObjectSet](https://docs.gradle.org/4.1/javadoc/org/gradle/api/DomainObjectSet.html), a MavenArtifactSet provides convenient methods for querying, filtering, and applying actions to the set of [MavenArtifact](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenArtifact.html)s. | |
| [MavenPom](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenPom.html) | The POM for a Maven publication. The [MavenPom.withXml(org.gradle.api.Action)](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenPom.html#org.gradle.api.publish.maven.MavenPom:withXml(org.gradle.api.Action)) method can be used to modify the descriptor after it has been generated according to the publication data. | |
| [MavenPluginRepository](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.MavenPluginRepository.html) | Represents a Maven repository which contains Gradle plugins. | |
| [PluginDependenciesSpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.use.PluginDependenciesSpec.html) | The DSL for declaring plugins to use in a script. | |
| [PluginDependencySpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.use.PluginDependencySpec.html) | A mutable specification of a dependency on a plugin. | |
| [PluginManagementSpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.management.PluginManagementSpec.html) | Configures how plugins are resolved. | |
| [PluginRepositoriesSpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.PluginRepositoriesSpec.html) | Used to declare [PluginRepository](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.PluginRepository.html) instances. | |
| [PluginRepository](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.PluginRepository.html) | Represents a repository from which Gradle plugins can be resolved. | |
| [ResourceHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.resources.ResourceHandler.html) | Provides access to resource-specific utility methods, for example factory methods that create various resources. | |
| [TextResourceFactory](https://docs.gradle.org/4.1/dsl/org.gradle.api.resources.TextResourceFactory.html) | Creates TextResources backed by sources such as strings, files, and archive entries. | |

|  |  |
| --- | --- |
| Type | 描述 |
| [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html) | 这个接口是你在脚本中用来同gradle交互主要的api。从Project接口，你已经程序上能够操作所有的gradle属性 |
| [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html) | 一个Task代表了build构建项目时的一个简单原子性的任务。例如编译classes文件或者产生javadoc文档 |
| [Gradle](https://docs.gradle.org/4.1/dsl/org.gradle.api.invocation.Gradle.html) | 代表了一Gradle的Invocation |
| [Settings](https://docs.gradle.org/4.1/dsl/org.gradle.api.initialization.Settings.html) | 定义例示和配置Project实例的层次等级的配置项，这个Project将要参与一次构建build。 |
| [IncludedBuild](https://docs.gradle.org/4.1/dsl/org.gradle.api.initialization.IncludedBuild.html) | 一个包含在本次作业中的build构建 |
| [Script](https://docs.gradle.org/4.1/dsl/org.gradle.api.Script.html) | 本接口被所有的gradle脚本实现，以便于增加一些gradle特定（即在接口中定义）的方法。因为你的编译脚本将会实现这个接口，你能够在你的脚本中直接使用此接口定义的方法。 |
| [JavaToolChain](https://docs.gradle.org/4.1/dsl/org.gradle.jvm.toolchain.JavaToolChain.html) | 从java源码编译的一套工具 |
| [SourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.SourceSet.html) | 源码集代表了java源码和资源的逻辑组成 |
| [SourceSetOutput](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.SourceSetOutput.html) | 所有输出目录（编译classes，进程所需资源等等）的集合。注意，SourceSetOutput继承自FileCollection |
| [SourceDirectorySet](https://docs.gradle.org/4.1/dsl/org.gradle.api.file.SourceDirectorySet.html) | [SourceDirectorySet](https://docs.gradle.org/4.1/dsl/org.gradle.api.file.SourceDirectorySet.html)代表了一个集合，这个集合包含源码文件，这些源码文件有一些源码目录以及一些相关的东西组成 |
| [IncrementalTaskInputs](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.incremental.IncrementalTaskInputs.html) | 提供了访问任何输入文件的接口，这些输入文件需要被一个IncrementalTask处理。 |
| [Configuration](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.Configuration.html) | Configruation代表了一组artifacts和他们的依赖。关于针对一个configuration定义一个依赖或者在docs中管理configurations的详细信息请参照[ConfigurationContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ConfigurationContainer.html) |
| [ResolutionStrategy](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ResolutionStrategy.html) | 定义关于依赖方案的策略。例如，强制确定依赖版本，substitutions，冲突方案或者snapshot的timeouts等 |
| [ArtifactResolutionQuery](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.query.ArtifactResolutionQuery.html) | 构建一个能够解决选择软件指定的部分的artifactsquery的builder |
| [ComponentSelection](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ComponentSelection.html) | 代表了一组component选择器。这些选择器包含了模块和一个在一个组件的选举规则中国将要被evaluted的候选版本 |
| [ComponentSelectionRules](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ComponentSelectionRules.html) |  |
| [GradlePluginPortal](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.GradlePluginPortal.html) |  |
| [ExtensionAware](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.ExtensionAware.html) |  |
| [ExtraPropertiesExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.ExtraPropertiesExtension.html) |  |
| [PublishingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.PublishingExtension.html) |  |
| [IvyPublication](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyPublication.html) |  |
| [IvyArtifact](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyArtifact.html) |  |
| [IvyArtifactSet](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyArtifactSet.html) |  |
| [IvyModuleDescriptorSpec](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyModuleDescriptorSpec.html) |  |
| [IvyPluginRepository](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.IvyPluginRepository.html) |  |
| [MavenPublication](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenPublication.html) |  |
| [MavenArtifact](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenArtifact.html) |  |
| [MavenArtifactSet](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenArtifactSet.html) |  |
| [MavenPom](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenPom.html) |  |
| [MavenPluginRepository](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.MavenPluginRepository.html) |  |
| [PluginDependenciesSpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.use.PluginDependenciesSpec.html) |  |
| [PluginDependencySpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.use.PluginDependencySpec.html) |  |
| [PluginManagementSpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.management.PluginManagementSpec.html) |  |
| [PluginRepositoriesSpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.PluginRepositoriesSpec.html) |  |
| [PluginRepository](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.PluginRepository.html) |  |
| [ResourceHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.resources.ResourceHandler.html) |  |
| [TextResourceFactory](https://docs.gradle.org/4.1/dsl/org.gradle.api.resources.TextResourceFactory.html) |  |

//================20170822 上表格未完待续

## Container types

Container types（容器类型）that handle various declarative（发布，公布） elements (e.g. dependencies, configurations, artifacts, etc.):

| **Type** | **Description** |
| --- | --- |
| [ConfigurationContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ConfigurationContainer.html) | A ConfigurationContainer is responsible for declaring and managing configurations. See also [Configuration](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.Configuration.html). |
| [RepositoryHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.RepositoryHandler.html) | A RepositoryHandler manages a set of repositories, allowing repositories to be defined and queried. |
| [DependencyHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html) | A DependencyHandler is used to declare dependencies. Dependencies are grouped into configurations (see [Configuration](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.Configuration.html)). |
| [ArtifactHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.ArtifactHandler.html) | This class is for defining artifacts to be published and adding them to configurations. Creating publish artifacts does not mean to create an archive. What is created is a domain object which represents a file to be published and information on how it should be published (e.g. the name). |

|  |  |
| --- | --- |
| Type | 描述 |
| [ConfigurationContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ConfigurationContainer.html) | [ConfigurationContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ConfigurationContainer.html)负责定义和管理configurations，参考[Configuration](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.Configuration.html).的说明 |
| [RepositoryHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.RepositoryHandler.html) | [RepositoryHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.RepositoryHandler.html)管理repositories的集合，允许仓库能够被定义和查询 |
| [DependencyHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html) | [DependencyHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html)常被用来定义依赖。这些依赖被分组成配置（参考[Configuration](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.Configuration.html)的说明） |
| [ArtifactHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.ArtifactHandler.html) | 这个类是用来定义将要被发布的artifacts，并将他们加入到配置。创建artifacts并不意味着创建了一个archive。它创建的是一个领域模型，这个领域模型代表了一个将要被发布的文件以及关于这个文件被发布的相关信息(如发布名) |

## Build Cache types

Types used to connect to and configure the build cache（这个类型常被用来连接和配置build cache）:

| **Type** | **Description** |
| --- | --- |
| [BuildCacheConfiguration](https://docs.gradle.org/4.1/dsl/org.gradle.caching.configuration.BuildCacheConfiguration.html) | Configuration for the [build cache](https://docs.gradle.org/4.1/userguide/build_cache.html) for an entire Gradle build. |
| [DirectoryBuildCache](https://docs.gradle.org/4.1/dsl/org.gradle.caching.local.DirectoryBuildCache.html) | Configuration object for the local directory build cache. |
| [HttpBuildCache](https://docs.gradle.org/4.1/dsl/org.gradle.caching.http.HttpBuildCache.html) | Configuration object for the HTTP build cache. The build cache only supports BASIC authentication currently. |

## Input Normalization types

Types used to configure input normalization

| **Type** | **Description** |
| --- | --- |
| [InputNormalizationHandler](https://docs.gradle.org/4.1/dsl/org.gradle.normalization.InputNormalizationHandler.html) | Used to configure input normalization. Currently, it is only possible to configure runtime classpath normalization. |
| [InputNormalization](https://docs.gradle.org/4.1/dsl/org.gradle.normalization.InputNormalization.html) | Input normalization configuration. Input normalization is used when Gradle tries to determine if two task inputs are different. Gradle normalizes both inputs and the inputs are considered different if and only if the normalizations are different. |
| [RuntimeClasspathNormalization](https://docs.gradle.org/4.1/dsl/org.gradle.normalization.RuntimeClasspathNormalization.html) | Configuration of runtime classpath normalization. |

## Help Task types

Below are the task types that are available for every Gradle project. Those task types can also be declared and configured directly in the build script.

| **Type** | **Description** |
| --- | --- |
| [TaskReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.TaskReportTask.html) | Displays a list of tasks in the project. An instance of this type is used when you execute the tasks task from the command-line. |
| [ProjectReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.ProjectReportTask.html) | Displays a list of projects in the build. An instance of this type is used when you execute the projects task from the command-line. |
| [DependencyReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.DependencyReportTask.html) | Displays the dependency tree for a project. An instance of this type is used when you execute the dependencies task from the command-line. |
| [DependencyInsightReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.DependencyInsightReportTask.html) | Generates a report that attempts to answer questions like: |
| [PropertyReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.PropertyReportTask.html) | Displays the properties of a project. An instance of this type is used when you execute the properties task from the command-line. |
| [ComponentReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.components.ComponentReport.html) | Displays some details about the software components produced by the project. |
| [DependentComponentsReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.dependents.DependentComponentsReport.html) | Displays dependent components. |
| [ModelReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.model.ModelReport.html) | Displays some details about the configuration model of the project. An instance of this type is used when you execute the model task from the command-line. |

## Task types

Listed below are the various task types which are available for use in your build script:

| **Type** | **Description** |
| --- | --- |
| [AntlrTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.antlr.AntlrTask.html) | Generates parsers from Antlr grammars. |
| [BuildEnvironmentReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.BuildEnvironmentReportTask.html) | Provides information about the build environment for the project that the task is associated with. |
| [Checkstyle](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.Checkstyle.html) | Runs Checkstyle against some source files. |
| [CodeNarc](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.CodeNarc.html) | Runs CodeNarc against some source files. |
| [CompareGradleBuilds](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.buildcomparison.gradle.CompareGradleBuilds.html) | Executes two Gradle builds (that can be the same build) with specified versions and compares the outcomes. Please see the “Comparing Builds” chapter of the Gradle User Guide for more information. |
| [Copy](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.Copy.html) | Copies files into a destination directory. This task can also rename and filter files as it copies. The task implements [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html) for specifying what to copy. |
| [CreateStartScripts](https://docs.gradle.org/4.1/dsl/org.gradle.jvm.application.tasks.CreateStartScripts.html) | Creates start scripts for launching JVM applications. |
| [Delete](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.Delete.html) | Deletes files or directories. Example: |
| [Ear](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ear.Ear.html) | Assembles an EAR archive. |
| [Exec](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.Exec.html) | Executes a command line process. Example: |
| [FindBugs](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.FindBugs.html) | Analyzes code with [FindBugs](http://findbugs.sourceforge.net/). See the [FindBugs Manual](http://findbugs.sourceforge.net/manual/) for additional information on configuration options. |
| [GenerateIvyDescriptor](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.tasks.GenerateIvyDescriptor.html) | Generates an Ivy XML Module Descriptor file. |
| [GenerateMavenPom](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.tasks.GenerateMavenPom.html) | Generates a Maven module descriptor (POM) file. |
| [GenerateBuildDashboard](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.GenerateBuildDashboard.html) | Generates build dashboard report. |
| [GradleBuild](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.GradleBuild.html) | Executes a Gradle build. |
| [GroovyCompile](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.compile.GroovyCompile.html) | Compiles Groovy source files, and optionally, Java source files. |
| [Groovydoc](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.javadoc.Groovydoc.html) | Generates HTML API documentation for Groovy source, and optionally, Java source. |
| [HtmlDependencyReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.dependencies.HtmlDependencyReportTask.html) | Generates an HTML dependency report. This report combines the features of the ASCII dependency report and those of the ASCII dependency insight report. For a given project, it generates a tree of the dependencies of every configuration, and each dependency can be clicked to show the insight of this dependency. |
| [JacocoReport](https://docs.gradle.org/4.1/dsl/org.gradle.testing.jacoco.tasks.JacocoReport.html) | Task to generate HTML, Xml and CSV reports of Jacoco coverage data. |
| [JacocoMerge](https://docs.gradle.org/4.1/dsl/org.gradle.testing.jacoco.tasks.JacocoMerge.html) | Task to merge multiple execution data files into one. |
| [JacocoCoverageVerification](https://docs.gradle.org/4.1/dsl/org.gradle.testing.jacoco.tasks.JacocoCoverageVerification.html) | Task for verifying code coverage metrics. Fails the task if violations are detected based on specified rules. |
| [Jar](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.bundling.Jar.html) | Assembles a JAR archive. |
| [JavaCompile](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.compile.JavaCompile.html) | Compiles Java source files. |
| [Javadoc](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.javadoc.Javadoc.html) | Generates HTML API documentation for Java classes. |
| [JavaExec](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.JavaExec.html) | Executes a Java application in a child process. |
| [JDepend](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.JDepend.html) | Analyzes code with [JDepend](https://github.com/clarkware/jdepend). |
| [Pmd](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.Pmd.html) | Runs a set of static code analysis rules on Java source code files and generates a report of problems found. |
| [PublishToIvyRepository](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.tasks.PublishToIvyRepository.html) | Publishes an IvyPublication to an IvyArtifactRepository. |
| [PublishToMavenRepository](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.tasks.PublishToMavenRepository.html) | Publishes a [MavenPublication](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenPublication.html) to a [MavenArtifactRepository](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.repositories.MavenArtifactRepository.html). |
| [ScalaCompile](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.scala.ScalaCompile.html) | Compiles Scala source files, and optionally, Java source files. |
| [ScalaDoc](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.scala.ScalaDoc.html) | Generates HTML API documentation for Scala source files. |
| [InitBuild](https://docs.gradle.org/4.1/dsl/org.gradle.buildinit.tasks.InitBuild.html) | Generates a Gradle project structure. |
| [Sign](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.signing.Sign.html) | A task for creating digital signature files for one or more; tasks, files, publishable artifacts or configurations. |
| [Sync](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.Sync.html) | Synchronizes the contents of a destination directory with some source directories and files. |
| [Tar](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.bundling.Tar.html) | Assembles a TAR archive. |
| [Test](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.testing.Test.html) | Executes JUnit (3.8.x or 4.x) or TestNG tests. Test are always run in (one or more) separate JVMs. The sample below shows various configuration options. |
| [TestReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.testing.TestReport.html) | Generates an HTML test report from the results of one or more [Test](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.testing.Test.html) tasks. |
| [Upload](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.Upload.html) | Uploads the artifacts of a [Configuration](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.Configuration.html) to a set of repositories. |
| [War](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.bundling.War.html) | Assembles a WAR archive. |
| [Wrapper](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.wrapper.Wrapper.html) | Generates scripts (for \*nix and windows) which allow you to build your project with Gradle, without having to install Gradle. |
| [WriteProperties](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.WriteProperties.html) | Writes a [Properties](http://download.oracle.com/javase/7/docs/api/java/util/Properties.html) in a way that the results can be expected to be reproducible. |
| [Zip](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.bundling.Zip.html) | Assembles a ZIP archive. The default is to compress the contents of the zip. |

## Reporting types

Listed below are some of the types which are used when generating reports:

| **Type** | **Description** |
| --- | --- |
| [CustomizableHtmlReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.CustomizableHtmlReport.html) | A HTML Report whose generation can be customized with a XSLT stylesheet. |
| [SingleFileReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.SingleFileReport.html) | A report that is a single file. |
| [DirectoryReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.DirectoryReport.html) | A directory based report to be created. |
| [FindBugsXmlReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.FindBugsXmlReport.html) | The single file XML report for FindBugs. |
| [Report](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.Report.html) | A file based report to be created. |
| [Reporting](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.Reporting.html) | An object that provides reporting options. |
| [ReportContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.ReportContainer.html) | A container of [Report](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.Report.html) objects, that represent potential reports. |
| [ReportingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.ReportingExtension.html) | A project extension named "reporting" that provides basic reporting settings and utilities. |

## Eclipse/IDEA model types

Used to configure Eclipse or IDEA plugins

| **Type** | **Description** |
| --- | --- |
| [EclipseModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseModel.html) | DSL-friendly model of the Eclipse project information. First point of entry for customizing Eclipse project generation. |
| [EclipseProject](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseProject.html) | Enables fine-tuning project details (.project file) of the Eclipse plugin |
| [EclipseClasspath](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseClasspath.html) | The build path settings for the generated Eclipse project. Used by the [GenerateEclipseClasspath](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseClasspath.html)task to generate an Eclipse .classpath file. |
| [EclipseJdt](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseJdt.html) | Enables fine-tuning jdt details of the Eclipse plugin |
| [EclipseWtp](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseWtp.html) | Enables fine-tuning wtp/wst details of the Eclipse plugin |
| [EclipseWtpComponent](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseWtpComponent.html) | Enables fine-tuning wtp component details of the Eclipse plugin |
| [EclipseWtpFacet](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseWtpFacet.html) | Enables fine-tuning wtp facet details of the Eclipse plugin |
| [IdeaModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaModel.html) | DSL-friendly model of the IDEA project information. First point of entry when it comes to customizing the IDEA generation. |
| [IdeaProject](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaProject.html) | Enables fine-tuning project details (\*.ipr file) of the IDEA plugin. |
| [IdeaModule](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html) | Enables fine-tuning module details (\*.iml file) of the IDEA plugin. |
| [IdeaWorkspace](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaWorkspace.html) | Enables fine-tuning workspace details (\*.iws file) of the IDEA plugin. |
| [XmlFileContentMerger](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.api.XmlFileContentMerger.html) | Models the generation/parsing/merging capabilities. Adds XML-related hooks. |
| [FileContentMerger](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.api.FileContentMerger.html) | Models the generation/parsing/merging capabilities. |

## Eclipse/IDEA task types

Tasks contributed by IDE plugins. To configure IDE plugins please use IDE model types.

| **Type** | **Description** |
| --- | --- |
| [GenerateEclipseProject](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseProject.html) | Generates an Eclipse .project file. If you want to fine tune the eclipse configuration |
| [GenerateEclipseClasspath](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseClasspath.html) | Generates an Eclipse .classpath file. If you want to fine tune the eclipse configuration |
| [GenerateEclipseJdt](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseJdt.html) | Generates the Eclipse JDT configuration file. If you want to fine tune the eclipse configuration |
| [GenerateEclipseWtpComponent](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseWtpComponent.html) | Generates the org.eclipse.wst.common.component settings file for Eclipse WTP. If you want to fine tune the eclipse configuration |
| [GenerateEclipseWtpFacet](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseWtpFacet.html) | Generates the org.eclipse.wst.common.project.facet.core settings file for Eclipse WTP. If you want to fine tune the eclipse configuration |
| [GenerateIdeaModule](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.GenerateIdeaModule.html) | Generates an IDEA module file. If you want to fine tune the idea configuration |
| [GenerateIdeaProject](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.GenerateIdeaProject.html) | Generates an IDEA project file for root project \*only\*. If you want to fine tune the idea configuration |
| [GenerateIdeaWorkspace](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.GenerateIdeaWorkspace.html) | Generates an IDEA workspace file \*only\* for root project. There's little you can configure about workspace generation at the moment. |

## Native software model types

Used to configure software components developed with native code.

| **Type** | **Description** |
| --- | --- |
| [PrebuiltLibrary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.PrebuiltLibrary.html) | A library component that is not built by gradle. |
| [PrebuiltSharedLibraryBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.PrebuiltSharedLibraryBinary.html) | A shared library that exists at a known location on the filesystem. |
| [PrebuiltStaticLibraryBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.PrebuiltStaticLibraryBinary.html) | A static library that exists at a known location on the filesystem. |
| [NativeComponentSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeComponentSpec.html) | Definition of a software component that is to be built by Gradle to run a on JVM platform. |
| [NativeExecutableSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeExecutableSpec.html) | Definition of a native executable component that is to be built by Gradle. |
| [NativeLibrarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeLibrarySpec.html) | Definition of a native library component that is to be built by Gradle. |
| [NativeTestSuiteSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.NativeTestSuiteSpec.html) | A component representing a suite of tests that will be executed together. |
| [CUnitTestSuiteSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.cunit.CUnitTestSuiteSpec.html) | Test suite of CUnit tests. |
| [GoogleTestTestSuiteSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.googletest.GoogleTestTestSuiteSpec.html) | Test suite of Google Test tests. |
| [NativeBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeBinarySpec.html) | Represents a binary artifact that is the result of building a native component. |
| [NativeExecutableBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeExecutableBinarySpec.html) | An binary built by Gradle for a native application. |
| [NativeLibraryBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeLibraryBinarySpec.html) | Represents a binary artifact that is the result of building a native library component. |
| [SharedLibraryBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.SharedLibraryBinarySpec.html) | A shared library binary built by Gradle for a native library. |
| [StaticLibraryBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.StaticLibraryBinarySpec.html) | A static library binary built by Gradle for a native library. |
| [NativeTestSuiteBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.NativeTestSuiteBinarySpec.html) | An executable which runs a suite of tests. |
| [CUnitTestSuiteBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.cunit.CUnitTestSuiteBinarySpec.html) | An executable which run a CUnit test suite. |
| [GoogleTestTestSuiteBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.googletest.GoogleTestTestSuiteBinarySpec.html) | An executable which run a Google Test test suite. |
| [NativePlatform](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.platform.NativePlatform.html) | A target platform for building native binaries. Each target platform is given a name, and may optionally be given a specific [Architecture](https://docs.gradle.org/4.1/javadoc/org/gradle/nativeplatform/platform/Architecture.html) and/or [OperatingSystem](https://docs.gradle.org/4.1/javadoc/org/gradle/nativeplatform/platform/OperatingSystem.html) to target. |
| [BuildType](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.BuildType.html) | Specifies a build-type for a native binary. Common build types are 'debug' and 'release', but others may be defined. |
| [Flavor](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.Flavor.html) | Defines a custom variant that differentiate a [NativeBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeBinary.html). |
| [Gcc](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.toolchain.Gcc.html) | The [GNU GCC](http://gcc.gnu.org/) tool chain. |
| [Clang](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.toolchain.Clang.html) | The [Clang](http://clang.llvm.org/) tool chain. |
| [VisualCpp](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.toolchain.VisualCpp.html) | The Visual C++ tool chain. |
| [AssemblerSourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.assembler.AssemblerSourceSet.html) | A set of assembly language sources. |
| [CSourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.c.CSourceSet.html) | A set of C source files. |
| [CppSourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.cpp.CppSourceSet.html) | A set of C++ source files. |
| [ObjectiveCSourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.objectivec.ObjectiveCSourceSet.html) | A set of Objective-C source files. |
| [ObjectiveCppSourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.objectivecpp.ObjectiveCppSourceSet.html) | A set of Objective-C++ source files. |
| [WindowsResourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.rc.WindowsResourceSet.html) | A set of Windows Resource definition files. |
| [VisualStudioProject](https://docs.gradle.org/4.1/dsl/org.gradle.ide.visualstudio.VisualStudioProject.html) | A visual studio project, created from one or more [NativeBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeBinary.html) instances. |
| [VisualStudioSolution](https://docs.gradle.org/4.1/dsl/org.gradle.ide.visualstudio.VisualStudioSolution.html) | A visual studio solution, representing one or more [NativeBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeBinarySpec.html) instances from the same [NativeComponentSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeComponentSpec.html). |
| [NativeExecutable](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeExecutable.html) | An executable native component that is built by Gradle. |
| [NativeLibrary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeLibrary.html) | A library component that is built by a gradle project. |
| [NativeBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeBinary.html) | Represents a particular binary artifact. |
| [NativeExecutableBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeExecutableBinary.html) | A binary artifact for a [NativeExecutable](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeExecutable.html), targeted at a particular platform with specific configuration. |
| [SharedLibraryBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.SharedLibraryBinary.html) | A [NativeLibrary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeLibrary.html) that has been compiled and linked as a shared library. |
| [StaticLibraryBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.StaticLibraryBinary.html) | A [NativeLibrary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeLibrary.html) that has been compiled and archived into a static library. |

## Native binary task types

Tasks used to build native binaries.

| **Type** | **Description** |
| --- | --- |
| [CppCompile](https://docs.gradle.org/4.1/dsl/org.gradle.language.cpp.tasks.CppCompile.html) | Compiles C++ source files into object files. |
| [CCompile](https://docs.gradle.org/4.1/dsl/org.gradle.language.c.tasks.CCompile.html) | Compiles C source files into object files. |
| [Assemble](https://docs.gradle.org/4.1/dsl/org.gradle.language.assembler.tasks.Assemble.html) | Translates Assembly language source files into object files. |
| [ObjectiveCCompile](https://docs.gradle.org/4.1/dsl/org.gradle.language.objectivec.tasks.ObjectiveCCompile.html) | Compiles Objective-C source files into object files. |
| [ObjectiveCppCompile](https://docs.gradle.org/4.1/dsl/org.gradle.language.objectivecpp.tasks.ObjectiveCppCompile.html) | Compiles Objective-C++ source files into object files. |
| [WindowsResourceCompile](https://docs.gradle.org/4.1/dsl/org.gradle.language.rc.tasks.WindowsResourceCompile.html) | Compiles Windows Resource scripts into .res files. |
| [LinkExecutable](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.tasks.LinkExecutable.html) | Links a binary executable from object files and libraries. |
| [LinkSharedLibrary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.tasks.LinkSharedLibrary.html) | Links a binary shared library from object files and imported libraries. |
| [CreateStaticLibrary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.tasks.CreateStaticLibrary.html) | Assembles a static library from object files. |
| [InstallExecutable](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.tasks.InstallExecutable.html) | Installs an executable with it's dependent libraries so it can be easily executed. |
| [RunTestExecutable](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.tasks.RunTestExecutable.html) | Runs a compiled and installed test executable. |

* [Home](https://docs.gradle.org/4.1/dsl/index.html)
* **Project**
  + [Properties](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14842)
    - [announce plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N149C1)
    - [application plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N149E5)
    - [checkstyle plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14A1F)
    - [codenarc plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14A43)
    - [distribution plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14A67)
    - [ear plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14A8B)
    - [eclipse plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14ABC)
    - [findbugs plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14AE0)
    - [idea plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14B04)
    - [jacoco plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14B28)
    - [java plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14B4C)
    - [jdepend plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14BF3)
    - [maven plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14C17)
    - [pmd plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14C3F)
    - [project-report plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14C63)
    - [publishing plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14C8B)
    - [signing plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14CAF)
    - [visual-studio plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14CD3)
    - [war plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14CF7)
  + [Methods](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N14D1F)
    - [ear plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N15054)
    - [java plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N15088)
    - [maven plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N150CB)
    - [osgi plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N150FF)
  + [Script blocks](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N15142)
    - [announce plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N151A0)
    - [checkstyle plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N151C4)
    - [codenarc plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N151E8)
    - [distribution plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N1520C)
    - [ear plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N15230)
    - [eclipse plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N1524F)
    - [findbugs plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N15273)
    - [idea plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N15297)
    - [jacoco plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N152BB)
    - [java plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N152DF)
    - [jdepend plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N1530C)
    - [pmd plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N15330)
    - [publishing plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N15354)
    - [signing plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N15378)
    - [visual-studio plugin](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#N1539C)
* **Build script blocks**
* [allprojects { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:allprojects(groovy.lang.Closure))
* [artifacts { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:artifacts(groovy.lang.Closure))
* [buildscript { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:buildscript(groovy.lang.Closure))
* [configurations { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:configurations(groovy.lang.Closure))
* [dependencies { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:dependencies(groovy.lang.Closure))
* [repositories { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:repositories(groovy.lang.Closure))
* [sourceSets { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:sourceSets(groovy.lang.Closure))
* [subprojects { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:subprojects(groovy.lang.Closure))
* [publishing { }](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:publishing(groovy.lang.Closure))
* **Core types**
* [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html)
* [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html)
* [Gradle](https://docs.gradle.org/4.1/dsl/org.gradle.api.invocation.Gradle.html)
* [Settings](https://docs.gradle.org/4.1/dsl/org.gradle.api.initialization.Settings.html)
* [IncludedBuild](https://docs.gradle.org/4.1/dsl/org.gradle.api.initialization.IncludedBuild.html)
* [Script](https://docs.gradle.org/4.1/dsl/org.gradle.api.Script.html)
* [JavaToolChain](https://docs.gradle.org/4.1/dsl/org.gradle.jvm.toolchain.JavaToolChain.html)
* [SourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.SourceSet.html)
* [SourceSetOutput](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.SourceSetOutput.html)
* [SourceDirectorySet](https://docs.gradle.org/4.1/dsl/org.gradle.api.file.SourceDirectorySet.html)
* [IncrementalTaskInputs](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.incremental.IncrementalTaskInputs.html)
* [Configuration](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.Configuration.html)
* [ResolutionStrategy](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ResolutionStrategy.html)
* [ArtifactResolutionQuery](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.query.ArtifactResolutionQuery.html)
* [ComponentSelection](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ComponentSelection.html)
* [ComponentSelectionRules](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ComponentSelectionRules.html)
* [GradlePluginPortal](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.GradlePluginPortal.html)
* [ExtensionAware](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.ExtensionAware.html)
* [ExtraPropertiesExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.ExtraPropertiesExtension.html)
* [PublishingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.PublishingExtension.html)
* [IvyPublication](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyPublication.html)
* [IvyArtifact](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyArtifact.html)
* [IvyArtifactSet](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyArtifactSet.html)
* [IvyModuleDescriptorSpec](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.IvyModuleDescriptorSpec.html)
* [IvyPluginRepository](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.IvyPluginRepository.html)
* [MavenPublication](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenPublication.html)
* [MavenArtifact](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenArtifact.html)
* [MavenArtifactSet](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenArtifactSet.html)
* [MavenPom](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.MavenPom.html)
* [MavenPluginRepository](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.MavenPluginRepository.html)
* [PluginDependenciesSpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.use.PluginDependenciesSpec.html)
* [PluginDependencySpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.use.PluginDependencySpec.html)
* [PluginManagementSpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.management.PluginManagementSpec.html)
* [PluginRepositoriesSpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.PluginRepositoriesSpec.html)
* [PluginRepository](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.repository.PluginRepository.html)
* [ResourceHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.resources.ResourceHandler.html)
* [TextResourceFactory](https://docs.gradle.org/4.1/dsl/org.gradle.api.resources.TextResourceFactory.html)
* **Container types**
* [ConfigurationContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ConfigurationContainer.html)
* [RepositoryHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.RepositoryHandler.html)
* [DependencyHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html)
* [ArtifactHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.ArtifactHandler.html)
* **Build Cache types**
* [BuildCacheConfiguration](https://docs.gradle.org/4.1/dsl/org.gradle.caching.configuration.BuildCacheConfiguration.html)
* [DirectoryBuildCache](https://docs.gradle.org/4.1/dsl/org.gradle.caching.local.DirectoryBuildCache.html)
* [HttpBuildCache](https://docs.gradle.org/4.1/dsl/org.gradle.caching.http.HttpBuildCache.html)
* **Input Normalization types**
* [InputNormalizationHandler](https://docs.gradle.org/4.1/dsl/org.gradle.normalization.InputNormalizationHandler.html)
* [InputNormalization](https://docs.gradle.org/4.1/dsl/org.gradle.normalization.InputNormalization.html)
* [RuntimeClasspathNormalization](https://docs.gradle.org/4.1/dsl/org.gradle.normalization.RuntimeClasspathNormalization.html)
* **Help Task types**
* [TaskReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.TaskReportTask.html)
* [ProjectReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.ProjectReportTask.html)
* [DependencyReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.DependencyReportTask.html)
* [DependencyInsightReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.DependencyInsightReportTask.html)
* [PropertyReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.PropertyReportTask.html)
* [ComponentReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.components.ComponentReport.html)
* [DependentComponentsReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.dependents.DependentComponentsReport.html)
* [ModelReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.model.ModelReport.html)
* **Task types**
* [AntlrTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.antlr.AntlrTask.html)
* [BuildEnvironmentReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.diagnostics.BuildEnvironmentReportTask.html)
* [Checkstyle](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.Checkstyle.html)
* [CodeNarc](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.CodeNarc.html)
* [CompareGradleBuilds](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.buildcomparison.gradle.CompareGradleBuilds.html)
* [Copy](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.Copy.html)
* [CreateStartScripts](https://docs.gradle.org/4.1/dsl/org.gradle.jvm.application.tasks.CreateStartScripts.html)
* [Delete](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.Delete.html)
* [Ear](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ear.Ear.html)
* [Exec](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.Exec.html)
* [FindBugs](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.FindBugs.html)
* [GenerateIvyDescriptor](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.tasks.GenerateIvyDescriptor.html)
* [GenerateMavenPom](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.tasks.GenerateMavenPom.html)
* [GenerateBuildDashboard](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.GenerateBuildDashboard.html)
* [GradleBuild](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.GradleBuild.html)
* [GroovyCompile](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.compile.GroovyCompile.html)
* [Groovydoc](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.javadoc.Groovydoc.html)
* [HtmlDependencyReportTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.dependencies.HtmlDependencyReportTask.html)
* [JacocoReport](https://docs.gradle.org/4.1/dsl/org.gradle.testing.jacoco.tasks.JacocoReport.html)
* [JacocoMerge](https://docs.gradle.org/4.1/dsl/org.gradle.testing.jacoco.tasks.JacocoMerge.html)
* [JacocoCoverageVerification](https://docs.gradle.org/4.1/dsl/org.gradle.testing.jacoco.tasks.JacocoCoverageVerification.html)
* [Jar](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.bundling.Jar.html)
* [JavaCompile](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.compile.JavaCompile.html)
* [Javadoc](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.javadoc.Javadoc.html)
* [JavaExec](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.JavaExec.html)
* [JDepend](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.JDepend.html)
* [Pmd](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.Pmd.html)
* [PublishToIvyRepository](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.ivy.tasks.PublishToIvyRepository.html)
* [PublishToMavenRepository](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.maven.tasks.PublishToMavenRepository.html)
* [ScalaCompile](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.scala.ScalaCompile.html)
* [ScalaDoc](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.scala.ScalaDoc.html)
* [InitBuild](https://docs.gradle.org/4.1/dsl/org.gradle.buildinit.tasks.InitBuild.html)
* [Sign](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.signing.Sign.html)
* [Sync](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.Sync.html)
* [Tar](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.bundling.Tar.html)
* [Test](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.testing.Test.html)
* [TestReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.testing.TestReport.html)
* [Upload](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.Upload.html)
* [War](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.bundling.War.html)
* [Wrapper](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.wrapper.Wrapper.html)
* [WriteProperties](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.WriteProperties.html)
* [Zip](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.bundling.Zip.html)
* **Reporting types**
* [CustomizableHtmlReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.CustomizableHtmlReport.html)
* [SingleFileReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.SingleFileReport.html)
* [DirectoryReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.DirectoryReport.html)
* [FindBugsXmlReport](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.FindBugsXmlReport.html)
* [Report](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.Report.html)
* [Reporting](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.Reporting.html)
* [ReportContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.ReportContainer.html)
* [ReportingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.ReportingExtension.html)
* **Eclipse/IDEA model types**
* [EclipseModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseModel.html)
* [EclipseProject](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseProject.html)
* [EclipseClasspath](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseClasspath.html)
* [EclipseJdt](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseJdt.html)
* [EclipseWtp](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseWtp.html)
* [EclipseWtpComponent](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseWtpComponent.html)
* [EclipseWtpFacet](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseWtpFacet.html)
* [IdeaModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaModel.html)
* [IdeaProject](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaProject.html)
* [IdeaModule](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)
* [IdeaWorkspace](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaWorkspace.html)
* [XmlFileContentMerger](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.api.XmlFileContentMerger.html)
* [FileContentMerger](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.api.FileContentMerger.html)
* **Eclipse/IDEA task types**
* [GenerateEclipseProject](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseProject.html)
* [GenerateEclipseClasspath](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseClasspath.html)
* [GenerateEclipseJdt](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseJdt.html)
* [GenerateEclipseWtpComponent](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseWtpComponent.html)
* [GenerateEclipseWtpFacet](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.GenerateEclipseWtpFacet.html)
* [GenerateIdeaModule](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.GenerateIdeaModule.html)
* [GenerateIdeaProject](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.GenerateIdeaProject.html)
* [GenerateIdeaWorkspace](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.GenerateIdeaWorkspace.html)
* **Native software types**
* [PrebuiltLibrary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.PrebuiltLibrary.html)
* [PrebuiltSharedLibraryBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.PrebuiltSharedLibraryBinary.html)
* [PrebuiltStaticLibraryBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.PrebuiltStaticLibraryBinary.html)
* [NativeComponentSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeComponentSpec.html)
* [NativeExecutableSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeExecutableSpec.html)
* [NativeLibrarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeLibrarySpec.html)
* [NativeTestSuiteSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.NativeTestSuiteSpec.html)
* [CUnitTestSuiteSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.cunit.CUnitTestSuiteSpec.html)
* [GoogleTestTestSuiteSpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.googletest.GoogleTestTestSuiteSpec.html)
* [NativeBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeBinarySpec.html)
* [NativeExecutableBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeExecutableBinarySpec.html)
* [NativeLibraryBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeLibraryBinarySpec.html)
* [SharedLibraryBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.SharedLibraryBinarySpec.html)
* [StaticLibraryBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.StaticLibraryBinarySpec.html)
* [NativeTestSuiteBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.NativeTestSuiteBinarySpec.html)
* [CUnitTestSuiteBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.cunit.CUnitTestSuiteBinarySpec.html)
* [GoogleTestTestSuiteBinarySpec](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.googletest.GoogleTestTestSuiteBinarySpec.html)
* [NativePlatform](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.platform.NativePlatform.html)
* [BuildType](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.BuildType.html)
* [Flavor](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.Flavor.html)
* [Gcc](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.toolchain.Gcc.html)
* [Clang](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.toolchain.Clang.html)
* [VisualCpp](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.toolchain.VisualCpp.html)
* [AssemblerSourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.assembler.AssemblerSourceSet.html)
* [CSourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.c.CSourceSet.html)
* [CppSourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.cpp.CppSourceSet.html)
* [ObjectiveCSourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.objectivec.ObjectiveCSourceSet.html)
* [ObjectiveCppSourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.objectivecpp.ObjectiveCppSourceSet.html)
* [WindowsResourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.language.rc.WindowsResourceSet.html)
* [VisualStudioProject](https://docs.gradle.org/4.1/dsl/org.gradle.ide.visualstudio.VisualStudioProject.html)
* [VisualStudioSolution](https://docs.gradle.org/4.1/dsl/org.gradle.ide.visualstudio.VisualStudioSolution.html)
* [NativeExecutable](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeExecutable.html)
* [NativeLibrary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeLibrary.html)
* [NativeBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeBinary.html)
* [NativeExecutableBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.NativeExecutableBinary.html)
* [SharedLibraryBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.SharedLibraryBinary.html)
* [StaticLibraryBinary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.StaticLibraryBinary.html)
* **Native component task types**
* [CppCompile](https://docs.gradle.org/4.1/dsl/org.gradle.language.cpp.tasks.CppCompile.html)
* [CCompile](https://docs.gradle.org/4.1/dsl/org.gradle.language.c.tasks.CCompile.html)
* [Assemble](https://docs.gradle.org/4.1/dsl/org.gradle.language.assembler.tasks.Assemble.html)
* [ObjectiveCCompile](https://docs.gradle.org/4.1/dsl/org.gradle.language.objectivec.tasks.ObjectiveCCompile.html)
* [ObjectiveCppCompile](https://docs.gradle.org/4.1/dsl/org.gradle.language.objectivecpp.tasks.ObjectiveCppCompile.html)
* [WindowsResourceCompile](https://docs.gradle.org/4.1/dsl/org.gradle.language.rc.tasks.WindowsResourceCompile.html)
* [LinkExecutable](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.tasks.LinkExecutable.html)
* [LinkSharedLibrary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.tasks.LinkSharedLibrary.html)
* [CreateStaticLibrary](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.tasks.CreateStaticLibrary.html)
* [InstallExecutable](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.tasks.InstallExecutable.html)
* [RunTestExecutable](https://docs.gradle.org/4.1/dsl/org.gradle.nativeplatform.test.tasks.RunTestExecutable.html)

# Project

|  |  |
| --- | --- |
| **API Documentation:** | [Project](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Project.html) |

This interface is the main API you use to interact with Gradle from your build file. From a Project（从Project接口，可以在程序上访问到Gradle的所有属性）, you have programmatic access to all of Gradle's features.

### Lifecycle

There is a one-to-one relationship between a Project and a build.gradle file（Project和build.gradle文件之间是一对一关系）. During build initialisation（构建初始化阶段，如在一个空目执行gradle init）, Gradle assembles a Project object for each project which is to participate in the build, as follows:

* Create a [Settings](https://docs.gradle.org/4.1/dsl/org.gradle.api.initialization.Settings.html) instance for the build.
* Evaluate the settings.gradle script（执行settings.gadle脚本）, if present, against the [Settings](https://docs.gradle.org/4.1/dsl/org.gradle.api.initialization.Settings.html) object to configure it（根据settings对象来配置Project）.
* Use the configured [Settings](https://docs.gradle.org/4.1/dsl/org.gradle.api.initialization.Settings.html) object to create the hierarchy of Project instances.（使用Settings对象来创建Project实例的层次结构）
* Finally, evaluate each Project by executing its build.gradle file, if present, against the project（最后，通过执行build.gadle来构建每个项目）. The projects are evaluated in breadthwise order（横向顺序）, such that a project is evaluated before its child projects（如一个项目在它的自项目之前先构建）. This order can be overridden（这个顺序可以被重写） by calling [Project.evaluationDependsOnChildren()](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Project.html#evaluationDependsOnChildren()) or by adding an explicit evaluation dependency using [Project.evaluationDependsOn(java.lang.String)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:evaluationDependsOn(java.lang.String)).

### Tasks

A project is essentially（本质上） a collection of [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html) objects（Gradle的Project本质上是Task对象的集合）. Each task performs some basic piece of work, such as compiling classes, or running unit tests, or zipping up（打包） a WAR file. You add tasks to a project using one of the create() methods on [TaskContainer](https://docs.gradle.org/4.1/javadoc/org/gradle/api/tasks/TaskContainer.html), such as [TaskContainer.create(java.lang.String)](https://docs.gradle.org/4.1/javadoc/org/gradle/api/tasks/TaskContainer.html#create(java.lang.String)). You can locate existing tasks using one of the lookup methods on [TaskContainer](https://docs.gradle.org/4.1/javadoc/org/gradle/api/tasks/TaskContainer.html), such as [TaskCollection.getByName(java.lang.String)](https://docs.gradle.org/4.1/javadoc/org/gradle/api/tasks/TaskCollection.html#getByName(java.lang.String)).（可以添加任务也可以通过名字定位任务）

### Dependencies

A project generally has a number of dependencies it needs in order to do its work. Also, a project generally produces a number of artifacts（产生许多的jar包）, which other projects can use. Those dependencies are grouped in configurations, and can be retrieved（获取） and uploaded（上传） from repositories. You use the [ConfigurationContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ConfigurationContainer.html) returned by [Project.getConfigurations()](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:configurations) method to manage the configurations. The [DependencyHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html) returned by [Project.getDependencies()](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:dependencies) method to manage the dependencies. The [ArtifactHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.ArtifactHandler.html) returned by [Project.getArtifacts()](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:artifacts) method to manage the artifacts. The [RepositoryHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.RepositoryHandler.html) returned by [Project.getRepositories()](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:repositories) method to manage the repositories.

### Multi-project Builds

（多项目构建）

Projects are arranged into a hierarchy of projects. A project has a name, and a fully qualified path(有效路径) which uniquely identifies it in the hierarchy.

### Plugins

Plugins can be used to modularise（模块化） and reuse project configuration（重用项目的配置）. Plugins can be applied（应用） using the [PluginAware.apply(java.util.Map)](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.PluginAware.html#org.gradle.api.plugins.PluginAware:apply(java.util.Map)) method, or by using the [PluginDependenciesSpec](https://docs.gradle.org/4.1/dsl/org.gradle.plugin.use.PluginDependenciesSpec.html).

### Properties

Gradle executes the project's build file against the Project instance to configure the project（Gradle执行构建脚本来配置项目）. Any property or method which your script uses is delegated（委托） through to the associated Project object. This means, that you can use any of the methods and properties on the Project interface directly in your script（这意味着，你可以在你的脚本中使用任何在Project接口中定义的方法和属性）.

For example:

defaultTasks('some-task') // Delegates to Project.defaultTasks()

reportsDir = file('reports') // Delegates to Project.file() and the Java Plugin

You can also access the Project instance using the project property. This can make the script clearer in some cases. For example, you could use project.name rather than name to access the project's name.（使用Project.name来引用Project的属性比直接使用name能够使脚本更清晰）

A project has 5 property 'scopes'（一个项目包含5个属性命名空间）, which it searches for properties. You can access these properties by name in your build file, or by calling the project's [Project.property(java.lang.String)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:property(java.lang.String)) method. The scopes are:

* The Project object itself（Project自身包含的属性）. This scope includes any property getters and setters declared by the Project implementation class. For example, [Project.getRootProject()](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:rootProject) is accessible as the rootProject property. The properties of this scope are readable or writable depending on the presence of the corresponding getter or setter method.
* The extra properties of the project（项目的扩展属性）. Each project maintains a map of extra properties, which can contain any arbitrary name -> value pair. Once defined, the properties of this scope are readable and writable. See [extra properties](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project.extraproperties) for more details.
* The extensions added to the project by the plugins（通过插件添加进项目的扩展）. Each extension is available as a read-only property with the same name as the extension.
* The convention properties added to the project by the plugins.（通过插件添加进项目的常规属性） A plugin can add properties and methods to a project through the project's [Convention](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/Convention.html) object. The properties of this scope may be readable or writable, depending on the convention objects.
* The tasks of the project（项目的任务）. A task is accessible by using its name as a property name. The properties of this scope are read-only. For example, a task called compile is accessible as the compile property.
* The extra properties and convention properties inherited from the project's parent, recursively up to the root project（继承自父项目的额外属性和传统属性，递归直到根项目）. The properties of this scope are read-only.

When reading a property, the project searches the above scopes in order, and returns the value from the first scope it finds the property in. If not found, an exception is thrown. See [Project.property(java.lang.String)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:property(java.lang.String)) for more details.

When writing a property, the project searches the above scopes in order, and sets the property in the first scope it finds the property in. If not found, an exception is thrown. See [Project.setProperty(java.lang.String, java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:setProperty(java.lang.String, java.lang.Object)) for more details.

#### Extra Properties

额外属性

All extra properties must be defined through the "ext" namespace. Once an extra property has been defined, it is available directly on the owning object (in the below case the Project, Task, and sub-projects respectively) and can be read and updated. Only the initial declaration that needs to be done via the namespace.

project.ext.prop1 = "foo"

task doStuff {

ext.prop2 = "bar"

}

subprojects { ext.${prop3} = false }

Reading extra properties is done through the "ext" or through the owning object.

ext.isSnapshot = version.endsWith("-SNAPSHOT")

if (isSnapshot) {

// do snapshot stuff

}

#### Dynamic Methods

动态方法

A project has 5 method 'scopes'（5个空间）, which it searches for methods:

* The Project object itself.
* The build file. The project searches for a matching method declared in the build file.
* The extensions added to the project by the plugins. Each extension is available as a method which takes a closure or [Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html) as a parameter.
* The convention methods added to the project by the plugins. A plugin can add properties and method to a project through the project's [Convention](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/Convention.html) object.
* The tasks of the project. A method is added for each task, using the name of the task as the method name and taking a single closure or [Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html) parameter. The method calls the [Task.configure(groovy.lang.Closure)](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Task.html#configure(groovy.lang.Closure)) method for the associated task with the provided closure. For example, if the project has a task called compile, then a method is added with the following signature: void compile(Closure configureClosure).
* The methods of the parent project, recursively up to the root project.
* A property of the project whose value is a closure. The closure is treated as a method and called with the provided parameters. The property is located as described above.

### Properties

| **Property** | **Description** |
| --- | --- |
| [allprojects](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:allprojects) | The set containing this project and its subprojects. |
| [ant](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:ant) | The AntBuilder for this project. You can use this in your build file to execute ant tasks. See example below. |
| [artifacts](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:artifacts) | Returns a handler for assigning artifacts produced by the project to configurations. |
| [buildDir](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:buildDir) | The build directory of this project. The build directory is the directory which all artifacts are generated into. The default value for the build directory is projectDir/build |
| [buildFile](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:buildFile) | The build script for this project. |
| [buildscript](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:buildscript) | The build script handler for this project. You can use this handler to query details about the build script for this project, and manage the classpath used to compile and execute the project's build script. |
| [childProjects](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:childProjects) | The direct children of this project. |
| [configurations](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:configurations) | The configurations of this project. |
| [convention](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:convention) | The [Convention](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/Convention.html) for this project. |
| [defaultTasks](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:defaultTasks) | The names of the default tasks of this project. These are used when no tasks names are provided when starting the build. |
| [dependencies](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:dependencies) | The dependency handler of this project. The returned dependency handler instance can be used for adding new dependencies. For accessing already declared dependencies, the configurations can be used. |
| [description](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:description) | The description of this project, if any. |
| [extensions](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:extensions) | Allows adding DSL extensions to the project. Useful for plugin authors. |
| [gradle](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:gradle) | The [Gradle](https://docs.gradle.org/4.1/dsl/org.gradle.api.invocation.Gradle.html) invocation which this project belongs to. |
| [group](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:group) | The group of this project. Gradle always uses the toString() value of the group. The group defaults to the path with dots as separators. |
| [logger](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:logger) | The logger for this project. You can use this in your build file to write log messages. |
| [logging](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:logging) | The [LoggingManager](https://docs.gradle.org/4.1/javadoc/org/gradle/api/logging/LoggingManager.html) which can be used to receive logging and to control the standard output/error capture for this project's build 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. |
| [name](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:name) | The name of this project. The project's name is not necessarily unique within a project hierarchy. You should use the [Project.getPath()](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:path) method for a unique identifier for the project. |
| [normalization](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:normalization) | *incubating*  Provides access to configuring input normalization. |
| [parent](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:parent) | The parent project of this project, if any. |
| [path](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:path) | The path of this project. The path is the fully qualified name of the project. |
| [pluginManager](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:pluginManager) | *incubating*  The plugin manager for this plugin aware object. |
| [plugins](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:plugins) | The container of plugins that have been applied to this object. |
| [project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:project) | Returns this project. This method is useful in build files to explicitly access project properties and methods. For example, using project.name can express your intent better than usingname. This method also allows you to access project properties from a scope where the property may be hidden, such as, for example, from a method or closure. |
| [projectDir](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:projectDir) | The directory containing the project build file. |
| [properties](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:properties) | The properties of this project. See [here](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project.properties) for details of the properties which are available for a project. |
| [repositories](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:repositories) | Returns a handler to create repositories which are used for retrieving dependencies and uploading artifacts produced by the project. |
| [resources](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:resources) | Provides access to resource-specific utility methods, for example factory methods that create various resources. |
| [rootDir](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:rootDir) | The root directory of this project. The root directory is the project directory of the root project. |
| [rootProject](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:rootProject) | The root project for the hierarchy that this project belongs to. In the case of a single-project build, this method returns this project. |
| [state](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:state) | The evaluation state of this project. You can use this to access information about the evaluation of this project, such as whether it has failed. |
| [status](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:status) | The status of this project. Gradle always uses the toString() value of the status. The status defaults to release. |
| [subprojects](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:subprojects) | The set containing the subprojects of this project. |
| [tasks](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:tasks) | The tasks of this project. |
| [version](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:version) | The version of this project. Gradle always uses the toString() value of the version. The version defaults to unspecified. |

#### Properties added by the announce plugin

| **Property** | **Description** |
| --- | --- |
| [announce](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:announce) | The [AnnouncePluginExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.announce.AnnouncePluginExtension.html) added by the announce plugin. |

#### Properties added by the application plugin

| **Property** | **Description** |
| --- | --- |
| [applicationDefaultJvmArgs](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:applicationDefaultJvmArgs) | Array of string arguments to pass to the JVM when running the application |
| [applicationDistribution](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:applicationDistribution) | The specification of the contents of the distribution. |
| [applicationName](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:applicationName) | The name of the application. |
| [mainClassName](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:mainClassName) | The fully qualified name of the application's main class. |

#### Properties added by the checkstyle plugin

| **Property** | **Description** |
| --- | --- |
| [checkstyle](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:checkstyle) | The [CheckstyleExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.CheckstyleExtension.html) added by the checkstyle plugin. |

#### Properties added by the codenarc plugin

| **Property** | **Description** |
| --- | --- |
| [codenarc](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:codenarc) | The [CodeNarcExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.CodeNarcExtension.html) added by the codenarc plugin. |

#### Properties added by the distribution plugin

| **Property** | **Description** |
| --- | --- |
| [distributions](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:distributions) | The [DistributionContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.distribution.DistributionContainer.html) added by the distribution plugin. |

#### Properties added by the ear plugin

| **Property** | **Description** |
| --- | --- |
| [appDirName](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:appDirName) | The name of the application directory, relative to the project directory. Default is "src/main/application". |
| [deploymentDescriptor](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:deploymentDescriptor) | A custom deployment descriptor configuration. Default is an "application.xml" with sensible defaults. |
| [libDirName](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:libDirName) | The name of the library directory in the EAR file. Default is "lib". |

#### Properties added by the eclipse plugin

| **Property** | **Description** |
| --- | --- |
| [eclipse](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:eclipse) | The [EclipseModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseModel.html) added by the eclipse plugin. |

#### Properties added by the findbugs plugin

| **Property** | **Description** |
| --- | --- |
| [findbugs](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:findbugs) | The [FindBugsExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.FindBugsExtension.html) added by the findbugs plugin. |

#### Properties added by the idea plugin

| **Property** | **Description** |
| --- | --- |
| [idea](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:idea) | The [IdeaModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaModel.html) added by the idea plugin. |

#### Properties added by the jacoco plugin

| **Property** | **Description** |
| --- | --- |
| [jacoco](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:jacoco) | The [JacocoPluginExtension](https://docs.gradle.org/4.1/dsl/org.gradle.testing.jacoco.plugins.JacocoPluginExtension.html) added by the jacoco plugin. |

#### Properties added by the java plugin

| **Property** | **Description** |
| --- | --- |
| [archivesBaseName](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:archivesBaseName) | The base name to use for archive files. |
| [distsDir](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:distsDir) | The directory to generate TAR and ZIP archives into. |
| [distsDirName](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:distsDirName) | The name for the distributions directory. This in interpreted relative to the project' build directory. |
| [docsDir](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:docsDir) | Returns a file pointing to the root directory supposed to be used for all docs. |
| [docsDirName](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:docsDirName) | The name of the docs directory. Can be a name or a path relative to the build dir. |
| [libsDir](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:libsDir) | The directory to generate JAR and WAR archives into. |
| [libsDirName](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:libsDirName) | The name for the libs directory. This in interpreted relative to the project' build directory. |
| [reporting](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:reporting) | The [ReportingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.ReportingExtension.html) added by the java plugin. |
| [sourceCompatibility](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:sourceCompatibility) | The source compatibility used for compiling Java sources. |
| [sourceSets](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:sourceSets) | The source sets container. |
| [targetCompatibility](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:targetCompatibility) | The target compatibility used for compiling Java sources. |
| [testReportDir](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:testReportDir) | Returns a file pointing to the root directory to be used for reports. |
| [testReportDirName](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:testReportDirName) | The name of the test reports directory. Can be a name or a path relative to [ReportingExtension.getBaseDir()](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.ReportingExtension.html#org.gradle.api.reporting.ReportingExtension:baseDir). |
| [testResultsDir](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:testResultsDir) | Returns a file pointing to the root directory of the test results. |
| [testResultsDirName](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:testResultsDirName) | The name of the test results directory. Can be a name or a path relative to the build dir. |

#### Properties added by the jdepend plugin

| **Property** | **Description** |
| --- | --- |
| [jdepend](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:jdepend) | The [JDependExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.JDependExtension.html) added by the jdepend plugin. |

#### Properties added by the maven plugin

| **Property** | **Description** |
| --- | --- |
| [conf2ScopeMappings](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:conf2ScopeMappings) | The set of rules for how to map Gradle dependencies to Maven scopes. |
| [mavenPomDir](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:mavenPomDir) | The directory to generate Maven POMs into. |

#### Properties added by the pmd plugin

| **Property** | **Description** |
| --- | --- |
| [pmd](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:pmd) | The [PmdExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.PmdExtension.html) added by the pmd plugin. |

#### Properties added by the project-report plugin

| **Property** | **Description** |
| --- | --- |
| [projectReportDir](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:projectReportDir) | The directory to generate the project reports into. |
| [projectReportDirName](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:projectReportDirName) | The name of the directory to generate the project reports into, relative to the project's reports dir. |

#### Properties added by the publishing plugin

| **Property** | **Description** |
| --- | --- |
| [publishing](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:publishing) | The [PublishingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.PublishingExtension.html) added by the publishing plugin. |

#### Properties added by the signing plugin

| **Property** | **Description** |
| --- | --- |
| [signing](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:signing) | The [SigningExtension](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.signing.SigningExtension.html) added by the signing plugin. |

#### Properties added by the visual-studio plugin

| **Property** | **Description** |
| --- | --- |
| [visualStudio](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:visualStudio) | The [VisualStudioExtension](https://docs.gradle.org/4.1/dsl/org.gradle.ide.visualstudio.VisualStudioExtension.html) added by the visual-studio plugin. |

#### Properties added by the war plugin

| **Property** | **Description** |
| --- | --- |
| [webAppDir](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:webAppDir) | The web application directory. |
| [webAppDirName](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:webAppDirName) | The name of the web application directory, relative to the project directory. |

### Methods

| **Method** | **Description** |
| --- | --- |
| [absoluteProjectPath](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:absoluteProjectPath(java.lang.String))(path) | Converts a name to an absolute project path, resolving names relative to this project. |
| [afterEvaluate](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:afterEvaluate(groovy.lang.Closure))(closure) | Adds a closure to be called immediately after this project has been evaluated. The project is passed to the closure as a parameter. Such a listener gets notified when the build file belonging to this project has been executed. A parent project may for example add such a listener to its child project. Such a listener can further configure those child projects based on the state of the child projects after their build files have been run. |
| [afterEvaluate](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:afterEvaluate(org.gradle.api.Action))(action) | Adds an action to execute immediately after this project is evaluated. |
| [allprojects](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:allprojects(org.gradle.api.Action))(action) | Configures this project and each of its sub-projects. |
| [ant](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:ant(org.gradle.api.Action))(configureAction) | Executes the given action against the AntBuilder for this project. You can use this in your build file to execute ant tasks. See example in javadoc for [Project.getAnt()](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:ant) |
| [apply](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:apply(groovy.lang.Closure))(closure) | Applies zero or more plugins or scripts. |
| [apply](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project: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](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:apply(org.gradle.api.Action))(action) | Applies zero or more plugins or scripts. |
| [artifacts](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:artifacts(org.gradle.api.Action))(configureAction) | Configures the published artifacts for this project. |
| [beforeEvaluate](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:beforeEvaluate(groovy.lang.Closure))(closure) | Adds a closure to be called immediately before this project is evaluated. The project is passed to the closure as a parameter. |
| [beforeEvaluate](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:beforeEvaluate(org.gradle.api.Action))(action) | Adds an action to execute immediately before this project is evaluated. |
| [configure](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:configure(java.lang.Iterable, groovy.lang.Closure))(objects, configureClosure) | Configures a collection of objects via a closure. This is equivalent to calling [Project.configure(java.lang.Object, groovy.lang.Closure)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:configure(java.lang.Object, groovy.lang.Closure)) for each of the given objects. |
| [configure](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:configure(java.lang.Iterable, org.gradle.api.Action))(objects, configureAction) | Configures a collection of objects via an action. |
| [configure](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:configure(java.lang.Object, groovy.lang.Closure))(object, configureClosure) | Configures an object via a closure, with the closure's delegate set to the supplied object. This way you don't have to specify the context of a configuration statement multiple times. |
| [container](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:container(java.lang.Class))(type) | Creates a container for managing named objects of the specified type. The specified type must have a public constructor which takes the name as a String parameter. |
| [container](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:container(java.lang.Class, groovy.lang.Closure))(type, factoryClosure) | Creates a container for managing named objects of the specified type. The given closure is used to create object instances. The name of the instance to be created is passed as a parameter to the closure. |
| [container](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:container(java.lang.Class, org.gradle.api.NamedDomainObjectFactory))(type, factory) | Creates a container for managing named objects of the specified type. The given factory is used to create object instances. |
| [copy](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:copy(groovy.lang.Closure))(closure) | Copies the specified files. The given closure is used to configure a [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html), which is then used to copy the files. Example: |
| [copy](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:copy(org.gradle.api.Action))(action) | Copies the specified files. The given action is used to configure a [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html), which is then used to copy the files. |
| [copySpec](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:copySpec())() | Creates a [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html) which can later be used to copy files or create an archive. |
| [copySpec](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:copySpec(groovy.lang.Closure))(closure) | Creates a [CopySpec](https://docs.gradle.org/4.1/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](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html) before it is returned by this method. |
| [copySpec](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:copySpec(org.gradle.api.Action))(action) | Creates a [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html) which can later be used to copy files or create an archive. The given action is used to configure the [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html) before it is returned by this method. |
| [delete](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:delete(java.lang.Object[]))(paths) | Deletes files and directories. |
| [delete](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:delete(org.gradle.api.Action))(action) | Deletes the specified files. The given action is used to configure a [DeleteSpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/DeleteSpec.html), which is then used to delete the files. |
| [evaluationDependsOn](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:evaluationDependsOn(java.lang.String))(path) | Declares that this project has an evaluation dependency on the project with the given path. |
| [exec](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:exec(groovy.lang.Closure))(closure) | Executes an external command. The closure configures a [ExecSpec](https://docs.gradle.org/4.1/javadoc/org/gradle/process/ExecSpec.html). |
| [exec](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:exec(org.gradle.api.Action))(action) | Executes an external command. |
| [file](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object))(path) | Resolves a file path relative to the project directory of this project. This method converts the supplied path based on its type: |
| [file](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object, org.gradle.api.PathValidation))(path, validation) | Resolves a file path relative to the project directory of this project and validates it using the given scheme. See [PathValidation](https://docs.gradle.org/4.1/javadoc/org/gradle/api/PathValidation.html) for the list of possible validations. |
| [fileTree](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:fileTree(java.lang.Object))(baseDir) | Creates a new ConfigurableFileTree using the given base directory. The given baseDir path is evaluated as per [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object)). |
| [fileTree](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:fileTree(java.lang.Object, groovy.lang.Closure))(baseDir, configureClosure) | Creates a new ConfigurableFileTree using the given base directory. The given baseDir path is evaluated as per [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project: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](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:fileTree(java.lang.Object, org.gradle.api.Action))(baseDir, configureAction) | Creates a new ConfigurableFileTree using the given base directory. The given baseDir path is evaluated as per [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object)). The action will be used to configure the new file tree. Example: |
| [fileTree](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project: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](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:files(java.lang.Object, groovy.lang.Closure))(paths, configureClosure) | Creates a new ConfigurableFileCollection using the given paths. The paths are evaluated as per [Project.files(java.lang.Object[])](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:files(java.lang.Object[])). The file collection is configured using the given closure. The file collection is passed to the closure as its delegate. Example: |
| [files](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:files(java.lang.Object, org.gradle.api.Action))(paths, configureAction) | Creates a new ConfigurableFileCollection using the given paths. The paths are evaluated as per [Project.files(java.lang.Object[])](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:files(java.lang.Object[])). The file collection is configured using the given action. Example: |
| [files](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:files(java.lang.Object[]))(paths) | Returns a [ConfigurableFileCollection](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/ConfigurableFileCollection.html) containing the given files. You can pass any of the following types to this method: |
| [findProject](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:findProject(java.lang.String))(path) | Locates a project by path. If the path is relative, it is interpreted relative to this project. |
| [findProperty](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:findProperty(java.lang.String))(propertyName) | *incubating*  Returns the value of the given property or null if not found. This method locates a property as follows: |
| [getAllTasks](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:getAllTasks(boolean))(recursive) | Returns a map of the tasks contained in this project, and optionally its subprojects. |
| [getTasksByName](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:getTasksByName(java.lang.String, boolean))(name, recursive) | Returns the set of tasks with the given name contained in this project, and optionally its subprojects. |
| [hasProperty](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:hasProperty(java.lang.String))(propertyName) | Determines if this project has the given property. See [here](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project.properties) for details of the properties which are available for a project. |
| [javaexec](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:javaexec(groovy.lang.Closure))(closure) | Executes a Java main class. The closure configures a [JavaExecSpec](https://docs.gradle.org/4.1/javadoc/org/gradle/process/JavaExecSpec.html). |
| [javaexec](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:javaexec(org.gradle.api.Action))(action) | Executes an external Java process. |
| [mkdir](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:mkdir(java.lang.Object))(path) | Creates a directory and returns a file pointing to it. |
| [normalization](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:normalization(org.gradle.api.Action))(configuration) | *incubating*  Configures input normalization. |
| [project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:project(java.lang.String))(path) | Locates a project by path. If the path is relative, it is interpreted relative to this project. |
| [project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:project(java.lang.String, groovy.lang.Closure))(path, configureClosure) | Locates a project by path and configures it using the given closure. If the path is relative, it is interpreted relative to this project. The target project is passed to the closure as the closure's delegate. |
| [project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:project(java.lang.String, org.gradle.api.Action))(path, configureAction) | Locates a project by path and configures it using the given action. If the path is relative, it is interpreted relative to this project. |
| [property](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:property(java.lang.Class))(clazz) | *incubating*  Creates a PropertyState implementation based on the provided class. |
| [property](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:property(java.lang.String))(propertyName) | Returns the value of the given property. This method locates a property as follows: |
| [relativePath](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:relativePath(java.lang.Object))(path) | Returns the relative path from the project directory to the given path. The given path object is (logically) resolved as described for [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object)), from which a relative path is calculated. |
| [relativeProjectPath](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:relativeProjectPath(java.lang.String))(path) | Converts a name to a project path relative to this project. |
| [setProperty](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:setProperty(java.lang.String, java.lang.Object))(name, value) | Sets a property of this project. This method searches for a property with the given name in the following locations, and sets the property on the first location where it finds the property. |
| [subprojects](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:subprojects(org.gradle.api.Action))(action) | Configures the sub-projects of this project |
| [sync](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:sync(org.gradle.api.Action))(action) | Synchronizes the contents of a destination directory with some source directories and files. The given action is used to configure a [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html), which is then used to synchronize the files. |
| [tarTree](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:tarTree(java.lang.Object))(tarPath) | Creates a new FileTree which contains the contents of the given TAR file. The given tarPath path can be: |
| [task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:task(java.lang.String))(name) | Creates a [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html) with the given name and adds it to this project. Calling this method is equivalent to calling [Project.task(java.util.Map, java.lang.String)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:task(java.util.Map, java.lang.String)) with an empty options map. |
| [task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:task(java.lang.String, groovy.lang.Closure))(name, configureClosure) | Creates a [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html) with the given name and adds it to this project. Before the task is returned, the given closure is executed to configure the task. |
| [task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:task(java.util.Map, java.lang.String))(args, name) | Creates a [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html) with the given name and adds it to this project. A map of creation options can be passed to this method to control how the task is created. The following options are available: |
| [task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:task(java.util.Map, java.lang.String, groovy.lang.Closure))(args, name, configureClosure) | Creates a [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html) with the given name and adds it to this project. Before the task is returned, the given closure is executed to configure the task. A map of creation options can be passed to this method to control how the task is created. See [Project.task(java.util.Map, java.lang.String)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:task(java.util.Map, java.lang.String)) for the available options. |
| [uri](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:uri(java.lang.Object))(path) | Resolves a file path to a URI, relative to the project directory of this project. Evaluates the provided path object as described for [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object)), with the exception that any URI scheme is supported, not just 'file:' URIs. |
| [zipTree](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project: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 [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object)). You can combine this method with the [Project.copy(groovy.lang.Closure)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:copy(groovy.lang.Closure)) method to unzip a ZIP file. |

#### Methods added by the ear plugin

| **Method** | **Description** |
| --- | --- |
| [appDirName](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:appDirName(java.lang.String))(appDirName) | Allows changing the application directory. Default is "src/main/application". |
| [deploymentDescriptor](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:deploymentDescriptor(org.gradle.api.Action))(configureAction) | Configures the deployment descriptor for this EAR archive. |
| [libDirName](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:libDirName(java.lang.String))(libDirName) | Allows changing the library directory in the EAR file. Default is "lib". |

#### Methods added by the java plugin

| **Method** | **Description** |
| --- | --- |
| [manifest](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:manifest())() | Creates a new instance of a [Manifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/java/archives/Manifest.html). |
| [manifest](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:manifest(groovy.lang.Closure))(closure) | Creates and configures a new instance of a [Manifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/java/archives/Manifest.html). The given closure configures the new manifest instance before it is returned. |
| [manifest](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:manifest(org.gradle.api.Action))(action) | Creates and configures a new instance of a [Manifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/java/archives/Manifest.html). |

#### Methods added by the maven plugin

| **Method** | **Description** |
| --- | --- |
| [pom](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:pom())() | Creates a new [MavenPom](https://docs.gradle.org/4.1/javadoc/org/gradle/api/artifacts/maven/MavenPom.html). |
| [pom](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:pom(groovy.lang.Closure))(configureClosure) | Creates and configures a new [MavenPom](https://docs.gradle.org/4.1/javadoc/org/gradle/api/artifacts/maven/MavenPom.html). The given closure is executed to configure the new POM instance. |

#### Methods added by the osgi plugin

| **Method** | **Description** |
| --- | --- |
| [osgiManifest](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:osgiManifest())() | Creates a new instance of [OsgiManifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/osgi/OsgiManifest.html). The returned object is preconfigured with: |
| [osgiManifest](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:osgiManifest(groovy.lang.Closure))(closure) | Creates and configures a new instance of an [OsgiManifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/osgi/OsgiManifest.html) . The closure configures the new manifest instance before it is returned. |
| [osgiManifest](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:osgiManifest(org.gradle.api.Action))(action) | Creates and configures a new instance of an [OsgiManifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/osgi/OsgiManifest.html). The action configures the new manifest instance before it is returned. |

### Script blocks

| **Block** | **Description** |
| --- | --- |
| [allprojects](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:allprojects(groovy.lang.Closure)) | Configures this project and each of its sub-projects. |
| [ant](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:ant(groovy.lang.Closure)) | Executes the given closure against the AntBuilder for this project. You can use this in your build file to execute ant tasks. The AntBuilder is passed to the closure as the closure's delegate. See example in javadoc for [Project.getAnt()](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:ant) |
| [artifacts](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:artifacts(groovy.lang.Closure)) | Configures the published artifacts for this project. |
| [buildscript](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:buildscript(groovy.lang.Closure)) | Configures the build script classpath for this project. |
| [configurations](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:configurations(groovy.lang.Closure)) | Configures the dependency configurations for this project. |
| [dependencies](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:dependencies(groovy.lang.Closure)) | Configures the dependencies for this project. |
| [repositories](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:repositories(groovy.lang.Closure)) | Configures the repositories for this project. |
| [subprojects](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:subprojects(groovy.lang.Closure)) | Configures the sub-projects of this project. |

#### Script blocks added by the announce plugin

| **Block** | **Description** |
| --- | --- |
| [announce](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:announce(groovy.lang.Closure)) | Configures the [AnnouncePluginExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.announce.AnnouncePluginExtension.html) added by the announce plugin. |

#### Script blocks added by the checkstyle plugin

| **Block** | **Description** |
| --- | --- |
| [checkstyle](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:checkstyle(groovy.lang.Closure)) | Configures the [CheckstyleExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.CheckstyleExtension.html) added by the checkstyle plugin. |

#### Script blocks added by the codenarc plugin

| **Block** | **Description** |
| --- | --- |
| [codenarc](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:codenarc(groovy.lang.Closure)) | Configures the [CodeNarcExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.CodeNarcExtension.html) added by the codenarc plugin. |

#### Script blocks added by the distribution plugin

| **Block** | **Description** |
| --- | --- |
| [distributions](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:distributions(groovy.lang.Closure)) | Configures the [DistributionContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.distribution.DistributionContainer.html) added by the distribution plugin. |

#### Script blocks added by the ear plugin

| **Block** | **Description** |
| --- | --- |
| [deploymentDescriptor](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:deploymentDescriptor(groovy.lang.Closure)) | Configures the deployment descriptor for this EAR archive. |

#### Script blocks added by the eclipse plugin

| **Block** | **Description** |
| --- | --- |
| [eclipse](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:eclipse(groovy.lang.Closure)) | Configures the [EclipseModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseModel.html) added by the eclipse plugin. |

#### Script blocks added by the findbugs plugin

| **Block** | **Description** |
| --- | --- |
| [findbugs](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:findbugs(groovy.lang.Closure)) | Configures the [FindBugsExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.FindBugsExtension.html) added by the findbugs plugin. |

#### Script blocks added by the idea plugin

| **Block** | **Description** |
| --- | --- |
| [idea](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:idea(groovy.lang.Closure)) | Configures the [IdeaModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaModel.html) added by the idea plugin. |

#### Script blocks added by the jacoco plugin

| **Block** | **Description** |
| --- | --- |
| [jacoco](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:jacoco(groovy.lang.Closure)) | Configures the [JacocoPluginExtension](https://docs.gradle.org/4.1/dsl/org.gradle.testing.jacoco.plugins.JacocoPluginExtension.html) added by the jacoco plugin. |

#### Script blocks added by the java plugin

| **Block** | **Description** |
| --- | --- |
| [reporting](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:reporting(groovy.lang.Closure)) | Configures the [ReportingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.ReportingExtension.html) added by the java plugin. |
| [sourceSets](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:sourceSets(groovy.lang.Closure)) | Configures the source sets of this project. |

#### Script blocks added by the jdepend plugin

| **Block** | **Description** |
| --- | --- |
| [jdepend](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:jdepend(groovy.lang.Closure)) | Configures the [JDependExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.JDependExtension.html) added by the jdepend plugin. |

#### Script blocks added by the pmd plugin

| **Block** | **Description** |
| --- | --- |
| [pmd](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:pmd(groovy.lang.Closure)) | Configures the [PmdExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.PmdExtension.html) added by the pmd plugin. |

#### Script blocks added by the publishing plugin

| **Block** | **Description** |
| --- | --- |
| [publishing](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:publishing(groovy.lang.Closure)) | Configures the [PublishingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.PublishingExtension.html) added by the publishing plugin. |

#### Script blocks added by the signing plugin

| **Block** | **Description** |
| --- | --- |
| [signing](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:signing(groovy.lang.Closure)) | Configures the [SigningExtension](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.signing.SigningExtension.html) added by the signing plugin. |

#### Script blocks added by the visual-studio plugin

| **Block** | **Description** |
| --- | --- |
| [visualStudio](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:visualStudio(groovy.lang.Closure)) | Configures the [VisualStudioExtension](https://docs.gradle.org/4.1/dsl/org.gradle.ide.visualstudio.VisualStudioExtension.html) added by the visual-studio plugin. |

### Property details

#### [Set](http://download.oracle.com/javase/7/docs/api/java/util/Set.html" \t "_top)<[Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html)> allprojects (read-only)

The set containing this project and its subprojects.

#### [AntBuilder](https://docs.gradle.org/4.1/javadoc/org/gradle/api/AntBuilder.html" \t "_top) ant (read-only)

The AntBuilder for this project. You can use this in your build file to execute ant tasks. See example below.

task printChecksum {

doLast {

ant {

//using ant checksum task to store the file checksum in the checksumOut ant property

checksum(property: 'checksumOut', file: 'someFile.txt')

//we can refer to the ant property created by checksum task:

println "The checksum is: " + checksumOut

}

//we can refer to the ant property later as well:

println "I just love to print checksums: " + ant.checksumOut

}

}

Consider following example of ant target:

<target name='printChecksum'>

<checksum property='checksumOut'>

<fileset dir='.'>

<include name='agile.txt'/>

</fileset>

</checksum>

<echo>The checksum is: ${checksumOut}</echo>

</target>

Here's how it would look like in gradle. Observe how the ant XML is represented in groovy by the ant builder

task printChecksum {

doLast {

ant {

checksum(property: 'checksumOut') {

fileset(dir: '.') {

include name: 'agile1.txt'

}

}

}

logger.lifecycle("The checksum is $ant.checksumOut")

}

}

#### [ArtifactHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.ArtifactHandler.html" \t "_top) artifacts (read-only)

Returns a handler for assigning artifacts produced by the project to configurations.

##### Examples:

See docs for [ArtifactHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.ArtifactHandler.html)

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html" \t "_top) buildDir

The build directory of this project. The build directory is the directory which all artifacts are generated into. The default value for the build directory is projectDir/build

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html" \t "_top) buildFile (read-only)

The build script for this project.

If the file exists, it will be evaluated against this project when this project is configured.

#### [ScriptHandler](https://docs.gradle.org/4.1/javadoc/org/gradle/api/initialization/dsl/ScriptHandler.html" \t "_top) buildscript (read-only)

The build script handler for this project. You can use this handler to query details about the build script for this project, and manage the classpath used to compile and execute the project's build script.

#### [Map](http://download.oracle.com/javase/7/docs/api/java/util/Map.html" \t "_top)<[String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html), [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html)> childProjects (read-only)

The direct children of this project.

#### [ConfigurationContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ConfigurationContainer.html" \t "_top) configurations (read-only)

The configurations of this project.

##### Examples:

See docs for [ConfigurationContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ConfigurationContainer.html)

#### [Convention](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/Convention.html" \t "_top) convention (read-only)

The [Convention](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/Convention.html) for this project.

You can access this property in your build file using convention. You can also can also access the properties and methods of the convention object as if they were properties and methods of this project. See [here](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project.properties) for more details

#### [List](http://download.oracle.com/javase/7/docs/api/java/util/List.html" \t "_top)<[String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html)> defaultTasks

The names of the default tasks of this project. These are used when no tasks names are provided when starting the build.

#### [DependencyHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html" \t "_top) dependencies (read-only)

The dependency handler of this project. The returned dependency handler instance can be used for adding new dependencies. For accessing already declared dependencies, the configurations can be used.

##### Examples:

See docs for [DependencyHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html)

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) description

The description of this project, if any.

#### [ExtensionContainer](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/ExtensionContainer.html" \t "_top) extensions (read-only)

Allows adding DSL extensions to the project. Useful for plugin authors.

#### [Gradle](https://docs.gradle.org/4.1/dsl/org.gradle.api.invocation.Gradle.html" \t "_top) gradle (read-only)

The [Gradle](https://docs.gradle.org/4.1/dsl/org.gradle.api.invocation.Gradle.html) invocation which this project belongs to.

#### [Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top) group

The group of this project. Gradle always uses the toString() value of the group. The group defaults to the path with dots as separators.

#### [Logger](https://docs.gradle.org/4.1/javadoc/org/gradle/api/logging/Logger.html" \t "_top) logger (read-only)

The logger for this project. You can use this in your build file to write log messages.

#### [LoggingManager](https://docs.gradle.org/4.1/javadoc/org/gradle/api/logging/LoggingManager.html" \t "_top) logging (read-only)

The [LoggingManager](https://docs.gradle.org/4.1/javadoc/org/gradle/api/logging/LoggingManager.html) which can be used to receive logging and to control the standard output/error capture for this project's build 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.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) name (read-only)

The name of this project. The project's name is not necessarily unique within a project hierarchy. You should use the [Project.getPath()](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:path) method for a unique identifier for the project.

#### [InputNormalizationHandler](https://docs.gradle.org/4.1/dsl/org.gradle.normalization.InputNormalizationHandler.html" \t "_top) normalization (read-only)

*Note: This property is*[*incubating*](https://docs.gradle.org/4.1/userguide/feature_lifecycle.html)*and may change in a future version of Gradle.*

Provides access to configuring input normalization.

#### [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html" \t "_top) parent (read-only)

The parent project of this project, if any.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) path (read-only)

The path of this project. The path is the fully qualified name of the project.

#### [PluginManager](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.PluginManager.html" \t "_top) pluginManager (read-only)

*Note: This property is*[*incubating*](https://docs.gradle.org/4.1/userguide/feature_lifecycle.html)*and may change in a future version of Gradle.*

The plugin manager for this plugin aware object.

#### [PluginContainer](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/PluginContainer.html" \t "_top) 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()](https://docs.gradle.org/4.1/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()](https://docs.gradle.org/4.1/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)](https://docs.gradle.org/4.1/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.

#### [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html" \t "_top) project (read-only)

Returns this project. This method is useful in build files to explicitly access project properties and methods. For example, using project.name can express your intent better than using name. This method also allows you to access project properties from a scope where the property may be hidden, such as, for example, from a method or closure.

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html" \t "_top) projectDir (read-only)

The directory containing the project build file.

#### [Map](http://download.oracle.com/javase/7/docs/api/java/util/Map.html" \t "_top)<[String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html), ?> properties (read-only)

The properties of this project. See [here](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project.properties) for details of the properties which are available for a project.

#### [RepositoryHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.RepositoryHandler.html" \t "_top) repositories (read-only)

Returns a handler to create repositories which are used for retrieving dependencies and uploading artifacts produced by the project.

#### [ResourceHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.resources.ResourceHandler.html" \t "_top) resources (read-only)

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

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html" \t "_top) rootDir (read-only)

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

#### [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html" \t "_top) rootProject (read-only)

The root project for the hierarchy that this project belongs to. In the case of a single-project build, this method returns this project.

#### [ProjectState](https://docs.gradle.org/4.1/javadoc/org/gradle/api/ProjectState.html" \t "_top) state (read-only)

The evaluation state of this project. You can use this to access information about the evaluation of this project, such as whether it has failed.

#### [Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top) status

The status of this project. Gradle always uses the toString() value of the status. The status defaults to release.

The status of the project is only relevant, if you upload libraries together with a module descriptor. The status specified here, will be part of this module descriptor.

#### [Set](http://download.oracle.com/javase/7/docs/api/java/util/Set.html" \t "_top)<[Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html)> subprojects (read-only)

The set containing the subprojects of this project.

#### [TaskContainer](https://docs.gradle.org/4.1/javadoc/org/gradle/api/tasks/TaskContainer.html" \t "_top) tasks (read-only)

The tasks of this project.

#### [Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top) version

The version of this project. Gradle always uses the toString() value of the version. The version defaults to unspecified.

#### [AnnouncePluginExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.announce.AnnouncePluginExtension.html" \t "_top) announce (read-only)

The [AnnouncePluginExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.announce.AnnouncePluginExtension.html) added by the announce plugin.

#### [Iterable](http://download.oracle.com/javase/7/docs/api/java/lang/Iterable.html" \t "_top)<[String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html)> applicationDefaultJvmArgs

Array of string arguments to pass to the JVM when running the application

#### [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html" \t "_top) applicationDistribution

The specification of the contents of the distribution.

Use this [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html) to include extra files/resource in the application distribution.

apply plugin: 'application'

applicationDistribution.from("some/dir") {

include "\*.txt"

}

Note that the application plugin pre configures this spec to; include the contents of "src/dist", copy the application start scripts into the "bin" directory, and copy the built jar and its dependencies into the "lib" directory.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) applicationName

The name of the application.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) mainClassName

The fully qualified name of the application's main class.

#### [CheckstyleExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.CheckstyleExtension.html" \t "_top) checkstyle (read-only)

The [CheckstyleExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.CheckstyleExtension.html) added by the checkstyle plugin.

#### [CodeNarcExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.CodeNarcExtension.html" \t "_top) codenarc (read-only)

The [CodeNarcExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.CodeNarcExtension.html) added by the codenarc plugin.

#### [DistributionContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.distribution.DistributionContainer.html" \t "_top) distributions (read-only)

The [DistributionContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.distribution.DistributionContainer.html) added by the distribution plugin.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) appDirName

The name of the application directory, relative to the project directory. Default is "src/main/application".

#### [DeploymentDescriptor](https://docs.gradle.org/4.1/javadoc/org/gradle/plugins/ear/descriptor/DeploymentDescriptor.html) deploymentDescriptor

A custom deployment descriptor configuration. Default is an "application.xml" with sensible defaults.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) libDirName

The name of the library directory in the EAR file. Default is "lib".

#### [EclipseModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseModel.html" \t "_top) eclipse (read-only)

The [EclipseModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseModel.html) added by the eclipse plugin.

#### [FindBugsExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.FindBugsExtension.html" \t "_top) findbugs (read-only)

The [FindBugsExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.FindBugsExtension.html) added by the findbugs plugin.

#### [IdeaModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaModel.html" \t "_top) idea (read-only)

The [IdeaModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaModel.html) added by the idea plugin.

#### [JacocoPluginExtension](https://docs.gradle.org/4.1/dsl/org.gradle.testing.jacoco.plugins.JacocoPluginExtension.html" \t "_top) jacoco (read-only)

The [JacocoPluginExtension](https://docs.gradle.org/4.1/dsl/org.gradle.testing.jacoco.plugins.JacocoPluginExtension.html) added by the jacoco plugin.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) archivesBaseName

The base name to use for archive files.

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html" \t "_top) distsDir (read-only)

The directory to generate TAR and ZIP archives into.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) distsDirName

The name for the distributions directory. This in interpreted relative to the project' build directory.

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html" \t "_top) docsDir (read-only)

Returns a file pointing to the root directory supposed to be used for all docs.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) docsDirName

The name of the docs directory. Can be a name or a path relative to the build dir.

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html" \t "_top) libsDir (read-only)

The directory to generate JAR and WAR archives into.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) libsDirName

The name for the libs directory. This in interpreted relative to the project' build directory.

#### [ReportingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.ReportingExtension.html" \t "_top) reporting (read-only)

The [ReportingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.ReportingExtension.html) added by the java plugin.

#### [JavaVersion](https://docs.gradle.org/4.1/javadoc/org/gradle/api/JavaVersion.html" \t "_top) sourceCompatibility

The source compatibility used for compiling Java sources.

#### [SourceSetContainer](https://docs.gradle.org/4.1/javadoc/org/gradle/api/tasks/SourceSetContainer.html" \t "_top) sourceSets (read-only)

The source sets container.

#### [JavaVersion](https://docs.gradle.org/4.1/javadoc/org/gradle/api/JavaVersion.html" \t "_top) targetCompatibility

The target compatibility used for compiling Java sources.

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html" \t "_top) testReportDir (read-only)

Returns a file pointing to the root directory to be used for reports.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) testReportDirName

The name of the test reports directory. Can be a name or a path relative to [ReportingExtension.getBaseDir()](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.ReportingExtension.html#org.gradle.api.reporting.ReportingExtension:baseDir).

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html" \t "_top) testResultsDir (read-only)

Returns a file pointing to the root directory of the test results.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) testResultsDirName

The name of the test results directory. Can be a name or a path relative to the build dir.

#### [JDependExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.JDependExtension.html" \t "_top) jdepend (read-only)

The [JDependExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.JDependExtension.html) added by the jdepend plugin.

#### [Conf2ScopeMappingContainer](https://docs.gradle.org/4.1/javadoc/org/gradle/api/artifacts/maven/Conf2ScopeMappingContainer.html" \t "_top) conf2ScopeMappings

The set of rules for how to map Gradle dependencies to Maven scopes.

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html" \t "_top) mavenPomDir

The directory to generate Maven POMs into.

#### [PmdExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.PmdExtension.html" \t "_top) pmd (read-only)

The [PmdExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.PmdExtension.html) added by the pmd plugin.

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html" \t "_top) projectReportDir (read-only)

The directory to generate the project reports into.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) projectReportDirName

The name of the directory to generate the project reports into, relative to the project's reports dir.

#### [PublishingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.PublishingExtension.html" \t "_top) publishing (read-only)

The [PublishingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.PublishingExtension.html) added by the publishing plugin.

#### [SigningExtension](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.signing.SigningExtension.html" \t "_top) signing (read-only)

The [SigningExtension](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.signing.SigningExtension.html) added by the signing plugin.

#### [VisualStudioExtension](https://docs.gradle.org/4.1/dsl/org.gradle.ide.visualstudio.VisualStudioExtension.html" \t "_top) visualStudio (read-only)

The [VisualStudioExtension](https://docs.gradle.org/4.1/dsl/org.gradle.ide.visualstudio.VisualStudioExtension.html) added by the visual-studio plugin.

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html" \t "_top) webAppDir (read-only)

The web application directory.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) webAppDirName

The name of the web application directory, relative to the project directory.

### Method details

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) absoluteProjectPath([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) path)

Converts a name to an absolute project path, resolving names relative to this project.

#### void afterEvaluate([Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html" \t "_top) closure)

Adds a closure to be called immediately after this project has been evaluated. The project is passed to the closure as a parameter. Such a listener gets notified when the build file belonging to this project has been executed. A parent project may for example add such a listener to its child project. Such a listener can further configure those child projects based on the state of the child projects after their build files have been run.

#### void afterEvaluate([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html)> action)

Adds an action to execute immediately after this project is evaluated.

#### void allprojects([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html)> action)

Configures this project and each of its sub-projects.

This method executes the given [Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html) against this project and each of its sub-projects.

#### [AntBuilder](https://docs.gradle.org/4.1/javadoc/org/gradle/api/AntBuilder.html" \t "_top) ant([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [AntBuilder](https://docs.gradle.org/4.1/javadoc/org/gradle/api/AntBuilder.html)> configureAction)

Executes the given action against the AntBuilder for this project. You can use this in your build file to execute ant tasks. See example in javadoc for [Project.getAnt()](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:ant)

#### void apply([Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html" \t "_top) closure)

Applies zero or more plugins or scripts.

The given closure is used to configure an [ObjectConfigurationAction](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/ObjectConfigurationAction.html), which “builds” the plugin application.

This method differs from [PluginAware.apply(java.util.Map)](https://docs.gradle.org/4.1/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/7/docs/api/java/util/Map.html)<[String](http://download.oracle.com/javase/7/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](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/ObjectConfigurationAction.html). That is, each key in the map is expected to be the name of a method [ObjectConfigurationAction](https://docs.gradle.org/4.1/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)](https://docs.gradle.org/4.1/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](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [ObjectConfigurationAction](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/ObjectConfigurationAction.html)> action)

Applies zero or more plugins or scripts.

The given closure is used to configure an [ObjectConfigurationAction](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/ObjectConfigurationAction.html), which “builds” the plugin application.

This method differs from [PluginAware.apply(java.util.Map)](https://docs.gradle.org/4.1/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 artifacts([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [ArtifactHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.ArtifactHandler.html)> configureAction)

Configures the published artifacts for this project.

This method executes the given action against the [ArtifactHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.ArtifactHandler.html) for this project.

Example:

configurations {

//declaring new configuration that will be used to associate with artifacts

schema

}

task schemaJar(type: Jar) {

//some imaginary task that creates a jar artifact with the schema

}

//associating the task that produces the artifact with the configuration

artifacts {

//configuration name and the task:

schema schemaJar

}

#### void beforeEvaluate([Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html" \t "_top) closure)

Adds a closure to be called immediately before this project is evaluated. The project is passed to the closure as a parameter.

#### void beforeEvaluate([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html)> action)

Adds an action to execute immediately before this project is evaluated.

#### [Iterable](http://download.oracle.com/javase/7/docs/api/java/lang/Iterable.html)<?> configure([Iterable](http://download.oracle.com/javase/7/docs/api/java/lang/Iterable.html)<?> objects, [Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html) configureClosure)

Configures a collection of objects via a closure. This is equivalent to calling [Project.configure(java.lang.Object, groovy.lang.Closure)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:configure(java.lang.Object, groovy.lang.Closure)) for each of the given objects.

#### [Iterable](http://download.oracle.com/javase/7/docs/api/java/lang/Iterable.html)<T> configure([Iterable](http://download.oracle.com/javase/7/docs/api/java/lang/Iterable.html)<T> objects, [Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super T> configureAction)

Configures a collection of objects via an action.

#### [Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top) configure([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html) object, [Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html) configureClosure)

Configures an object via a closure, with the closure's delegate set to the supplied object. This way you don't have to specify the context of a configuration statement multiple times.

Instead of:

MyType myType = new MyType()

myType.doThis()

myType.doThat()

you can do:

MyType myType = configure(new MyType()) {

doThis()

doThat()

}

The object being configured is also passed to the closure as a parameter, so you can access it explicitly if required:

configure(someObj) { obj -> obj.doThis() }

#### [NamedDomainObjectContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.NamedDomainObjectContainer.html)<T> container([Class](http://download.oracle.com/javase/7/docs/api/java/lang/Class.html)<T> type)

Creates a container for managing named objects of the specified type. The specified type must have a public constructor which takes the name as a String parameter.

All objects MUST expose their name as a bean property named "name". The name must be constant for the life of the object.

#### [NamedDomainObjectContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.NamedDomainObjectContainer.html)<T> container([Class](http://download.oracle.com/javase/7/docs/api/java/lang/Class.html)<T> type, [Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html)factoryClosure)

Creates a container for managing named objects of the specified type. The given closure is used to create object instances. The name of the instance to be created is passed as a parameter to the closure.

All objects MUST expose their name as a bean property named "name". The name must be constant for the life of the object.

#### [NamedDomainObjectContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.NamedDomainObjectContainer.html" \t "_top)<T> container([Class](http://download.oracle.com/javase/7/docs/api/java/lang/Class.html)<T> type, [NamedDomainObjectFactory](https://docs.gradle.org/4.1/javadoc/org/gradle/api/NamedDomainObjectFactory.html)<T> factory)

Creates a container for managing named objects of the specified type. The given factory is used to create object instances.

All objects MUST expose their name as a bean property named "name". The name must be constant for the life of the object.

#### [WorkResult](https://docs.gradle.org/4.1/javadoc/org/gradle/api/tasks/WorkResult.html" \t "_top) copy([Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html" \t "_top) closure)

Copies the specified files. The given closure is used to configure a [CopySpec](https://docs.gradle.org/4.1/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'

}

}

#### [WorkResult](https://docs.gradle.org/4.1/javadoc/org/gradle/api/tasks/WorkResult.html" \t "_top) copy([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html)> action)

Copies the specified files. The given action is used to configure a [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html), which is then used to copy the files.

#### [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html" \t "_top) copySpec()

Creates a [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html) which can later be used to copy files or create an archive.

#### [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html" \t "_top) copySpec([Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html" \t "_top) closure)

Creates a [CopySpec](https://docs.gradle.org/4.1/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](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html)before it is returned by this method.

def baseSpec = copySpec {

from "source"

include "\*\*/\*.java"

}

task copy(type: Copy) {

into "target"

with baseSpec

}

#### [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html" \t "_top) copySpec([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html)> action)

Creates a [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html) which can later be used to copy files or create an archive. The given action is used to configure the [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html)before it is returned by this method.

#### boolean delete([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top)... paths)

Deletes files and directories.

This will not follow symlinks. If you need to follow symlinks too use [Project.delete(org.gradle.api.Action)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:delete(org.gradle.api.Action)).

#### [WorkResult](https://docs.gradle.org/4.1/javadoc/org/gradle/api/tasks/WorkResult.html" \t "_top) delete([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [DeleteSpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/DeleteSpec.html)> action)

Deletes the specified files. The given action is used to configure a [DeleteSpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/DeleteSpec.html), which is then used to delete the files.

Example:

project.delete {

delete 'somefile'

followSymlinks = true

}

#### [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html" \t "_top) evaluationDependsOn([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) path)

Declares that this project has an evaluation dependency on the project with the given path.

#### [ExecResult](https://docs.gradle.org/4.1/javadoc/org/gradle/process/ExecResult.html" \t "_top) exec([Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html" \t "_top) closure)

Executes an external command. The closure configures a [ExecSpec](https://docs.gradle.org/4.1/javadoc/org/gradle/process/ExecSpec.html).

#### [ExecResult](https://docs.gradle.org/4.1/javadoc/org/gradle/process/ExecResult.html" \t "_top) exec([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [ExecSpec](https://docs.gradle.org/4.1/javadoc/org/gradle/process/ExecSpec.html)> action)

Executes an external command.

The given action configures a [ExecSpec](https://docs.gradle.org/4.1/javadoc/org/gradle/process/ExecSpec.html), which is used to launch the process. This method blocks until the process terminates, with its result being returned.

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html) file([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top) path)

Resolves a file path relative to the project directory of this project. This method converts the supplied path based on its type:

* A [CharSequence](http://download.oracle.com/javase/7/docs/api/java/lang/CharSequence.html), including [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html) or [GString](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/GString.html). Interpreted relative to the project directory. A string that starts with file: is treated as a file URL.
* A [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html). If the file is an absolute file, it is returned as is. Otherwise, the file's path is interpreted relative to the project directory.
* A [Path](http://download.oracle.com/javase/7/docs/api/java/nio/file/Path.html). The path must be associated with the default provider and is treated the same way as an instance of File.
* A [URI](http://download.oracle.com/javase/7/docs/api/java/net/URI.html) or [URL](http://download.oracle.com/javase/7/docs/api/java/net/URL.html). The URL's path is interpreted as the file path. Only file: URLs are supported.
* A [Directory](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/Directory.html) or [RegularFile](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/RegularFile.html).
* A [Provider](https://docs.gradle.org/4.1/javadoc/org/gradle/api/provider/Provider.html) of any supported type. The provider's value is resolved recursively.
* A [Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html) that returns any supported type. The closure's return value is resolved recursively.
* A [Callable](http://download.oracle.com/javase/7/docs/api/java/util/concurrent/Callable.html) that returns any supported type. The callable's return value is resolved recursively.

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html" \t "_top) file([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html) path, [PathValidation](https://docs.gradle.org/4.1/javadoc/org/gradle/api/PathValidation.html) validation)

Resolves a file path relative to the project directory of this project and validates it using the given scheme. See [PathValidation](https://docs.gradle.org/4.1/javadoc/org/gradle/api/PathValidation.html) for the list of possible validations.

#### [ConfigurableFileTree](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/ConfigurableFileTree.html) fileTree([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top) baseDir)

Creates a new ConfigurableFileTree using the given base directory. The given baseDir path is evaluated as per [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project: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.

def myTree = fileTree("src")

myTree.include "\*\*/\*.java"

myTree.builtBy "someTask"

task copy(type: Copy) {

from myTree

}

#### [ConfigurableFileTree](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/ConfigurableFileTree.html) fileTree([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html) baseDir, [Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html) configureClosure)

Creates a new ConfigurableFileTree using the given base directory. The given baseDir path is evaluated as per [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project: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:

def myTree = fileTree('src') {

exclude '\*\*/.data/\*\*'

builtBy 'someTask'

}

task copy(type: Copy) {

from myTree

}

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](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/ConfigurableFileTree.html" \t "_top) fileTree([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html) baseDir, [Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [ConfigurableFileTree](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/ConfigurableFileTree.html)> configureAction)

Creates a new ConfigurableFileTree using the given base directory. The given baseDir path is evaluated as per [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object)). The action will be used to configure the new file tree. Example:

def myTree = fileTree('src') {

exclude '\*\*/.data/\*\*'

builtBy 'someTask'

}

task copy(type: Copy) {

from myTree

}

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](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/ConfigurableFileTree.html" \t "_top) fileTree([Map](http://download.oracle.com/javase/7/docs/api/java/util/Map.html)<[String](http://download.oracle.com/javase/7/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:

def myTree = fileTree(dir:'src', excludes:['\*\*/ignore/\*\*', '\*\*/.data/\*\*'])

task copy(type: Copy) {

from myTree

}

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](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/ConfigurableFileCollection.html) files([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html) paths, [Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html) configureClosure)

Creates a new ConfigurableFileCollection using the given paths. The paths are evaluated as per [Project.files(java.lang.Object[])](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:files(java.lang.Object[])). The file collection is configured using the given closure. The file collection is passed to the closure as its delegate. Example:

files "$buildDir/classes" {

builtBy 'compile'

}

The returned file collection is lazy, so that the paths are evaluated only when the contents of the file collection are queried. The file collection is also live, so that it evaluates the above each time the contents of the collection is queried.

#### [ConfigurableFileCollection](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/ConfigurableFileCollection.html) files([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html) paths, [Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [ConfigurableFileCollection](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/ConfigurableFileCollection.html)> configureAction)

Creates a new ConfigurableFileCollection using the given paths. The paths are evaluated as per [Project.files(java.lang.Object[])](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:files(java.lang.Object[])). The file collection is configured using the given action. Example:

files "$buildDir/classes" {

builtBy 'compile'

}

The returned file collection is lazy, so that the paths are evaluated only when the contents of the file collection are queried. The file collection is also live, so that it evaluates the above each time the contents of the collection is queried.

#### [ConfigurableFileCollection](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/ConfigurableFileCollection.html" \t "_top) files([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top)... paths)

Returns a [ConfigurableFileCollection](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/ConfigurableFileCollection.html) containing the given files. You can pass any of the following types to this method:

* A [CharSequence](http://download.oracle.com/javase/7/docs/api/java/lang/CharSequence.html), including [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html) or [GString](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/GString.html). Interpreted relative to the project directory, as per [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object)). A string that starts with file: is treated as a file URL.
* A [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html). Interpreted relative to the project directory, as per [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object)).
* A [Path](http://download.oracle.com/javase/7/docs/api/java/nio/file/Path.html) as defined by [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object)).
* A [URI](http://download.oracle.com/javase/7/docs/api/java/net/URI.html) or [URL](http://download.oracle.com/javase/7/docs/api/java/net/URL.html). The URL's path is interpreted as a file path. Only file: URLs are supported.
* A [Directory](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/Directory.html) or [RegularFile](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/RegularFile.html).
* A [Collection](http://download.oracle.com/javase/7/docs/api/java/util/Collection.html), [Iterable](http://download.oracle.com/javase/7/docs/api/java/lang/Iterable.html), or an array that contains objects of any supported type. The elements of the collection are recursively converted to files.
* A [FileCollection](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/FileCollection.html). The contents of the collection are included in the returned collection.
* A [Provider](https://docs.gradle.org/4.1/javadoc/org/gradle/api/provider/Provider.html) of any supported type. The provider's value is recursively converted to files.
* A [Callable](http://download.oracle.com/javase/7/docs/api/java/util/concurrent/Callable.html) that returns any supported type. The return value of the call() method is recursively converted to files. A null return value is treated as an empty collection.
* A [Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html) that returns any of the types listed here. The return value of the closure is recursively converted to files. A null return value is treated as an empty collection.
* A [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html). Converted to the task's output files.
* A [TaskOutputs](https://docs.gradle.org/4.1/javadoc/org/gradle/api/tasks/TaskOutputs.html). Converted to the output files the related task.
* Anything else is treated as a failure.

The returned file collection is lazy, so that the paths are evaluated only when the contents of the file collection are queried. The file collection is also live, so that it evaluates the above each time the contents of the collection is queried.

The returned file collection maintains the iteration order of the supplied paths.

#### [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html" \t "_top) findProject([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) path)

Locates a project by path. If the path is relative, it is interpreted relative to this project.

#### [Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top) findProperty([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) propertyName)

*Note: This method is*[*incubating*](https://docs.gradle.org/4.1/userguide/feature_lifecycle.html)*and may change in a future version of Gradle.*

Returns the value of the given property or null if not found. This method locates a property as follows:

1. If this project object has a property with the given name, return the value of the property.
2. If this project has an extension with the given name, return the extension.
3. If this project's convention object has a property with the given name, return the value of the property.
4. If this project has an extra property with the given name, return the value of the property.
5. If this project has a task with the given name, return the task.
6. Search up through this project's ancestor projects for a convention property or extra property with the given name.
7. If not found, null value is returned.

#### [Map](http://download.oracle.com/javase/7/docs/api/java/util/Map.html" \t "_top)<[Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html), [Set](http://download.oracle.com/javase/7/docs/api/java/util/Set.html)<[Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html)>> getAllTasks(boolean recursive)

Returns a map of the tasks contained in this project, and optionally its subprojects.

#### [Set](http://download.oracle.com/javase/7/docs/api/java/util/Set.html" \t "_top)<[Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html)> getTasksByName([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html) name, boolean recursive)

Returns the set of tasks with the given name contained in this project, and optionally its subprojects.

#### boolean hasProperty([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) propertyName)

Determines if this project has the given property. See [here](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project.properties) for details of the properties which are available for a project.

#### [ExecResult](https://docs.gradle.org/4.1/javadoc/org/gradle/process/ExecResult.html" \t "_top) javaexec([Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html" \t "_top) closure)

Executes a Java main class. The closure configures a [JavaExecSpec](https://docs.gradle.org/4.1/javadoc/org/gradle/process/JavaExecSpec.html).

#### [ExecResult](https://docs.gradle.org/4.1/javadoc/org/gradle/process/ExecResult.html" \t "_top) javaexec([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [JavaExecSpec](https://docs.gradle.org/4.1/javadoc/org/gradle/process/JavaExecSpec.html)> action)

Executes an external Java process.

The given action configures a [JavaExecSpec](https://docs.gradle.org/4.1/javadoc/org/gradle/process/JavaExecSpec.html), which is used to launch the process. This method blocks until the process terminates, with its result being returned.

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html" \t "_top) mkdir([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top) path)

Creates a directory and returns a file pointing to it.

#### void normalization([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [InputNormalizationHandler](https://docs.gradle.org/4.1/dsl/org.gradle.normalization.InputNormalizationHandler.html)> configuration)

*Note: This method is*[*incubating*](https://docs.gradle.org/4.1/userguide/feature_lifecycle.html)*and may change in a future version of Gradle.*

Configures input normalization.

#### [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html) project([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) path)

Locates a project by path. If the path is relative, it is interpreted relative to this project.

#### [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html) project([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html) path, [Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html) configureClosure)

Locates a project by path and configures it using the given closure. If the path is relative, it is interpreted relative to this project. The target project is passed to the closure as the closure's delegate.

#### [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html" \t "_top) project([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html) path, [Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html)> configureAction)

Locates a project by path and configures it using the given action. If the path is relative, it is interpreted relative to this project.

#### [PropertyState](https://docs.gradle.org/4.1/javadoc/org/gradle/api/provider/PropertyState.html)<T> property([Class](http://download.oracle.com/javase/7/docs/api/java/lang/Class.html)<T> clazz)

*Note: This method is*[*incubating*](https://docs.gradle.org/4.1/userguide/feature_lifecycle.html)*and may change in a future version of Gradle.*

Creates a PropertyState implementation based on the provided class.

#### [Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top) property([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) propertyName)

Returns the value of the given property. This method locates a property as follows:

1. If this project object has a property with the given name, return the value of the property.
2. If this project has an extension with the given name, return the extension.
3. If this project's convention object has a property with the given name, return the value of the property.
4. If this project has an extra property with the given name, return the value of the property.
5. If this project has a task with the given name, return the task.
6. Search up through this project's ancestor projects for a convention property or extra property with the given name.
7. If not found, a [MissingPropertyException](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/MissingPropertyException.html) is thrown.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) relativePath([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top) path)

Returns the relative path from the project directory to the given path. The given path object is (logically) resolved as described for [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object)), from which a relative path is calculated.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) relativeProjectPath([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) path)

Converts a name to a project path relative to this project.

#### void setProperty([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html) name, [Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html) value)

Sets a property of this project. This method searches for a property with the given name in the following locations, and sets the property on the first location where it finds the property.

1. The project object itself. For example, the rootDir project property.
2. The project's [Convention](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/Convention.html) object. For example, the srcRootName java plugin property.
3. The project's extra properties.

If the property is not found, a [MissingPropertyException](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/MissingPropertyException.html) is thrown.

#### void subprojects([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html)> action)

Configures the sub-projects of this project

This method executes the given [Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html) against the sub-projects of this project.

#### [WorkResult](https://docs.gradle.org/4.1/javadoc/org/gradle/api/tasks/WorkResult.html" \t "_top) sync([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html)> action)

Synchronizes the contents of a destination directory with some source directories and files. The given action is used to configure a [CopySpec](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/CopySpec.html), which is then used to synchronize the files.

This method is like the [Project.copy(org.gradle.api.Action)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:copy(org.gradle.api.Action)) task, except the destination directory will only contain the files copied. All files that exist in the destination directory will be deleted before copying files, unless a preserve option is specified.

Example:

project.sync {

from 'my/shared/dependencyDir'

into 'build/deps/compile'

}

Note that you can preserve output that already exists in the destination directory:

project.sync {

from 'source'

into 'dest'

preserve {

include 'extraDir/\*\*'

include 'dir1/\*\*'

exclude 'dir1/extra.txt'

}

}

#### [FileTree](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/FileTree.html" \t "_top) tarTree([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top) tarPath)

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

* an instance of [Resource](https://docs.gradle.org/4.1/javadoc/org/gradle/api/resources/Resource.html)
* any other object is evaluated as per [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project: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 [Project.copy(groovy.lang.Closure)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project: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'

}

#### [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html) task([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) name)

Creates a [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html) with the given name and adds it to this project. Calling this method is equivalent to calling [Project.task(java.util.Map, java.lang.String)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:task(java.util.Map, java.lang.String)) with an empty options map.

After the task is added to the project, it is made available as a property of the project, so that you can reference the task by name in your build file. See [here](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project.properties) for more details

If a task with the given name already exists in this project, an exception is thrown.

#### [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html" \t "_top) task([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html) name, [Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html) configureClosure)

Creates a [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html) with the given name and adds it to this project. Before the task is returned, the given closure is executed to configure the task.

After the task is added to the project, it is made available as a property of the project, so that you can reference the task by name in your build file. See [here](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project.properties) for more details

#### [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html) task([Map](http://download.oracle.com/javase/7/docs/api/java/util/Map.html)<[String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html), ?> args, [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html) name)

Creates a [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html) with the given name and adds it to this project. A map of creation options can be passed to this method to control how the task is created. The following options are available:

| **Option** | **Description** | **Default Value** |
| --- | --- | --- |
| type | The class of the task to create. | [DefaultTask](https://docs.gradle.org/4.1/dsl/org.gradle.api.DefaultTask.html) |
| overwrite | Replace an existing task? | false |
| dependsOn | A task name or set of task names which this task depends on | [] |
| action | A closure or [Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html) to add to the task. | null |
| description | A description of the task. | null |
| group | A task group which this task belongs to. | null |

After the task is added to the project, it is made available as a property of the project, so that you can reference the task by name in your build file. See [here](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project.properties) for more details

If a task with the given name already exists in this project and the override option is not set to true, an exception is thrown.

#### [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html" \t "_top) task([Map](http://download.oracle.com/javase/7/docs/api/java/util/Map.html)<[String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html), ?> args, [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html) name, [Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html) configureClosure)

Creates a [Task](https://docs.gradle.org/4.1/dsl/org.gradle.api.Task.html) with the given name and adds it to this project. Before the task is returned, the given closure is executed to configure the task. A map of creation options can be passed to this method to control how the task is created. See [Project.task(java.util.Map, java.lang.String)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:task(java.util.Map, java.lang.String)) for the available options.

After the task is added to the project, it is made available as a property of the project, so that you can reference the task by name in your build file. See [here](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project.properties) for more details

If a task with the given name already exists in this project and the override option is not set to true, an exception is thrown.

#### [URI](http://download.oracle.com/javase/7/docs/api/java/net/URI.html" \t "_top) uri([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top) path)

Resolves a file path to a URI, relative to the project directory of this project. Evaluates the provided path object as described for [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object)), with the exception that any URI scheme is supported, not just 'file:' URIs.

#### [FileTree](https://docs.gradle.org/4.1/javadoc/org/gradle/api/file/FileTree.html" \t "_top) zipTree([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top) zipPath)

Creates a new FileTree which contains the contents of the given ZIP file. The given zipPath path is evaluated as per [Project.file(java.lang.Object)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:file(java.lang.Object)). You can combine this method with the [Project.copy(groovy.lang.Closure)](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project: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.

#### void appDirName([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) appDirName)

Allows changing the application directory. Default is "src/main/application".

#### [EarPluginConvention](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ear.EarPluginConvention.html) deploymentDescriptor([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [DeploymentDescriptor](https://docs.gradle.org/4.1/javadoc/org/gradle/plugins/ear/descriptor/DeploymentDescriptor.html)>configureAction)

Configures the deployment descriptor for this EAR archive.

The given action is executed to configure the deployment descriptor.

#### void libDirName([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) libDirName)

Allows changing the library directory in the EAR file. Default is "lib".

#### [Manifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/java/archives/Manifest.html" \t "_top) manifest()

Creates a new instance of a [Manifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/java/archives/Manifest.html).

#### [Manifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/java/archives/Manifest.html" \t "_top) manifest([Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html" \t "_top) closure)

Creates and configures a new instance of a [Manifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/java/archives/Manifest.html). The given closure configures the new manifest instance before it is returned.

#### [Manifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/java/archives/Manifest.html" \t "_top) manifest([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [Manifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/java/archives/Manifest.html)> action)

Creates and configures a new instance of a [Manifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/java/archives/Manifest.html).

#### [MavenPom](https://docs.gradle.org/4.1/javadoc/org/gradle/api/artifacts/maven/MavenPom.html" \t "_top) pom()

Creates a new [MavenPom](https://docs.gradle.org/4.1/javadoc/org/gradle/api/artifacts/maven/MavenPom.html).

#### [MavenPom](https://docs.gradle.org/4.1/javadoc/org/gradle/api/artifacts/maven/MavenPom.html" \t "_top) pom([Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html" \t "_top) configureClosure)

Creates and configures a new [MavenPom](https://docs.gradle.org/4.1/javadoc/org/gradle/api/artifacts/maven/MavenPom.html). The given closure is executed to configure the new POM instance.

#### [OsgiManifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/osgi/OsgiManifest.html" \t "_top) osgiManifest()

Creates a new instance of [OsgiManifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/osgi/OsgiManifest.html). The returned object is preconfigured with:

version: project.version

name: project.archivesBaseName

symbolicName: project.group + "." + project.archivesBaseName (see below for exceptions to this rule)

The symbolic name is usually the group + "." + archivesBaseName, with the following exceptions

* if group has only one section (no dots) and archivesBaseName is not null then the first package name with classes is returned. eg. commons-logging:commons-logging -> org.apache.commons.logging
* if archivesBaseName is equal to last section of group then group is returned. eg. org.gradle:gradle -> org.gradle
* if archivesBaseName starts with last section of group that portion is removed. eg. org.gradle:gradle-core -> org.gradle.core

#### [OsgiManifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/osgi/OsgiManifest.html" \t "_top) osgiManifest([Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html" \t "_top) closure)

Creates and configures a new instance of an [OsgiManifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/osgi/OsgiManifest.html) . The closure configures the new manifest instance before it is returned.

#### [OsgiManifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/osgi/OsgiManifest.html" \t "_top) osgiManifest([Action](https://docs.gradle.org/4.1/javadoc/org/gradle/api/Action.html)<? super [OsgiManifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/osgi/OsgiManifest.html)> action)

Creates and configures a new instance of an [OsgiManifest](https://docs.gradle.org/4.1/javadoc/org/gradle/api/plugins/osgi/OsgiManifest.html). The action configures the new manifest instance before it is returned.

### Script block details

(脚本块细节)

#### allprojects { }

Configures this project and each of its sub-projects.

This method executes the given closure against this project and its sub-projects. The target [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html) is passed to the closure as the closure's delegate（目标[Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html) 被传递到这个闭包作为这个闭包的代理）.

**Delegates to:**

Each [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html) in [allprojects](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:allprojects)

#### ant { }

Executes the given closure against the AntBuilder for this project. You can use this in your build file to execute ant tasks. The AntBuilder is passed to the closure as the closure's delegate. See example in javadoc for [Project.getAnt()](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:ant)

**Delegates to:**

[AntBuilder](https://docs.gradle.org/4.1/javadoc/org/gradle/api/AntBuilder.html) from [ant](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:ant)

#### artifacts { }

Configures the published artifacts for this project.

This method executes the given closure against the [ArtifactHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.ArtifactHandler.html) for this project. The [ArtifactHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.ArtifactHandler.html) is passed to the closure as the closure's delegate.

Example:

configurations {

//declaring new configuration that will be used to associate with artifacts

schema

}

task schemaJar(type: Jar) {

//some imaginary task that creates a jar artifact with the schema

}

//associating the task that produces the artifact with the configuration

artifacts {

//configuration name and the task:

schema schemaJar

}

**Delegates to:**

[ArtifactHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.ArtifactHandler.html) from [artifacts](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:artifacts)

#### buildscript { }

Configures the build script classpath for this project.

The given closure is executed against this project's [ScriptHandler](https://docs.gradle.org/4.1/javadoc/org/gradle/api/initialization/dsl/ScriptHandler.html). The [ScriptHandler](https://docs.gradle.org/4.1/javadoc/org/gradle/api/initialization/dsl/ScriptHandler.html) is passed to the closure as the closure's delegate.

**Delegates to:**

[ScriptHandler](https://docs.gradle.org/4.1/javadoc/org/gradle/api/initialization/dsl/ScriptHandler.html) from [buildscript](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:buildscript)

#### configurations { }

Configures the dependency configurations for this project.

This method executes the given closure against the [ConfigurationContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ConfigurationContainer.html) for this project. The [ConfigurationContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ConfigurationContainer.html) is passed to the closure as the closure's delegate.

##### Examples:

See docs for [ConfigurationContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ConfigurationContainer.html)

**Delegates to:**

[ConfigurationContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.ConfigurationContainer.html) from [configurations](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:configurations)

#### dependencies { }

Configures the dependencies for this project.

This method executes the given closure against the [DependencyHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html) for this project. The [DependencyHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html) is passed to the closure as the closure's delegate.

##### Examples:

See docs for [DependencyHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html)

**Delegates to:**

[DependencyHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.DependencyHandler.html) from [dependencies](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:dependencies)

#### repositories { }

Configures the repositories for this project.

This method executes the given closure against the [RepositoryHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.RepositoryHandler.html) for this project. The [RepositoryHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.RepositoryHandler.html) is passed to the closure as the closure's delegate.

**Delegates to:**

[RepositoryHandler](https://docs.gradle.org/4.1/dsl/org.gradle.api.artifacts.dsl.RepositoryHandler.html) from [repositories](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:repositories)

#### subprojects { }

Configures the sub-projects of this project.

This method executes the given closure against each of the sub-projects of this project. The target [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html) is passed to the closure as the closure's delegate.

**Delegates to:**

Each [Project](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html) in [subprojects](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:subprojects)

#### announce { }

Configures the [AnnouncePluginExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.announce.AnnouncePluginExtension.html) added by the announce plugin.

**Delegates to:**

[AnnouncePluginExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.announce.AnnouncePluginExtension.html) from [announce](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:announce)

#### checkstyle { }

Configures the [CheckstyleExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.CheckstyleExtension.html) added by the checkstyle plugin.

**Delegates to:**

[CheckstyleExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.CheckstyleExtension.html) from [checkstyle](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:checkstyle)

#### codenarc { }

Configures the [CodeNarcExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.CodeNarcExtension.html) added by the codenarc plugin.

**Delegates to:**

[CodeNarcExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.CodeNarcExtension.html) from [codenarc](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:codenarc)

#### distributions { }

Configures the [DistributionContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.distribution.DistributionContainer.html) added by the distribution plugin.

**Delegates to:**

[DistributionContainer](https://docs.gradle.org/4.1/dsl/org.gradle.api.distribution.DistributionContainer.html) from [distributions](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:distributions)

#### deploymentDescriptor { }

Configures the deployment descriptor for this EAR archive.

The given closure is executed to configure the deployment descriptor. The [DeploymentDescriptor](https://docs.gradle.org/4.1/javadoc/org/gradle/plugins/ear/descriptor/DeploymentDescriptor.html) is passed to the closure as its delegate.

**Delegates to:**

[DeploymentDescriptor](https://docs.gradle.org/4.1/javadoc/org/gradle/plugins/ear/descriptor/DeploymentDescriptor.html) from [deploymentDescriptor](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:deploymentDescriptor)

#### eclipse { }

Configures the [EclipseModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseModel.html) added by the eclipse plugin.

**Delegates to:**

[EclipseModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.eclipse.model.EclipseModel.html) from [eclipse](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:eclipse)

#### findbugs { }

Configures the [FindBugsExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.FindBugsExtension.html) added by the findbugs plugin.

**Delegates to:**

[FindBugsExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.FindBugsExtension.html) from [findbugs](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:findbugs)

#### idea { }

Configures the [IdeaModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaModel.html) added by the idea plugin.

**Delegates to:**

[IdeaModel](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.ide.idea.model.IdeaModel.html) from [idea](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:idea)

#### jacoco { }

Configures the [JacocoPluginExtension](https://docs.gradle.org/4.1/dsl/org.gradle.testing.jacoco.plugins.JacocoPluginExtension.html) added by the jacoco plugin.

**Delegates to:**

[JacocoPluginExtension](https://docs.gradle.org/4.1/dsl/org.gradle.testing.jacoco.plugins.JacocoPluginExtension.html) from [jacoco](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:jacoco)

#### reporting { }

Configures the [ReportingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.ReportingExtension.html) added by the java plugin.

**Delegates to:**

[ReportingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.reporting.ReportingExtension.html) from [reporting](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:reporting)

#### sourceSets { }

Configures the source sets of this project.

The given closure is executed to configure the [SourceSetContainer](https://docs.gradle.org/4.1/javadoc/org/gradle/api/tasks/SourceSetContainer.html). The [SourceSetContainer](https://docs.gradle.org/4.1/javadoc/org/gradle/api/tasks/SourceSetContainer.html) is passed to the closure as its delegate.

See the example below how [SourceSet](https://docs.gradle.org/4.1/dsl/org.gradle.api.tasks.SourceSet.html) 'main' is accessed and how the [SourceDirectorySet](https://docs.gradle.org/4.1/dsl/org.gradle.api.file.SourceDirectorySet.html) 'java' is configured to exclude some package from compilation.

apply plugin: 'java'

sourceSets {

main {

java {

exclude 'some/unwanted/package/\*\*'

}

}

}

**Delegates to:**

[SourceSetContainer](https://docs.gradle.org/4.1/javadoc/org/gradle/api/tasks/SourceSetContainer.html) from [sourceSets](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:sourceSets)

#### jdepend { }

Configures the [JDependExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.JDependExtension.html) added by the jdepend plugin.

**Delegates to:**

[JDependExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.JDependExtension.html) from [jdepend](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:jdepend)

#### pmd { }

Configures the [PmdExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.PmdExtension.html) added by the pmd plugin.

**Delegates to:**

[PmdExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.plugins.quality.PmdExtension.html) from [pmd](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:pmd)

#### publishing { }

Configures the [PublishingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.PublishingExtension.html) added by the publishing plugin.

**Delegates to:**

[PublishingExtension](https://docs.gradle.org/4.1/dsl/org.gradle.api.publish.PublishingExtension.html) from [publishing](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:publishing)

#### signing { }

Configures the [SigningExtension](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.signing.SigningExtension.html) added by the signing plugin.

**Delegates to:**

[SigningExtension](https://docs.gradle.org/4.1/dsl/org.gradle.plugins.signing.SigningExtension.html) from [signing](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:signing)

#### visualStudio { }

Configures the [VisualStudioExtension](https://docs.gradle.org/4.1/dsl/org.gradle.ide.visualstudio.VisualStudioExtension.html) added by the visual-studio plugin.

配置visual studio扩展，扩展通过visual-studo插件添加

**Delegates to:**

[VisualStudioExtension](https://docs.gradle.org/4.1/dsl/org.gradle.ide.visualstudio.VisualStudioExtension.html) from [visualStudio](https://docs.gradle.org/4.1/dsl/org.gradle.api.Project.html#org.gradle.api.Project:visualStudio)

# Task

|  |  |
| --- | --- |
| **API Documentation:** | [Task](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Task.html) |

A Task represents a single atomic piece of work(构建的简单的原子操作任务) for a build, such as compiling classes or generating javadoc.

Each task belongs to a [Project](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Project.html)（每一个task属于一个project）. You can use the various methods on [TaskContainer](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\tasks\TaskContainer.html) to create and lookup task instances. For example, [TaskContainer.create(java.lang.String)](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\tasks\TaskContainer.html#create(java.lang.String)) creates an empty task with the given name. You can also use the task keyword in your build file:

task myTask

task myTask { configure closure }

task myTask(type: SomeType)

task myTask(type: SomeType) { configure closure }

Each task has a name, which can be used to refer to the task within its owning project, and a fully qualified path（有效的路径）, which is unique across all tasks in all projects. The path is the concatenation（相关） of the owning project's path and the task's name. Path elements are separated using the : character（使用:分隔路径各个元素）.

### Task Actions

任务行为

A Task is made up of a sequence of [Action](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Action.html) objects（一个任务由一系列Action对象组成）. When the task is executed, each of the actions is executed in turn（当任务执行的时候每个行为也按照顺序执行）, by calling [Action.execute(T)](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Action.html#execute(T)). You can add actions（添加行为） to a task by calling [Task.doFirst(org.gradle.api.Action)](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:doFirst(org.gradle.api.Action)) or [Task.doLast(org.gradle.api.Action)](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:doLast(org.gradle.api.Action)).

Groovy closures（闭包） can also be used to provide a task action. When the action is executed, the closure is called with the task as parameter. You can add action closures to a task by calling [Task.doFirst(groovy.lang.Closure)](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:doFirst(groovy.lang.Closure))or [Task.doLast(groovy.lang.Closure)](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:doLast(groovy.lang.Closure)).

There are 2 special exceptions which a task action can throw to abort execution and continue without failing the build. A task action can abort execution of the action and continue to the next action of the task by throwing a [StopActionException](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\tasks\StopActionException.html)（该异常停止当前行为的执行但是继续执行下一个行为）. A task action can abort execution of the task and continue to the next task by throwing a [StopExecutionException](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\tasks\StopExecutionException.html)（停止当前任务的执行，继续执行下一个任务）. Using these exceptions allows you to have precondition actions which skip execution of the task, or part of the task, if not true.

### Task Dependencies and Task Ordering

A task may have dependencies on other tasks or might be scheduled to always run after another task. Gradle ensures that all task dependencies and ordering rules are honored when executing tasks, so that the task is executed after all of its dependencies and any "must run after" tasks have been executed.（Gradle保证了任务的执行在他所依赖的任务执行之后，或者设置某个任务在其他任务之后执行）

Dependencies to a task are controlled using [Task.dependsOn(java.lang.Object[])](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:dependsOn(java.lang.Object[])) or [Task.setDependsOn(java.lang.Iterable)](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Task.html#setDependsOn(java.lang.Iterable)), and [Task.mustRunAfter(java.lang.Object[])](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:mustRunAfter(java.lang.Object[])), [Task.setMustRunAfter(java.lang.Iterable)](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Task.html#setMustRunAfter(java.lang.Iterable)), [Task.shouldRunAfter(java.lang.Object[])](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Task.html#shouldRunAfter(java.lang.Object[])) and [Task.setShouldRunAfter(java.lang.Iterable)](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Task.html#setShouldRunAfter(java.lang.Iterable)) are used to specify ordering between tasks. You can use objects of any of the following types to specify dependencies and ordering:

* A String, CharSequence or groovy.lang.GString task path or name. A relative path is interpreted relative to the task's [Project](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Project.html)（相对有task所在的任务来解析任务名）. This allows you to refer to tasks in other projects.
* A [Task](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html).
* A closure. The closure may take a Task as parameter. It may return any of the types listed here. Its return value is recursively converted to tasks. A null return value is treated as an empty collection.
* A [TaskDependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\tasks\TaskDependency.html) object.
* A [Buildable](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Buildable.html) object.
* A Iterable, Collection, Map or array. May contain any of the types listed here. The elements of the iterable/collection/map/array are recursively converted to tasks.
* A Callable. The call() method may return any of the types listed here. Its return value is recursively converted to tasks. A null return value is treated as an empty collection.

### Using a Task in a Build File

#### Dynamic Properties

A Task has 4 'scopes' for properties. You can access these properties by name from the build file or by calling the [Task.property(java.lang.String)](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:property(java.lang.String)) method. You can change the value of these properties by calling the [Task.setProperty(java.lang.String, java.lang.Object)](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:setProperty(java.lang.String, java.lang.Object)) method.

* The Task object itself. This includes any property getters and setters declared by the Task implementation class. The properties of this scope are readable or writable based on the presence of the corresponding getter and setter methods.
* The extensions added to the task by plugins. Each extension is available as a read-only property with the same name as the extension.
* The convention properties added to the task by plugins. A plugin can add properties and methods to a task through the task's [Convention](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\plugins\Convention.html) object. The properties of this scope may be readable or writable, depending on the convention objects.
* The extra properties of the task. Each task object maintains a map of additional properties. These are arbitrary name -> value pairs which you can use to dynamically add properties to a task object. Once defined, the properties of this scope are readable and writable.

#### Dynamic Methods

A [Plugin](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Plugin.html) may add methods to a Task using its [Convention](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\plugins\Convention.html) object.

#### Parallel Execution

By default, tasks are not executed in parallel unless a task is waiting on asynchronous work and another task (which is not dependent) is ready to execute. Parallel execution can be enabled by the --parallel flag when the build is initiated. In parallel mode, the tasks of different projects (i.e. in a multi project build) are able to be executed in parallel.

### Properties

| **Property** | **Description** |
| --- | --- |
| [actions](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:actions) | The sequence of [Action](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Action.html) objects which will be executed by this task, in the order of execution. |
| [ant](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:ant) | The AntBuilder for this task. You can use this in your build file to execute ant tasks. |
| [convention](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:convention) | The [Convention](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\plugins\Convention.html) object for this task. A [Plugin](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Plugin.html) can use the convention object to contribute properties and methods to this task. |
| [dependsOn](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:dependsOn) | The dependencies of this task. |
| [description](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:description) | The description of this task. |
| [destroyables](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:destroyables) | *incubating*  The destroyables of this task. |
| [didWork](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:didWork) | Checks if the task actually did any work. Even if a Task executes, it may determine that it has nothing to do. For example, a compilation task may determine that source files have not changed since the last time a the task was run. |
| [enabled](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:enabled) | Returns if this task is enabled or not. |
| [extensions](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:extensions) | The container of extensions. |
| [finalizedBy](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:finalizedBy) | *incubating*  Returns tasks that finalize this task. |
| [group](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:group) | The task group which this task belongs to. The task group is used in reports and user interfaces to group related tasks together when presenting a list of tasks to the user. |
| [inputs](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:inputs) | The inputs of this task. |
| [logger](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:logger) | The logger for this task. You can use this in your build file to write log messages. |
| [logging](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:logging) | The [LoggingManager](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\logging\LoggingManager.html) which can be used to receive logging and to control the standard output/error capture for this task. 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. |
| [mustRunAfter](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:mustRunAfter) | *incubating*  Returns tasks that this task must run after. |
| [name](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:name) | The name of this task. The name uniquely identifies the task within its [Project](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Project.html). |
| [outputs](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:outputs) | The outputs of this task. |
| [path](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:path) | The path of the task, which is a fully qualified name for the task. The path of a task is the path of its [Project](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Project.html) plus the name of the task, separated by :. |
| [project](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:project) | The [Project](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Project.html) which this task belongs to. |
| [state](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:state) | The execution state of this task. This provides information about the execution of this task, such as whether it has executed, been skipped, has failed, etc. |
| [taskDependencies](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:taskDependencies) | Returns a [TaskDependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\tasks\TaskDependency.html) which contains all the tasks that this task depends on. |
| [temporaryDir](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:temporaryDir) | Returns a directory which this task can use to write temporary files to. Each task instance is provided with a separate temporary directory. There are no guarantees that the contents of this directory will be kept beyond the execution of the task. |

### Methods

| **Method** | **Description** |
| --- | --- |
| [deleteAllActions](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:deleteAllActions())() | Removes all the actions of this task. |
| [dependsOn](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:dependsOn(java.lang.Object[]))(paths) | Adds the given dependencies to this task. See [here](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task.dependencies) for a description of the types of objects which can be used as task dependencies. |
| [doFirst](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:doFirst(groovy.lang.Closure))(action) | Adds the given closure to the beginning of this task's action list. The closure is passed this task as a parameter when executed. |
| [doFirst](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:doFirst(org.gradle.api.Action))(action) | Adds the given [Action](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Action.html) to the beginning of this task's action list. |
| [doLast](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:doLast(groovy.lang.Closure))(action) | Adds the given closure to the end of this task's action list. The closure is passed this task as a parameter when executed. |
| [doLast](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:doLast(org.gradle.api.Action))(action) | Adds the given [Action](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Action.html) to the end of this task's action list. |
| [finalizedBy](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:finalizedBy(java.lang.Object[]))(paths) | *incubating*  Adds the given finalizer tasks for this task. |
| [hasProperty](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:hasProperty(java.lang.String))(propertyName) | Determines if this task has the given property. See [here](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task.properties) for details of the properties which are available for a task. |
| [leftShift](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:leftShift(groovy.lang.Closure))(action) | *deprecated*  Adds the given closure to the end of this task's action list. The closure is passed this task as a parameter when executed. You can call this method from your build script using the << left shift operator. |
| [mustRunAfter](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:mustRunAfter(java.lang.Object[]))(paths) | *incubating*  Specifies that this task must run after all of the supplied tasks. |
| [onlyIf](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:onlyIf(groovy.lang.Closure))(onlyIfClosure) | Execute the task only if the given closure returns true. The closure will be evaluated at task execution time, not during configuration. The closure will be passed a single parameter, this task. If the closure returns false, the task will be skipped. |
| [onlyIf](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:onlyIf(org.gradle.api.specs.Spec))(onlyIfSpec) | Execute the task only if the given spec is satisfied. The spec will be evaluated at task execution time, not during configuration. If the Spec is not satisfied, the task will be skipped. |
| [property](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:property(java.lang.String))(propertyName) | Returns the value of the given property of this task. This method locates a property as follows: |
| [setProperty](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task:setProperty(java.lang.String, java.lang.Object))(name, value) | Sets a property of this task. This method searches for a property with the given name in the following locations, and sets the property on the first location where it finds the property. |

### Script blocks

No script blocks

### Property details

#### [List](http://download.oracle.com/javase/7/docs/api/java/util/List.html" \t "_top)<[Action](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Action.html)<? super [Task](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html)>> actions

The sequence of [Action](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Action.html) objects which will be executed by this task, in the order of execution.

#### [AntBuilder](file:///E:\\gradle-4.1\\docs\\javadoc\\org\\gradle\\api\\AntBuilder.html" \t "_top) ant (read-only)

The AntBuilder for this task. You can use this in your build file to execute ant tasks.

#### [Convention](file:///E:\\gradle-4.1\\docs\\javadoc\\org\\gradle\\api\\plugins\\Convention.html" \t "_top) convention (read-only)

The [Convention](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\plugins\Convention.html) object for this task. A [Plugin](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Plugin.html) can use the convention object to contribute properties and methods to this task.

#### [Set](http://download.oracle.com/javase/7/docs/api/java/util/Set.html" \t "_top)<[Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html)> dependsOn

The dependencies of this task.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) description

The description of this task.

#### [TaskDestroyables](file:///E:\\gradle-4.1\\docs\\javadoc\\org\\gradle\\api\\tasks\\TaskDestroyables.html" \t "_top) destroyables (read-only)

*Note: This property is*[*incubating*](file:///E:\gradle-4.1\docs\userguide\feature_lifecycle.html)*and may change in a future version of Gradle.*

The destroyables of this task.

#### boolean didWork

Checks if the task actually did any work. Even if a Task executes, it may determine that it has nothing to do. For example, a compilation task may determine that source files have not changed since the last time a the task was run.

#### boolean enabled

Returns if this task is enabled or not.

#### [ExtensionContainer](file:///E:\\gradle-4.1\\docs\\javadoc\\org\\gradle\\api\\plugins\\ExtensionContainer.html" \t "_top) extensions (read-only)

The container of extensions.

#### [TaskDependency](file:///E:\\gradle-4.1\\docs\\javadoc\\org\\gradle\\api\\tasks\\TaskDependency.html" \t "_top) finalizedBy

*Note: This property is*[*incubating*](file:///E:\gradle-4.1\docs\userguide\feature_lifecycle.html)*and may change in a future version of Gradle.*

Returns tasks that finalize this task.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) group

The task group which this task belongs to. The task group is used in reports and user interfaces to group related tasks together when presenting a list of tasks to the user.

#### [TaskInputs](file:///E:\\gradle-4.1\\docs\\javadoc\\org\\gradle\\api\\tasks\\TaskInputs.html" \t "_top) inputs (read-only)

The inputs of this task.

#### [Logger](file:///E:\\gradle-4.1\\docs\\javadoc\\org\\gradle\\api\\logging\\Logger.html" \t "_top) logger (read-only)

The logger for this task. You can use this in your build file to write log messages.

#### [LoggingManager](file:///E:\\gradle-4.1\\docs\\javadoc\\org\\gradle\\api\\logging\\LoggingManager.html" \t "_top) logging (read-only)

The [LoggingManager](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\logging\LoggingManager.html) which can be used to receive logging and to control the standard output/error capture for this task. 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.

#### [TaskDependency](file:///E:\\gradle-4.1\\docs\\javadoc\\org\\gradle\\api\\tasks\\TaskDependency.html" \t "_top) mustRunAfter

*Note: This property is*[*incubating*](file:///E:\gradle-4.1\docs\userguide\feature_lifecycle.html)*and may change in a future version of Gradle.*

Returns tasks that this task must run after.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) name (read-only)

The name of this task. The name uniquely identifies the task within its [Project](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Project.html).

#### [TaskOutputs](file:///E:\\gradle-4.1\\docs\\javadoc\\org\\gradle\\api\\tasks\\TaskOutputs.html" \t "_top) outputs (read-only)

The outputs of this task.

#### [String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) path (read-only)

The path of the task, which is a fully qualified name for the task. The path of a task is the path of its [Project](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Project.html) plus the name of the task, separated by :.

#### [Project](file:///E:\\gradle-4.1\\docs\\dsl\\org.gradle.api.Project.html" \t "_top) project (read-only)

The [Project](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Project.html) which this task belongs to.

#### [TaskState](file:///E:\\gradle-4.1\\docs\\javadoc\\org\\gradle\\api\\tasks\\TaskState.html" \t "_top) state (read-only)

The execution state of this task. This provides information about the execution of this task, such as whether it has executed, been skipped, has failed, etc.

#### [TaskDependency](file:///E:\\gradle-4.1\\docs\\javadoc\\org\\gradle\\api\\tasks\\TaskDependency.html" \t "_top) taskDependencies (read-only)

Returns a [TaskDependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\tasks\TaskDependency.html) which contains all the tasks that this task depends on.

#### [File](http://download.oracle.com/javase/7/docs/api/java/io/File.html" \t "_top) temporaryDir (read-only)

Returns a directory which this task can use to write temporary files to. Each task instance is provided with a separate temporary directory. There are no guarantees that the contents of this directory will be kept beyond the execution of the task.

### Method details

#### [Task](file:///E:\\gradle-4.1\\docs\\dsl\\org.gradle.api.Task.html" \t "_top) deleteAllActions()

Removes all the actions of this task.

#### [Task](file:///E:\\gradle-4.1\\docs\\dsl\\org.gradle.api.Task.html" \t "_top) dependsOn([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top)... paths)

Adds the given dependencies to this task. See [here](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task.dependencies) for a description of the types of objects which can be used as task dependencies.

#### [Task](file:///E:\\gradle-4.1\\docs\\dsl\\org.gradle.api.Task.html" \t "_top) doFirst([Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html" \t "_top) action)

Adds the given closure to the beginning of this task's action list. The closure is passed this task as a parameter when executed.

#### [Task](file:///E:\\gradle-4.1\\docs\\dsl\\org.gradle.api.Task.html" \t "_top) doFirst([Action](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Action.html)<? super [Task](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html)> action)

Adds the given [Action](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Action.html) to the beginning of this task's action list.

#### [Task](file:///E:\\gradle-4.1\\docs\\dsl\\org.gradle.api.Task.html" \t "_top) doLast([Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html" \t "_top) action)

Adds the given closure to the end of this task's action list. The closure is passed this task as a parameter when executed.

#### [Task](file:///E:\\gradle-4.1\\docs\\dsl\\org.gradle.api.Task.html" \t "_top) doLast([Action](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Action.html)<? super [Task](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html)> action)

Adds the given [Action](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Action.html) to the end of this task's action list.

#### [Task](file:///E:\\gradle-4.1\\docs\\dsl\\org.gradle.api.Task.html" \t "_top) finalizedBy([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top)... paths)

*Note: This method is*[*incubating*](file:///E:\gradle-4.1\docs\userguide\feature_lifecycle.html)*and may change in a future version of Gradle.*

Adds the given finalizer tasks for this task.

task taskY {

finalizedBy "taskX"

}

See [here](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task.dependencies) for a description of the types of objects which can be used to specify a finalizer task.

#### boolean hasProperty([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) propertyName)

Determines if this task has the given property. See [here](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task.properties) for details of the properties which are available for a task.

#### [Task](file:///E:\\gradle-4.1\\docs\\dsl\\org.gradle.api.Task.html" \t "_top) leftShift([Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html" \t "_top) action)

*Note: This method is*[*deprecated*](file:///E:\gradle-4.1\docs\userguide\feature_lifecycle.html)*and will be removed in the next major version of Gradle.*

Adds the given closure to the end of this task's action list. The closure is passed this task as a parameter when executed. You can call this method from your build script using the << left shift operator.

#### [Task](file:///E:\\gradle-4.1\\docs\\dsl\\org.gradle.api.Task.html" \t "_top) mustRunAfter([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top)... paths)

*Note: This method is*[*incubating*](file:///E:\gradle-4.1\docs\userguide\feature_lifecycle.html)*and may change in a future version of Gradle.*

Specifies that this task must run after all of the supplied tasks.

task taskY {

mustRunAfter "taskX"

}

For each supplied task, this action adds a task 'ordering', and does not specify a 'dependency' between the tasks. As such, it is still possible to execute 'taskY' without first executing the 'taskX' in the example.

See [here](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html#org.gradle.api.Task.dependencies) for a description of the types of objects which can be used to specify an ordering relationship.

#### void onlyIf([Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html" \t "_top) onlyIfClosure)

Execute the task only if the given closure returns true. The closure will be evaluated at task execution time, not during configuration. The closure will be passed a single parameter, this task. If the closure returns false, the task will be skipped.

You may add multiple such predicates. The task is skipped if any of the predicates return false.

Typical usage:myTask.onlyIf{ dependsOnTaskDidWork() }

#### void onlyIf([Spec](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\specs\Spec.html)<? super [Task](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.Task.html)> onlyIfSpec)

Execute the task only if the given spec is satisfied. The spec will be evaluated at task execution time, not during configuration. If the Spec is not satisfied, the task will be skipped.

You may add multiple such predicates. The task is skipped if any of the predicates return false.

Typical usage (from Java):

myTask.onlyIf(new Spec<Task>() {

boolean isSatisfiedBy(Task task) {

return task.dependsOnTaskDidWork();

}

});

#### [Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top) property([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html" \t "_top) propertyName)

Returns the value of the given property of this task. This method locates a property as follows:

1. If this task object has a property with the given name, return the value of the property.
2. If this task has an extension with the given name, return the extension.
3. If this task's convention object has a property with the given name, return the value of the property.
4. If this task has an extra property with the given name, return the value of the property.
5. If not found, throw [MissingPropertyException](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/MissingPropertyException.html)

#### void setProperty([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html) name, [Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html) value)

Sets a property of this task. This method searches for a property with the given name in the following locations, and sets the property on the first location where it finds the property.

1. The task object itself. For example, the enabled project property.
2. The task's convention object.
3. The task's extra properties.

If the property is not found, a [MissingPropertyException](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/MissingPropertyException.html) is thrown.

# DependencyHandler

|  |  |
| --- | --- |
| **API Documentation:** | [DependencyHandler](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\dsl\DependencyHandler.html) |

A DependencyHandler is used to declare dependencies（依赖HandencyHandler被用来定义依赖项）. Dependencies are grouped into configurations （依赖被组织进configurations）(see [Configuration](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.Configuration.html)).

To declare a specific dependency for a configuration you can use the following syntax:

dependencies {

configurationName dependencyNotation1, dependencyNotation2, ...

}

Example shows a basic way of declaring dependencies.

apply plugin: 'java'

//so that we can use 'compile', 'testCompile' for dependencies

dependencies {

//for dependencies found in artifact repositories(包仓库) you can use

//the group:name:version notation（可以使用格式：group:name:version）

compile 'commons-lang:commons-lang:2.6'

testCompile 'org.mockito:mockito:1.9.0-rc1'

//map-style notation:map类型的notion

compile group: 'com.google.code.guice', name: 'guice', version: '1.0'

//declaring arbitrary files as dependencies 定义任意的文件作为jar包

compile files('hibernate.jar', 'libs/spring.jar')

//putting all jars from 'libs' onto compile classpath 从lib加载所有的jar到classpath

compile fileTree('libs')

}

### Advanced dependency configuration

高级依赖配置

To do some advanced configuration on a dependency when it is declared, you can additionally pass a configuration closure（传递一个配置闭包）:

dependencies {

configurationName(dependencyNotation){

configStatement1

configStatement2

}

}

Examples of advanced dependency declaration including（高级依赖定义包含的例子如下）:

* Forcing certain dependency version in case of the conflict（强制确定的依赖版本一面冲突）.
* Excluding certain dependencies by name, group or both（通过名字、group或者两者排除确定的依赖）. More details about per-dependency exclusions can be found in docs for [ModuleDependency.exclude(java.util.Map)](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\ModuleDependency.html#exclude(java.util.Map)).
* Avoiding transitive dependencies for certain dependency（避免对某个依赖项的过度依赖）.

apply plugin: 'java' //so that I can declare 'compile' dependencies

dependencies {

compile('org.hibernate:hibernate:3.1') {

//in case of versions conflict '3.1' version of hibernate wins:

force = true

//excluding a particular transitive dependency:

exclude module: 'cglib' //by artifact name

exclude group: 'org.jmock' //by group

exclude group: 'org.unwanted', module: 'iAmBuggy' //by both name and group

//disabling all transitive dependencies of this dependency

transitive = false

}

}

More examples of advanced configuration, useful when dependency module has multiple artifacts:

* Declaring dependency to a specific configuration of the module.
* Explicit specification of the artifact. See also [ModuleDependency.artifact(groovy.lang.Closure)](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\ModuleDependency.html#artifact(groovy.lang.Closure)).

apply plugin: 'java' //so that I can declare 'compile' testCompile dependencies

dependencies {

//configuring dependency to specific configuration of the module 指定依赖模块的特定配置

compile configuration: 'someConf', group: 'org.someOrg', name: 'someModule', version: '1.0'

//configuring dependency on 'someLib' module 配置依赖于某个模块

compile(group: 'org.myorg', name: 'someLib', version:'1.0') {

//explicitly adding the dependency artifact: 显示的添加依赖的jar包

artifact {

//useful when some artifact properties unconventional 当一些jar的属性是非conventional是非常有用

name = 'someArtifact' //artifact name different than module name artifact名字而不是模块名

extension = 'someExt'

type = 'someType'

classifier = 'someClassifier'

}

}

}

### Dependency notations

依赖注释

There are several supported dependency notations. These are described below. For each dependency declared this way, a [Dependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\Dependency.html) object is created（每一个依赖定义都会创建一个Dependency对象）. You can use this object to query or further configure the dependency（可以使用这个对象查询或者更进一步配置依赖）.

You can also always add instances of [Dependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\Dependency.html) directly（也可以直接添加依赖实例）:

configurationName <instance>

#### External dependencies

There are two notations supported for declaring a dependency on an external module. One is a string notation formatted this way:

configurationName "group:name:version:classifier@extension"

The other is a map notation:

configurationName group: group, name: name, version: version, classifier: classifier, ext: extension

In both notations, all properties, except name, are optional.

External dependencies are represented by a [ExternalModuleDependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\ExternalModuleDependency.html).

apply plugin: 'java'

//so that we can use 'compile', 'testCompile' for dependencies

dependencies {

//for dependencies found in artifact repositories you can use

//the string notation, e.g. group:name:version

compile 'commons-lang:commons-lang:2.6'

testCompile 'org.mockito:mockito:1.9.0-rc1'

//map notation:

compile group: 'com.google.code.guice', name: 'guice', version: '1.0'

}

#### Project dependencies

To add a project dependency, you use the following notation:

configurationName project(':someProject')

The notation project(':projectA') is similar to the syntax you use when configuring a projectA in a multi-module gradle project.

By default, when you declare dependency to projectA, you actually declare dependency to the 'default' configuration of the projectA. If you need to depend on a specific configuration of projectA, use map notation for projects:

configurationName project(path: ':projectA', configuration: 'someOtherConfiguration')

Project dependencies are represented using a [ProjectDependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\ProjectDependency.html).

#### File dependencies

You can also add a dependency using a [FileCollection](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\file\FileCollection.html):

configurationName files('a file')

apply plugin: 'java'

//so that we can use 'compile', 'testCompile' for dependencies

dependencies {

//declaring arbitrary files as dependencies

compile files('hibernate.jar', 'libs/spring.jar')

//putting all jars from 'libs' onto compile classpath

compile fileTree('libs')

}

File dependencies are represented using a [SelfResolvingDependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\SelfResolvingDependency.html).

#### Dependencies to other configurations

You can add a dependency using a [Configuration](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.Configuration.html).

When the configuration is from the same project as the target configuration, the target configuration is changed to extend from the provided configuration.

When the configuration is from a different project, a project dependency is added.

#### Gradle distribution specific dependencies

It is possible to depend on certain Gradle APIs or libraries that Gradle ships with. It is particularly useful for Gradle plugin development. Example:

//Our Gradle plugin is written in groovy

apply plugin: 'groovy'

//now we can use the 'compile' configuration for declaring dependencies

dependencies {

//we will use the Groovy version that ships with Gradle:

compile localGroovy()

//our plugin requires Gradle API interfaces and classes to compile:

compile gradleApi()

//we will use the Gradle test-kit to test build logic:

testCompile gradleTestKit()

}

#### Client module dependencies

To add a client module to a configuration you can use the notation:

configurationName module(moduleNotation) {

module dependencies

}

The module notation is the same as the dependency notations described above, except that the classifier property is not available. Client modules are represented using a [ClientModule](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\ClientModule.html).

### Properties

| **Property** | **Description** |
| --- | --- |
| [components](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:components) | *incubating*  The component metadata handler for this project. The returned handler can be used for adding rules that modify the metadata of depended-on software components. |
| [modules](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:modules) | *incubating*  The component module metadata handler for this project. The returned handler can be used for adding rules that modify the metadata of depended-on software components. |

### Methods

| **Method** | **Description** |
| --- | --- |
| [add](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:add(java.lang.String, java.lang.Object))(configurationName, dependencyNotation) | Adds a dependency to the given configuration. |
| [add](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:add(java.lang.String, java.lang.Object, groovy.lang.Closure))(configurationName, dependencyNotation, configureClosure) | Adds a dependency to the given configuration, and configures the dependency using the given closure. |
| [components](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:components(org.gradle.api.Action))(configureAction) | *incubating*  Configures component metadata for this project. |
| [create](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:create(java.lang.Object))(dependencyNotation) | Creates a dependency without adding it to a configuration. |
| [create](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:create(java.lang.Object, groovy.lang.Closure))(dependencyNotation, configureClosure) | Creates a dependency without adding it to a configuration, and configures the dependency using the given closure. |
| [createArtifactResolutionQuery](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:createArtifactResolutionQuery())() | *incubating*  Creates an artifact resolution query. |
| [gradleApi](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:gradleApi())() | Creates a dependency on the API of the current version of Gradle. |
| [gradleTestKit](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:gradleTestKit())() | *incubating*  Creates a dependency on the [Gradle test-kit](https://docs.gradle.org/current/userguide/test_kit.html) API. |
| [localGroovy](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:localGroovy())() | Creates a dependency on the Groovy that is distributed with the current version of Gradle. |
| [module](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:module(java.lang.Object))(notation) | Creates a dependency on a client module. |
| [module](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:module(java.lang.Object, groovy.lang.Closure))(notation, configureClosure) | Creates a dependency on a client module. The dependency is configured using the given closure before it is returned. |
| [modules](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:modules(org.gradle.api.Action))(configureAction) | *incubating*  Configures module metadata for this project. |
| [project](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.DependencyHandler.html#org.gradle.api.artifacts.dsl.DependencyHandler:project(java.util.Map))(notation) | Creates a dependency on a project. |

### Script blocks

No script blocks

### Property details

#### [ComponentMetadataHandler](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.ComponentMetadataHandler.html) components (read-only)

*Note: This property is*[*incubating*](file:///E:\gradle-4.1\docs\userguide\feature_lifecycle.html)*and may change in a future version of Gradle.*

The component metadata handler for this project. The returned handler can be used for adding rules that modify the metadata of depended-on software components.

#### [ComponentModuleMetadataHandler](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.ComponentModuleMetadataHandler.html) modules (read-only)

*Note: This property is*[*incubating*](file:///E:\gradle-4.1\docs\userguide\feature_lifecycle.html)*and may change in a future version of Gradle.*

The component module metadata handler for this project. The returned handler can be used for adding rules that modify the metadata of depended-on software components.

### Method details

#### [Dependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\Dependency.html) add([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html) configurationName, [Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html) dependencyNotation)

Adds a dependency to the given configuration.

#### [Dependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\Dependency.html) add([String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html) configurationName, [Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html) dependencyNotation, [Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html) configureClosure)

Adds a dependency to the given configuration, and configures the dependency using the given closure.

#### void components([Action](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Action.html)<? super [ComponentMetadataHandler](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.ComponentMetadataHandler.html)> configureAction)

*Note: This method is*[*incubating*](file:///E:\gradle-4.1\docs\userguide\feature_lifecycle.html)*and may change in a future version of Gradle.*

Configures component metadata for this project.

This method executes the given action against the [ComponentMetadataHandler](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.ComponentMetadataHandler.html) for this project.

#### [Dependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\Dependency.html) create([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html" \t "_top) dependencyNotation)

Creates a dependency without adding it to a configuration.

#### [Dependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\Dependency.html) create([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html) dependencyNotation, [Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html) configureClosure)

Creates a dependency without adding it to a configuration, and configures the dependency using the given closure.

#### [ArtifactResolutionQuery](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.query.ArtifactResolutionQuery.html) createArtifactResolutionQuery()

*Note: This method is*[*incubating*](file:///E:\gradle-4.1\docs\userguide\feature_lifecycle.html)*and may change in a future version of Gradle.*

Creates an artifact resolution query.

#### [Dependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\Dependency.html) gradleApi()

Creates a dependency on the API of the current version of Gradle.

#### [Dependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\Dependency.html) gradleTestKit()

*Note: This method is*[*incubating*](file:///E:\gradle-4.1\docs\userguide\feature_lifecycle.html)*and may change in a future version of Gradle.*

Creates a dependency on the [Gradle test-kit](https://docs.gradle.org/current/userguide/test_kit.html) API.

#### [Dependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\Dependency.html) localGroovy()

Creates a dependency on the Groovy that is distributed with the current version of Gradle.

#### [Dependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\Dependency.html) module([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html) notation)

Creates a dependency on a client module.

#### [Dependency](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\artifacts\Dependency.html) module([Object](http://download.oracle.com/javase/7/docs/api/java/lang/Object.html) notation, [Closure](http://docs.groovy-lang.org/2.4.11/html/gapi/groovy/lang/Closure.html) configureClosure)

Creates a dependency on a client module. The dependency is configured using the given closure before it is returned.

#### void modules([Action](file:///E:\gradle-4.1\docs\javadoc\org\gradle\api\Action.html)<? super [ComponentModuleMetadataHandler](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.ComponentModuleMetadataHandler.html)> configureAction)

*Note: This method is*[*incubating*](file:///E:\gradle-4.1\docs\userguide\feature_lifecycle.html)*and may change in a future version of Gradle.*

Configures module metadata for this project.

This method executes the given action against the [ComponentModuleMetadataHandler](file:///E:\gradle-4.1\docs\dsl\org.gradle.api.artifacts.dsl.ComponentModuleMetadataHandler.html) for this project.

#### [Dependency](file:///E:\\gradle-4.1\\docs\\javadoc\\org\\gradle\\api\\artifacts\\Dependency.html" \t "_top) project([Map](http://download.oracle.com/javase/7/docs/api/java/util/Map.html)<[String](http://download.oracle.com/javase/7/docs/api/java/lang/String.html), ?> notation)

Creates a dependency on a project.