4.10.1

* Docs  
  [User Manual](http://docs.google.com/userguide/userguide.html)  
  [Guides and Tutorials](https://guides.gradle.org)  
  [DSL Reference](http://docs.google.com/dsl/)  
  [Javadoc](http://docs.google.com/javadoc/)  
  [Release Notes](http://docs.google.com/release-notes.html)
* [Forums](https://discuss.gradle.org/)
* [Training](https://gradle.org/training/)
* [Try Gradle Enterprise](https://gradle.com/enterprise)
* [User Manual Home](http://docs.google.com/userguide/userguide.html)
* [DSL Reference Home](http://docs.google.com/dsl/)
* [Release Notes](http://docs.google.com/release-notes.html)
  + [Properties](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#N137E8)
  + [Methods](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#N138D4)
  + [Script blocks](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#N138EA)

### Build script blocks

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

### Core types

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

### Publishing types

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

### Container types

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

### Build Cache types

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

### Input Normalization types

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

### Help Task types

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

### Task types

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

### Reporting types

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

### Eclipse/IDEA model types

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

### Eclipse/IDEA task types

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

### Native software types

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

### Native component task types

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

IdeaModule

**Table of Contents**

[Properties](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#N137E8)[Methods](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#N138D4)[Script blocks](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#N138EA)[Property details](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#N138FF)[Method details](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#N13C64)[Script block details](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#N13C86)

| API Documentation: | [IdeaModule](http://docs.google.com/javadoc/org/gradle/plugins/ide/idea/model/IdeaModule.html) |
| --- | --- |

Enables fine-tuning module details (\*.iml file) of the IDEA plugin.

Example of use with a blend of most possible properties. Typically you don't have to configure this model directly because Gradle configures it for you.

apply plugin: 'java'  
apply plugin: 'idea'  
  
//for the sake of this example, let's introduce a 'performanceTestCompile' configuration  
configurations {  
 performanceTestCompile  
 performanceTestCompile.extendsFrom(testCompile)  
}  
  
dependencies {  
 //performanceTestCompile "some.interesting:dependency:1.0"  
}  
  
idea {  
  
 //if you want parts of paths in resulting files (\*.iml, etc.) to be replaced by variables (Files)  
 pathVariables GRADLE\_HOME: file('~/cool-software/gradle')  
  
 module {  
 //if for some reason you want to add an extra sourceDirs  
 sourceDirs += file('some-extra-source-folder')  
  
 //and some extra test source dirs  
 testSourceDirs += file('some-extra-test-dir')  
  
 //and some extra resource dirs  
 resourceDirs += file('some-extra-resource-dir')  
  
 //and some extra test resource dirs  
 testResourceDirs += file('some-extra-test-resource-dir')  
  
 //and hint to mark some of existing source dirs as generated sources  
 generatedSourceDirs += file('some-extra-source-folder')  
  
 //and some extra dirs that should be excluded by IDEA  
 excludeDirs += file('some-extra-exclude-dir')  
  
 //if you don't like the name Gradle has chosen  
 name = 'some-better-name'  
  
 //if you prefer different output folders  
 inheritOutputDirs = false  
 outputDir = file('muchBetterOutputDir')  
 testOutputDir = file('muchBetterTestOutputDir')  
  
 //if you prefer different SDK than the one inherited from IDEA project  
 jdkName = '1.6'  
  
 //put our custom test dependencies onto IDEA's TEST scope  
 scopes.TEST.plus += [ configurations.performanceTestCompile ]  
  
 //if 'content root' (as IDEA calls it) of the module is different  
 contentRoot = file('my-module-content-root')  
  
 //if you love browsing Javadoc  
 downloadJavadoc = true  
  
 //and hate reading sources :)  
 downloadSources = false  
 }  
}

For tackling edge cases, users can perform advanced configuration on the resulting XML file. It is also possible to affect the way the IDEA plugin merges the existing configuration via beforeMerged and whenMerged closures.

beforeMerged and whenMerged closures receive a [Module](http://docs.google.com/javadoc/org/gradle/plugins/ide/idea/model/Module.html) parameter

Examples of advanced configuration:

apply plugin: 'java'  
apply plugin: 'idea'  
  
idea {  
 module {  
 iml {  
 //if you like to keep \*.iml in a secret folder  
 generateTo = file('secret-modules-folder')  
  
 //if you want to mess with the resulting XML in whatever way you fancy  
 withXml {  
 def node = it.asNode()  
 node.appendNode('iLoveGradle', 'true')  
 node.appendNode('butAlso', 'I find increasing pleasure tinkering with output \*.iml contents. Yeah!!!')  
 }  
  
 //closure executed after \*.iml content is loaded from existing file  
 //but before gradle build information is merged  
 beforeMerged { module ->  
 //if you want skip merging exclude dirs  
 module.excludeFolders.clear()  
 }  
  
 //closure executed after \*.iml content is loaded from existing file  
 //and after gradle build information is merged  
 whenMerged { module ->  
 //you can tinker with Module  
 }  
 }  
 }  
}

Properties

| Property | Description |
| --- | --- |
| [contentRoot](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:contentRoot) | The content root directory of the module. |
| [downloadJavadoc](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:downloadJavadoc) | Whether to download and add javadoc associated with the dependency jars. |
| [downloadSources](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:downloadSources) | Whether to download and add sources associated with the dependency jars. |
| [excludeDirs](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:excludeDirs) | Directories to be excluded. |
| [generatedSourceDirs](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:generatedSourceDirs) | Incubating  The directories containing the generated sources (both production and test sources). |
| [iml](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:iml) | See [IdeaModule.iml(org.gradle.api.Action)](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:iml(org.gradle.api.Action)) |
| [inheritOutputDirs](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:inheritOutputDirs) | If true, output directories for this module will be located below the output directory for the project; otherwise, they will be set to the directories specified by [IdeaModule.getOutputDir()](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:outputDir) and [IdeaModule.getTestOutputDir()](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:testOutputDir). |
| [jdkName](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:jdkName) | The JDK to use for this module. If null, the value of the existing or default ipr XML (inherited) is used. If it is set to inherited, the project SDK is used. Otherwise the SDK for the corresponding value of java version is used for this module. |
| [languageLevel](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:languageLevel) | Incubating  The module specific language Level to use for this module. When null, the module will inherit the language level from the idea project. |
| [name](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:name) | Configures module name, that is the name of the \*.iml file. |
| [outputDir](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:outputDir) | The output directory for production classes. If null, no entry will be created. |
| [outputFile](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:outputFile) | Configures output \*.iml file. It's *optional* because the task should configure it correctly for you (including making sure it is unique in the multi-module build). If you really need to change the output file name (or the module name) it is much easier to do it via the *moduleName* property! |
| [resourceDirs](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:resourceDirs) | Incubating  The directories containing resources. |
| [scopes](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:scopes) | The keys of this map are the IDEA scopes. Each key points to another map that has two keys, plus and minus. The values of those keys are collections of [Configuration](http://docs.google.com/dsl/org.gradle.api.artifacts.Configuration.html) objects. The files of the plus configurations are added minus the files from the minus configurations. See example below... |
| [sourceDirs](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:sourceDirs) | The directories containing the production sources. |
| [targetBytecodeVersion](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:targetBytecodeVersion) | Incubating  The module specific bytecode version to use for this module. When null, the module will inherit the bytecode version from the idea project. |
| [testOutputDir](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:testOutputDir) | The output directory for test classes. If null, no entry will be created. |
| [testResourceDirs](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:testResourceDirs) | Incubating  The directories containing the test resources. |
| [testSourceDirs](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:testSourceDirs) | The directories containing the test sources. |

Methods

| Method | Description |
| --- | --- |
| [iml](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:iml(org.gradle.api.Action))(action) | Enables advanced configuration like tinkering with the output XML or affecting the way existing \*.iml content is merged with gradle build information. |

Script blocks

| Block | Description |
| --- | --- |
| [iml](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:iml(groovy.lang.Closure)) | Enables advanced configuration like tinkering with the output XML or affecting the way existing \*.iml content is merged with gradle build information. |

Property details

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

The content root directory of the module.

For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)

Default with idea plugin:project.projectDir

boolean downloadJavadoc

Whether to download and add javadoc associated with the dependency jars.

For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)

Default with idea plugin:false

boolean downloadSources

Whether to download and add sources associated with the dependency jars.

For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)

Default with idea plugin:true

[**Set**](http://download.oracle.com/javase/8/docs/api/java/util/Set.html)<[**File**](http://download.oracle.com/javase/8/docs/api/java/io/File.html)> excludeDirs

Directories to be excluded.

For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)

Default with idea plugin:[project.buildDir, project.file('.gradle')]

[**Set**](http://download.oracle.com/javase/8/docs/api/java/util/Set.html)<[**File**](http://download.oracle.com/javase/8/docs/api/java/io/File.html)> generatedSourceDirs

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

The directories containing the generated sources (both production and test sources).

For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)

Default with idea plugin:[]

[**IdeaModuleIml**](http://docs.google.com/javadoc/org/gradle/plugins/ide/idea/model/IdeaModuleIml.html) iml (read-only)

See [IdeaModule.iml(org.gradle.api.Action)](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:iml(org.gradle.api.Action))

[**Boolean**](http://download.oracle.com/javase/8/docs/api/java/lang/Boolean.html) inheritOutputDirs

If true, output directories for this module will be located below the output directory for the project; otherwise, they will be set to the directories specified by [IdeaModule.getOutputDir()](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:outputDir) and [IdeaModule.getTestOutputDir()](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:testOutputDir).

For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)

Default with idea plugin:nullDefault with idea and java plugin:null

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

The JDK to use for this module. If null, the value of the existing or default ipr XML (inherited) is used. If it is set to inherited, the project SDK is used. Otherwise the SDK for the corresponding value of java version is used for this module.

For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)

Default with idea plugin:'inherited'

[**IdeaLanguageLevel**](http://docs.google.com/javadoc/org/gradle/plugins/ide/idea/model/IdeaLanguageLevel.html) languageLevel

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

The module specific language Level to use for this module. When null, the module will inherit the language level from the idea project.

The Idea module language level is based on the sourceCompatibility settings for the associated Gradle project.

Default with idea and java plugin:project.sourceCompatibility

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

Configures module name, that is the name of the \*.iml file.

It's *optional* because the task should configure it correctly for you. By default it will try to use the *project.name* or prefix it with a part of a *project.path* to make sure the module name is unique in the scope of a multi-module build. The 'uniqueness' of a module name is required for correct import into IDEA and the task will make sure the name is unique.

*since* 1.0-milestone-2

If your project has problems with unique names it is recommended to always run gradle idea from the root, i.e. for all subprojects. If you run the generation of the IDEA module only for a single subproject then you may have different results because the unique names are calculated based on IDEA modules that are involved in the specific build run.

If you update the module names then make sure you run gradle idea from the root, e.g. for all subprojects, including generation of IDEA project. The reason is that there may be subprojects that depend on the subproject with amended module name. So you want them to be generated as well because the module dependencies need to refer to the amended project name. Basically, for non-trivial projects it is recommended to always run gradle idea from the root.

For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)

Default with idea plugin:*${project.name}* (sometimes prefixed with parts of *${project.path}* to guarantee uniqeness)

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

The output directory for production classes. If null, no entry will be created.

For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)

Default with idea plugin:nullDefault with idea and java plugin:null

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

Configures output \*.iml file. It's *optional* because the task should configure it correctly for you (including making sure it is unique in the multi-module build). If you really need to change the output file name (or the module name) it is much easier to do it via the *moduleName* property!

Please refer to documentation on *moduleName* property. In IntelliJ IDEA the module name is the same as the name of the \*.iml file.

Default with idea plugin:#name + '.iml'

[**Set**](http://download.oracle.com/javase/8/docs/api/java/util/Set.html)<[**File**](http://download.oracle.com/javase/8/docs/api/java/io/File.html)> resourceDirs

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

The directories containing resources.

For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)

Default with idea plugin:[]Default with idea and java plugin:resource dirs from project.sourceSets.main.resources

[**Map**](http://download.oracle.com/javase/8/docs/api/java/util/Map.html)<[**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html), [**Map**](http://download.oracle.com/javase/8/docs/api/java/util/Map.html)<[**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html), [**Collection**](http://download.oracle.com/javase/8/docs/api/java/util/Collection.html)<[**Configuration**](http://docs.google.com/dsl/org.gradle.api.artifacts.Configuration.html)>>> scopes

The keys of this map are the IDEA scopes. Each key points to another map that has two keys, plus and minus. The values of those keys are collections of [Configuration](http://docs.google.com/dsl/org.gradle.api.artifacts.Configuration.html) objects. The files of the plus configurations are added minus the files from the minus configurations. See example below...

Example how to use scopes property to enable 'performanceTestCompile' dependencies in the output \*.iml file:

apply plugin: 'java'  
apply plugin: 'idea'  
  
configurations {  
 performanceTestCompile  
 performanceTestCompile.extendsFrom(testCompile)  
}  
  
dependencies {  
 //performanceTestCompile "some.interesting:dependency:1.0"  
}  
  
idea {  
 module {  
 scopes.TEST.plus += [ configurations.performanceTestCompile ]  
 }  
}

Default with idea plugin:[:]Default with idea and java plugin:

* COMPILE -> project.configurations.compile
* RUNTIME -> project.configurations.runtime - project.configurations.compile
* TEST -> project.configurations.testRuntime - project.configurations.runtime
* PROVIDED

[**Set**](http://download.oracle.com/javase/8/docs/api/java/util/Set.html)<[**File**](http://download.oracle.com/javase/8/docs/api/java/io/File.html)> sourceDirs

The directories containing the production sources.

For backward-compatibility it is set of directories containing the production sources from project.sourceSets.main.allSource.

Starting with Gradle 5.0 it is set of directories containing the production sources from project.sourceSets.main.allJava. For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)

Default with idea plugin:[]Default with idea and java plugin:

* with Gradle < 5.0: source dirs from project.sourceSets.main.allSource.
* with Gradle >= 5.0: source dirs from project.sourceSets.main.allJava.

[**JavaVersion**](http://docs.google.com/javadoc/org/gradle/api/JavaVersion.html) targetBytecodeVersion

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

The module specific bytecode version to use for this module. When null, the module will inherit the bytecode version from the idea project.

The Idea module bytecode version is based on the targetCompatibility settings for the associated Gradle project.

Default with idea and java plugin:project.targetCompatibility

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

The output directory for test classes. If null, no entry will be created.

For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)

Default with idea plugin:nullDefault with idea and java plugin:null

[**Set**](http://download.oracle.com/javase/8/docs/api/java/util/Set.html)<[**File**](http://download.oracle.com/javase/8/docs/api/java/io/File.html)> testResourceDirs

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

The directories containing the test resources.

For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)

Default with idea plugin:[]Default with idea and java plugin:resource dirs from project.sourceSets.test.resources

[**Set**](http://download.oracle.com/javase/8/docs/api/java/util/Set.html)<[**File**](http://download.oracle.com/javase/8/docs/api/java/io/File.html)> testSourceDirs

The directories containing the test sources.

For backward-compatibility it is set of directories containing the test sources from project.sourceSets.test.allSource.

Starting with Gradle 5.0 it is set of directories containing the test sources from project.sourceSets.test.allJava. For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html)

Default with idea plugin:[]Default with idea and java plugin:

* with Gradle < 5.0: source dirs from project.sourceSets.test.allSource.
* with Gradle >= 5.0: source dirs from project.sourceSets.test.allJava.

Method details

void iml([**Action**](http://docs.google.com/javadoc/org/gradle/api/Action.html)<? super [**IdeaModuleIml**](http://docs.google.com/javadoc/org/gradle/plugins/ide/idea/model/IdeaModuleIml.html)> action)

Enables advanced configuration like tinkering with the output XML or affecting the way existing \*.iml content is merged with gradle build information.

For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html).

Script block details

iml { }

Enables advanced configuration like tinkering with the output XML or affecting the way existing \*.iml content is merged with gradle build information.

For example see docs for [IdeaModule](http://docs.google.com/dsl/org.gradle.plugins.ide.idea.model.IdeaModule.html).

Delegates to:[IdeaModuleIml](http://docs.google.com/javadoc/org/gradle/plugins/ide/idea/model/IdeaModuleIml.html) from [iml](http://docs.google.com/org.gradle.plugins.ide.idea.model.IdeaModule.html#org.gradle.plugins.ide.idea.model.IdeaModule:iml)

**Docs**

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

**News**

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

**Products**

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

**Get Help**

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

Subscribe for important Gradle updates and news

Subscribe

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

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

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