4.10.1

* Docs  
  [User Manual](http://docs.google.com/userguide/userguide.html)  
  [Guides and Tutorials](https://guides.gradle.org)  
  [DSL Reference](http://docs.google.com/dsl/)  
  [Javadoc](http://docs.google.com/javadoc/)  
  [Release Notes](http://docs.google.com/release-notes.html)
* [Forums](https://discuss.gradle.org/)
* [Training](https://gradle.org/training/)
* [Try Gradle Enterprise](https://gradle.com/enterprise)
* [User Manual Home](http://docs.google.com/userguide/userguide.html)
* [DSL Reference Home](http://docs.google.com/dsl/)
* [Release Notes](http://docs.google.com/release-notes.html)
  + [Properties](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#N1A768)
  + [Methods](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#N1A797)

### 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)

ResolutionStrategy

**Table of Contents**

[Properties](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#N1A768)[Methods](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#N1A797)[Script blocks](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#N1A830)[Property details](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#N1A835)[Method details](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#N1A87A)

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

Defines the strategies around dependency resolution. For example, forcing certain dependency versions, substitutions, conflict resolutions or snapshot timeouts.

Examples:

apply plugin: 'java' //so that there are some configurations  
  
configurations.all {  
 resolutionStrategy {  
 // fail eagerly on version conflict (includes transitive dependencies)  
 // e.g. multiple different versions of the same dependency (group and name are equal)  
 failOnVersionConflict()  
  
 // prefer modules that are part of this build (multi-project or composite build) over external modules  
 preferProjectModules()  
  
 // force certain versions of dependencies (including transitive)  
 // \*append new forced modules:  
 force 'asm:asm-all:3.3.1', 'commons-io:commons-io:1.4'  
 // \*replace existing forced modules with new ones:  
 forcedModules = ['asm:asm-all:3.3.1']  
  
 // add dependency substitution rules  
 dependencySubstitution {  
 substitute module('org.gradle:api') with project(':api')  
 substitute project(':util') with module('org.gradle:util:3.0')  
 }  
  
 // cache dynamic versions for 10 minutes  
 cacheDynamicVersionsFor 10\*60, 'seconds'  
 // don't cache changing modules at all  
 cacheChangingModulesFor 0, 'seconds'  
 }  
}

Properties

| Property | Description |
| --- | --- |
| [componentSelection](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:componentSelection) | Incubating  The currently configured version selection rules object. |
| [dependencySubstitution](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:dependencySubstitution) | Incubating  The set of dependency substitution rules that are set for this configuration. |
| [forcedModules](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:forcedModules) | Returns currently configured forced modules. For more information on forcing versions see [ResolutionStrategy.force(java.lang.Object[])](http://docs.google.com/dsl/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:force(java.lang.Object%5B%5D)) |

Methods

| Method | Description |
| --- | --- |
| [activateDependencyLocking](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:activateDependencyLocking())() | Incubating  Activates dependency locking support in Gradle. Once turned on on a configuration, resolution result can be saved and then reused for subsequent builds. This enables reproducible builds when using dynamic versions. |
| [cacheChangingModulesFor](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:cacheChangingModulesFor(int,%20java.lang.String))(value, units) | Sets the length of time that changing modules will be cached, with units expressed as a String. |
| [cacheChangingModulesFor](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:cacheChangingModulesFor(int,%20java.util.concurrent.TimeUnit))(value, units) | Sets the length of time that changing modules will be cached. |
| [cacheDynamicVersionsFor](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:cacheDynamicVersionsFor(int,%20java.lang.String))(value, units) | Sets the length of time that dynamic versions will be cached, with units expressed as a String. |
| [cacheDynamicVersionsFor](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:cacheDynamicVersionsFor(int,%20java.util.concurrent.TimeUnit))(value, units) | Sets the length of time that dynamic versions will be cached. |
| [componentSelection](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:componentSelection(org.gradle.api.Action))(action) | Incubating  The componentSelection block provides rules to filter or blacklist certain components from appearing in the resolution result. |
| [dependencySubstitution](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:dependencySubstitution(org.gradle.api.Action))(action) | Incubating  Configures the set of dependency substitution rules for this configuration. The action receives an instance of [DependencySubstitutions](http://docs.google.com/dsl/org.gradle.api.artifacts.DependencySubstitutions.html) which can then be configured with substitution rules. |
| [eachDependency](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:eachDependency(org.gradle.api.Action))(rule) | Incubating  Adds a dependency substitution rule that is triggered for every dependency (including transitive) when the configuration is being resolved. The action receives an instance of [DependencyResolveDetails](http://docs.google.com/javadoc/org/gradle/api/artifacts/DependencyResolveDetails.html) that can be used to find out what dependency is being resolved and to influence the resolution process. Example: |
| [failOnVersionConflict](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:failOnVersionConflict())() | In case of conflict, Gradle by default uses the newest of conflicting versions. However, you can change this behavior. Use this method to configure the resolution to fail eagerly on any version conflict, e.g. multiple different versions of the same dependency (group and name are equal) in the same [Configuration](http://docs.google.com/dsl/org.gradle.api.artifacts.Configuration.html). The check includes both first level and transitive dependencies. See example below: |
| [force](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:force(java.lang.Object%5B%5D))(moduleVersionSelectorNotations) | Allows forcing certain versions of dependencies, including transitive dependencies. *Appends* new forced modules to be considered when resolving dependencies. |
| [preferProjectModules](http://docs.google.com/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:preferProjectModules())() | Incubating  Gradle can resolve conflicts purely by version number or prioritize project dependencies over binary. The default is *by version number*. |

Script blocks

No script blocks

Property details

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

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

The currently configured version selection rules object.

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

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

The set of dependency substitution rules that are set for this configuration.

[**Set**](http://download.oracle.com/javase/8/docs/api/java/util/Set.html)<[**ModuleVersionSelector**](http://docs.google.com/javadoc/org/gradle/api/artifacts/ModuleVersionSelector.html)> forcedModules

Returns currently configured forced modules. For more information on forcing versions see [ResolutionStrategy.force(java.lang.Object[])](http://docs.google.com/dsl/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:force(java.lang.Object%5B%5D))

Method details

[**ResolutionStrategy**](http://docs.google.com/dsl/org.gradle.api.artifacts.ResolutionStrategy.html) activateDependencyLocking()

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

Activates dependency locking support in Gradle. Once turned on on a configuration, resolution result can be saved and then reused for subsequent builds. This enables reproducible builds when using dynamic versions.

void cacheChangingModulesFor(int value, [**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html) units)

Sets the length of time that changing modules will be cached, with units expressed as a String.

A convenience method for [ResolutionStrategy.cacheChangingModulesFor(int, java.util.concurrent.TimeUnit)](http://docs.google.com/dsl/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:cacheChangingModulesFor(int,%20java.util.concurrent.TimeUnit)) with units expressed as a String. Units are resolved by calling the valueOf(String) method of [TimeUnit](http://download.oracle.com/javase/8/docs/api/java/util/concurrent/TimeUnit.html) with the upper-cased string value.

void cacheChangingModulesFor(int value, [**TimeUnit**](http://download.oracle.com/javase/8/docs/api/java/util/concurrent/TimeUnit.html) units)

Sets the length of time that changing modules will be cached.

Gradle caches the contents and artifacts of changing modules. By default, these cached values are kept for 24 hours, after which the cached entry is expired and the module is resolved again.

Use this method to provide a custom expiry time after which the cached entries for any changing module will be expired.

void cacheDynamicVersionsFor(int value, [**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html) units)

Sets the length of time that dynamic versions will be cached, with units expressed as a String.

A convenience method for [ResolutionStrategy.cacheDynamicVersionsFor(int, java.util.concurrent.TimeUnit)](http://docs.google.com/dsl/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:cacheDynamicVersionsFor(int,%20java.util.concurrent.TimeUnit)) with units expressed as a String. Units are resolved by calling the valueOf(String) method of [TimeUnit](http://download.oracle.com/javase/8/docs/api/java/util/concurrent/TimeUnit.html) with the upper-cased string value.

void cacheDynamicVersionsFor(int value, [**TimeUnit**](http://download.oracle.com/javase/8/docs/api/java/util/concurrent/TimeUnit.html) units)

Sets the length of time that dynamic versions will be cached.

Gradle keeps a cache of dynamic version => resolved version (ie 2.+ => 2.3). By default, these cached values are kept for 24 hours, after which the cached entry is expired and the dynamic version is resolved again.

Use this method to provide a custom expiry time after which the cached value for any dynamic version will be expired.

[**ResolutionStrategy**](http://docs.google.com/dsl/org.gradle.api.artifacts.ResolutionStrategy.html) componentSelection([**Action**](http://docs.google.com/javadoc/org/gradle/api/Action.html)<? super [**ComponentSelectionRules**](http://docs.google.com/dsl/org.gradle.api.artifacts.ComponentSelectionRules.html)> action)

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

The componentSelection block provides rules to filter or blacklist certain components from appearing in the resolution result.

[**ResolutionStrategy**](http://docs.google.com/dsl/org.gradle.api.artifacts.ResolutionStrategy.html) dependencySubstitution([**Action**](http://docs.google.com/javadoc/org/gradle/api/Action.html)<? super [**DependencySubstitutions**](http://docs.google.com/dsl/org.gradle.api.artifacts.DependencySubstitutions.html)> action)

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

Configures the set of dependency substitution rules for this configuration. The action receives an instance of [DependencySubstitutions](http://docs.google.com/dsl/org.gradle.api.artifacts.DependencySubstitutions.html) which can then be configured with substitution rules.

Examples:

// add dependency substitution rules  
configurations.all {  
 resolutionStrategy.dependencySubstitution {  
 // Substitute project and module dependencies  
 substitute module('org.gradle:api') with project(':api')  
 substitute project(':util') with module('org.gradle:util:3.0')  
  
 // Substitute one module dependency for another  
 substitute module('org.gradle:api:2.0') with module('org.gradle:api:2.1')  
 }  
}

[**ResolutionStrategy**](http://docs.google.com/dsl/org.gradle.api.artifacts.ResolutionStrategy.html) eachDependency([**Action**](http://docs.google.com/javadoc/org/gradle/api/Action.html)<? super [**DependencyResolveDetails**](http://docs.google.com/javadoc/org/gradle/api/artifacts/DependencyResolveDetails.html)> rule)

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

Adds a dependency substitution rule that is triggered for every dependency (including transitive) when the configuration is being resolved. The action receives an instance of [DependencyResolveDetails](http://docs.google.com/javadoc/org/gradle/api/artifacts/DependencyResolveDetails.html) that can be used to find out what dependency is being resolved and to influence the resolution process. Example:

configurations {  
 compile.resolutionStrategy {  
 eachDependency { DependencyResolveDetails details ->  
 //specifying a fixed version for all libraries with 'org.gradle' group  
 if (details.requested.group == 'org.gradle') {  
 details.useVersion '1.4'  
 }  
 }  
 eachDependency { details ->  
 //multiple actions can be specified  
 if (details.requested.name == 'groovy-all') {  
 //changing the name:  
 details.useTarget group: details.requested.group, name: 'groovy', version: details.requested.version  
 }  
 }  
 }  
}

The rules are evaluated in order they are declared. Rules are evaluated after forced modules are applied (see [ResolutionStrategy.force(java.lang.Object[])](http://docs.google.com/dsl/org.gradle.api.artifacts.ResolutionStrategy.html#org.gradle.api.artifacts.ResolutionStrategy:force(java.lang.Object%5B%5D))

[**ResolutionStrategy**](http://docs.google.com/dsl/org.gradle.api.artifacts.ResolutionStrategy.html) failOnVersionConflict()

In case of conflict, Gradle by default uses the newest of conflicting versions. However, you can change this behavior. Use this method to configure the resolution to fail eagerly on any version conflict, e.g. multiple different versions of the same dependency (group and name are equal) in the same [Configuration](http://docs.google.com/dsl/org.gradle.api.artifacts.Configuration.html). The check includes both first level and transitive dependencies. See example below:

apply plugin: 'java' //so that there are some configurations  
  
configurations.all {  
 resolutionStrategy.failOnVersionConflict()  
}

[**ResolutionStrategy**](http://docs.google.com/dsl/org.gradle.api.artifacts.ResolutionStrategy.html) force([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html)... moduleVersionSelectorNotations)

Allows forcing certain versions of dependencies, including transitive dependencies. *Appends* new forced modules to be considered when resolving dependencies.

It accepts following notations:

* String in a format of: 'group:name:version', for example: 'org.gradle:gradle-core:1.0'
* instance of [ModuleVersionSelector](http://docs.google.com/javadoc/org/gradle/api/artifacts/ModuleVersionSelector.html)
* any collection or array of above will be automatically flattened

Example:

apply plugin: 'java' //so that there are some configurations  
  
configurations.all {  
 resolutionStrategy.force 'asm:asm-all:3.3.1', 'commons-io:commons-io:1.4'  
}

void preferProjectModules()

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

Gradle can resolve conflicts purely by version number or prioritize project dependencies over binary. The default is *by version number*.

This applies to both first level and transitive dependencies. See example below:

apply plugin: 'java' //so that there are some configurations  
  
configurations.all {  
 resolutionStrategy.preferProjectModules()  
}

**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/)