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.Task.html#N18744)
  + [Methods](http://docs.google.com/org.gradle.api.Task.html#N1884A)

### 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)

Task

**Table of Contents**

[Task Actions](http://docs.google.com/org.gradle.api.Task.html#N1863B)[Task Dependencies and Task Ordering](http://docs.google.com/org.gradle.api.Task.html#N18671)[Using a Task in a Build File](http://docs.google.com/org.gradle.api.Task.html#N186F6)[Properties](http://docs.google.com/org.gradle.api.Task.html#N18744)[Methods](http://docs.google.com/org.gradle.api.Task.html#N1884A)[Script blocks](http://docs.google.com/org.gradle.api.Task.html#N1891E)[Property details](http://docs.google.com/org.gradle.api.Task.html#N18923)[Method details](http://docs.google.com/org.gradle.api.Task.html#N18AA4)

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

A Task represents a single atomic piece of work for a build, such as compiling classes or generating javadoc.

Each task belongs to a [Project](http://docs.google.com/dsl/org.gradle.api.Project.html). You can use the various methods on [TaskContainer](http://docs.google.com/dsl/org.gradle.api.tasks.TaskContainer.html) to create and lookup task instances. For example, [TaskContainer.create(java.lang.String)](http://docs.google.com/dsl/org.gradle.api.tasks.TaskContainer.html#org.gradle.api.tasks.TaskContainer: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, which is unique across all tasks in all projects. The path is the concatenation of the owning project's path and the task's name. Path elements are separated using the : character.

Task Actions

A Task is made up of a sequence of [Action](http://docs.google.com/javadoc/org/gradle/api/Action.html) objects. When the task is executed, each of the actions is executed in turn, by calling [Action.execute(T)](http://docs.google.com/javadoc/org/gradle/api/Action.html#execute-T-). You can add actions to a task by calling [Task.doFirst(org.gradle.api.Action)](http://docs.google.com/dsl/org.gradle.api.Task.html#org.gradle.api.Task:doFirst(org.gradle.api.Action)) or [Task.doLast(org.gradle.api.Action)](http://docs.google.com/dsl/org.gradle.api.Task.html#org.gradle.api.Task:doLast(org.gradle.api.Action)).

Groovy closures 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)](http://docs.google.com/dsl/org.gradle.api.Task.html#org.gradle.api.Task:doFirst(groovy.lang.Closure)) or [Task.doLast(groovy.lang.Closure)](http://docs.google.com/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](http://docs.google.com/javadoc/org/gradle/api/tasks/StopActionException.html). A task action can abort execution of the task and continue to the next task by throwing a [StopExecutionException](http://docs.google.com/javadoc/org/gradle/api/tasks/StopExecutionException.html). 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.

Dependencies to a task are controlled using [Task.dependsOn(java.lang.Object[])](http://docs.google.com/dsl/org.gradle.api.Task.html#org.gradle.api.Task:dependsOn(java.lang.Object%5B%5D)) or [Task.setDependsOn(java.lang.Iterable)](http://docs.google.com/javadoc/org/gradle/api/Task.html#setDependsOn-java.lang.Iterable-), and [Task.mustRunAfter(java.lang.Object[])](http://docs.google.com/dsl/org.gradle.api.Task.html#org.gradle.api.Task:mustRunAfter(java.lang.Object%5B%5D)), [Task.setMustRunAfter(java.lang.Iterable)](http://docs.google.com/javadoc/org/gradle/api/Task.html#setMustRunAfter-java.lang.Iterable-), [Task.shouldRunAfter(java.lang.Object[])](http://docs.google.com/javadoc/org/gradle/api/Task.html#shouldRunAfter-java.lang.Object%5B%5D-) and [Task.setShouldRunAfter(java.lang.Iterable)](http://docs.google.com/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](http://docs.google.com/dsl/org.gradle.api.Project.html). This allows you to refer to tasks in other projects.
* A [Task](http://docs.google.com/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](http://docs.google.com/javadoc/org/gradle/api/tasks/TaskDependency.html) object.
* A [TaskReference](http://docs.google.com/javadoc/org/gradle/api/tasks/TaskReference.html) object.
* A [Buildable](http://docs.google.com/dsl/org.gradle.api.Buildable.html) object.
* A [RegularFileProperty](http://docs.google.com/javadoc/org/gradle/api/file/RegularFileProperty.html) or [DirectoryProperty](http://docs.google.com/javadoc/org/gradle/api/file/DirectoryProperty.html).
* A [Provider](http://docs.google.com/javadoc/org/gradle/api/provider/Provider.html) object. May contain any of the types listed here.
* 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.
* Anything else is treated as a failure.

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)](http://docs.google.com/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)](http://docs.google.com/dsl/org.gradle.api.Task.html#org.gradle.api.Task:setProperty(java.lang.String,%20java.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](http://docs.google.com/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](http://docs.google.com/javadoc/org/gradle/api/Plugin.html) may add methods to a Task using its [Convention](http://docs.google.com/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](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:actions) | The sequence of [Action](http://docs.google.com/javadoc/org/gradle/api/Action.html) objects which will be executed by this task, in the order of execution. |
| [ant](http://docs.google.com/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](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:convention) | The [Convention](http://docs.google.com/javadoc/org/gradle/api/plugins/Convention.html) object for this task. A [Plugin](http://docs.google.com/javadoc/org/gradle/api/Plugin.html) can use the convention object to contribute properties and methods to this task. |
| [dependsOn](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:dependsOn) | The dependencies of this task. |
| [description](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:description) | The description of this task. |
| [destroyables](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:destroyables) | Incubating  The destroyables of this task. |
| [didWork](http://docs.google.com/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](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:enabled) | Returns if this task is enabled or not. |
| [extensions](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:extensions) | The container of extensions. |
| [finalizedBy](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:finalizedBy) | Incubating  Returns tasks that finalize this task. |
| [group](http://docs.google.com/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](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:inputs) | The inputs of this task. |
| [logger](http://docs.google.com/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](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:logging) | The [LoggingManager](http://docs.google.com/javadoc/org/gradle/api/logging/LoggingManager.html) which can be used to receive logging and to control the standard output/error capture for this 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](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:mustRunAfter) | Incubating  Returns tasks that this task must run after. |
| [name](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:name) | The name of this task. The name uniquely identifies the task within its [Project](http://docs.google.com/dsl/org.gradle.api.Project.html). |
| [outputs](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:outputs) | The outputs of this task. |
| [path](http://docs.google.com/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](http://docs.google.com/dsl/org.gradle.api.Project.html) plus the name of the task, separated by :. |
| [project](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:project) | The [Project](http://docs.google.com/dsl/org.gradle.api.Project.html) which this task belongs to. |
| [state](http://docs.google.com/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](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:taskDependencies) | Returns a [TaskDependency](http://docs.google.com/javadoc/org/gradle/api/tasks/TaskDependency.html) which contains all the tasks that this task depends on. |
| [temporaryDir](http://docs.google.com/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](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:deleteAllActions())() | Deprecated  Removes all the actions of this task. |
| [dependsOn](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:dependsOn(java.lang.Object%5B%5D))(paths) | Adds the given dependencies to this task. See [here](http://docs.google.com/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](http://docs.google.com/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](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:doFirst(java.lang.String,%20org.gradle.api.Action))(actionName, action) | Incubating  Adds the given [Action](http://docs.google.com/javadoc/org/gradle/api/Action.html) to the beginning of this task's action list. |
| [doFirst](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:doFirst(org.gradle.api.Action))(action) | Adds the given [Action](http://docs.google.com/javadoc/org/gradle/api/Action.html) to the beginning of this task's action list. |
| [doLast](http://docs.google.com/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](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:doLast(java.lang.String,%20org.gradle.api.Action))(actionName, action) | Incubating  Adds the given [Action](http://docs.google.com/javadoc/org/gradle/api/Action.html) to the end of this task's action list. |
| [doLast](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:doLast(org.gradle.api.Action))(action) | Adds the given [Action](http://docs.google.com/javadoc/org/gradle/api/Action.html) to the end of this task's action list. |
| [finalizedBy](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:finalizedBy(java.lang.Object%5B%5D))(paths) | Incubating  Adds the given finalizer tasks for this task. |
| [hasProperty](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:hasProperty(java.lang.String))(propertyName) | Determines if this task has the given property. See [here](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task.properties) for details of the properties which are available for a task. |
| [leftShift](http://docs.google.com/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](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:mustRunAfter(java.lang.Object%5B%5D))(paths) | Incubating  Specifies that this task must run after all of the supplied tasks. |
| [onlyIf](http://docs.google.com/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](http://docs.google.com/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](http://docs.google.com/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](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task:setProperty(java.lang.String,%20java.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/8/docs/api/java/util/List.html)<[**Action**](http://docs.google.com/javadoc/org/gradle/api/Action.html)<? super [**Task**](http://docs.google.com/dsl/org.gradle.api.Task.html)>> actions

The sequence of [Action](http://docs.google.com/javadoc/org/gradle/api/Action.html) objects which will be executed by this task, in the order of execution.

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

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

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

The [Convention](http://docs.google.com/javadoc/org/gradle/api/plugins/Convention.html) object for this task. A [Plugin](http://docs.google.com/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/8/docs/api/java/util/Set.html)<[**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html)> dependsOn

The dependencies of this task.

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

The description of this task.

[**TaskDestroyables**](http://docs.google.com/javadoc/org/gradle/api/tasks/TaskDestroyables.html) destroyables (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 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**](http://docs.google.com/javadoc/org/gradle/api/plugins/ExtensionContainer.html) extensions (read-only)

The container of extensions.

[**TaskDependency**](http://docs.google.com/javadoc/org/gradle/api/tasks/TaskDependency.html) finalizedBy

Note: This property is [incubating](http://docs.google.com/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/8/docs/api/java/lang/String.html) 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**](http://docs.google.com/javadoc/org/gradle/api/tasks/TaskInputs.html) inputs (read-only)

The inputs of this task.

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

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

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

The [LoggingManager](http://docs.google.com/javadoc/org/gradle/api/logging/LoggingManager.html) which can be used to receive logging and to control the standard output/error capture for this 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**](http://docs.google.com/javadoc/org/gradle/api/tasks/TaskDependency.html) mustRunAfter

Note: This property is [incubating](http://docs.google.com/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/8/docs/api/java/lang/String.html) name (read-only)

The name of this task. The name uniquely identifies the task within its [Project](http://docs.google.com/dsl/org.gradle.api.Project.html).

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

The outputs of this task.

[**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html) 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](http://docs.google.com/dsl/org.gradle.api.Project.html) plus the name of the task, separated by :.

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

The [Project](http://docs.google.com/dsl/org.gradle.api.Project.html) which this task belongs to.

[**TaskState**](http://docs.google.com/javadoc/org/gradle/api/tasks/TaskState.html) 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**](http://docs.google.com/javadoc/org/gradle/api/tasks/TaskDependency.html) taskDependencies (read-only)

Returns a [TaskDependency](http://docs.google.com/javadoc/org/gradle/api/tasks/TaskDependency.html) which contains all the tasks that this task depends on.

[**File**](http://download.oracle.com/javase/8/docs/api/java/io/File.html) 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**](http://docs.google.com/dsl/org.gradle.api.Task.html) deleteAllActions()

Note: This method is [deprecated](http://docs.google.com/userguide/feature_lifecycle.html) and will be removed in the next major version of Gradle.

Removes all the actions of this task.

[**Task**](http://docs.google.com/dsl/org.gradle.api.Task.html) dependsOn([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html)... paths)

Adds the given dependencies to this task. See [here](http://docs.google.com/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**](http://docs.google.com/dsl/org.gradle.api.Task.html) doFirst([**Closure**](http://docs.groovy-lang.org/2.4.15/html/gapi/groovy/lang/Closure.html) 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**](http://docs.google.com/dsl/org.gradle.api.Task.html) doFirst([**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html) actionName, [**Action**](http://docs.google.com/javadoc/org/gradle/api/Action.html)<? super [**Task**](http://docs.google.com/dsl/org.gradle.api.Task.html)> action)

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

Adds the given [Action](http://docs.google.com/javadoc/org/gradle/api/Action.html) to the beginning of this task's action list.

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

Adds the given [Action](http://docs.google.com/javadoc/org/gradle/api/Action.html) to the beginning of this task's action list.

[**Task**](http://docs.google.com/dsl/org.gradle.api.Task.html) doLast([**Closure**](http://docs.groovy-lang.org/2.4.15/html/gapi/groovy/lang/Closure.html) 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**](http://docs.google.com/dsl/org.gradle.api.Task.html) doLast([**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html) actionName, [**Action**](http://docs.google.com/javadoc/org/gradle/api/Action.html)<? super [**Task**](http://docs.google.com/dsl/org.gradle.api.Task.html)> action)

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

Adds the given [Action](http://docs.google.com/javadoc/org/gradle/api/Action.html) to the end of this task's action list.

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

Adds the given [Action](http://docs.google.com/javadoc/org/gradle/api/Action.html) to the end of this task's action list.

[**Task**](http://docs.google.com/dsl/org.gradle.api.Task.html) finalizedBy([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html)... paths)

Note: This method is [incubating](http://docs.google.com/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](http://docs.google.com/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/8/docs/api/java/lang/String.html) propertyName)

Determines if this task has the given property. See [here](http://docs.google.com/org.gradle.api.Task.html#org.gradle.api.Task.properties) for details of the properties which are available for a task.

[**Task**](http://docs.google.com/dsl/org.gradle.api.Task.html) leftShift([**Closure**](http://docs.groovy-lang.org/2.4.15/html/gapi/groovy/lang/Closure.html) action)

Note: This method is [deprecated](http://docs.google.com/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**](http://docs.google.com/dsl/org.gradle.api.Task.html) mustRunAfter([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html)... paths)

Note: This method is [incubating](http://docs.google.com/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](http://docs.google.com/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/2.4.15/html/gapi/groovy/lang/Closure.html) 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**](http://docs.google.com/javadoc/org/gradle/api/specs/Spec.html)<? super [**Task**](http://docs.google.com/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/8/docs/api/java/lang/Object.html) property([**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html) 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/2.4.15/html/gapi/groovy/lang/MissingPropertyException.html)

void setProperty([**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html) name, [**Object**](http://download.oracle.com/javase/8/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/2.4.15/html/gapi/groovy/lang/MissingPropertyException.html) is thrown.

**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/)