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.plugins.quality.Checkstyle.html#N1DBB3)
  + [Methods](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#N1DC44)
  + [Script blocks](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#N1DCC3)

### 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)

Checkstyle

**Table of Contents**

[Properties](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#N1DBB3)[Methods](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#N1DC44)[Script blocks](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#N1DCC3)[Property details](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#N1DCD8)[Method details](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#N1DE2F)[Script block details](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#N1DF31)

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

Runs Checkstyle against some source files.

Properties

| Property | Description |
| --- | --- |
| [checkstyleClasspath](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:checkstyleClasspath) | The class path containing the Checkstyle library to be used. |
| [classpath](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:classpath) | The class path containing the compiled classes for the source files to be analyzed. |
| [config](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:config) | Incubating  The Checkstyle configuration to use. Replaces the configFile property. |
| [configDir](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:configDir) | Incubating  Path to other Checkstyle configuration files. |
| [configFile](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:configFile) | The Checkstyle configuration file to use. |
| [configProperties](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:configProperties) | The properties available for use in the configuration file. These are substituted into the configuration file. |
| [excludes](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:excludes) | The set of exclude patterns. |
| [ignoreFailures](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:ignoreFailures) | Whether this task will ignore failures and continue running the build. |
| [includes](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:includes) | The set of include patterns. |
| [maxErrors](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:maxErrors) | The maximum number of errors that are tolerated before breaking the build or setting the failure property. |
| [maxWarnings](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:maxWarnings) | The maximum number of warnings that are tolerated before breaking the build or setting the failure property. |
| [reports](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:reports) | The reports to be generated by this task. |
| [showViolations](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:showViolations) | Whether rule violations are to be displayed on the console. |
| [source](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:source) | The source for this task, after the include and exclude patterns have been applied. Ignores source files which do not exist. |

Methods

| Method | Description |
| --- | --- |
| [exclude](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:exclude(groovy.lang.Closure))(excludeSpec) | Adds an exclude spec. This method may be called multiple times to append new specs.The given closure is passed a [FileTreeElement](http://docs.google.com/javadoc/org/gradle/api/file/FileTreeElement.html) as its parameter. The closure should return true or false. Example: |
| [exclude](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:exclude(java.lang.Iterable))(excludes) | Adds an ANT style exclude pattern. This method may be called multiple times to append new patterns and multiple patterns may be specified in a single call. If excludes are not provided, then no files will be excluded. If excludes are provided, then files must not match any exclude pattern to be processed. |
| [exclude](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:exclude(java.lang.String%5B%5D))(excludes) | Adds an ANT style exclude pattern. This method may be called multiple times to append new patterns and multiple patterns may be specified in a single call. If excludes are not provided, then no files will be excluded. If excludes are provided, then files must not match any exclude pattern to be processed. |
| [exclude](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:exclude(org.gradle.api.specs.Spec))(excludeSpec) | Adds an exclude spec. This method may be called multiple times to append new specs. If excludes are not provided, then no files will be excluded. If excludes are provided, then files must not match any exclude pattern to be processed. |
| [include](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:include(groovy.lang.Closure))(includeSpec) | Adds an include spec. This method may be called multiple times to append new specs. The given closure is passed a [FileTreeElement](http://docs.google.com/javadoc/org/gradle/api/file/FileTreeElement.html) as its parameter. If includes are not provided, then all files in this container will be included. If includes are provided, then a file must match at least one of the include patterns or specs to be included. |
| [include](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:include(java.lang.Iterable))(includes) | Adds an ANT style include pattern. This method may be called multiple times to append new patterns and multiple patterns may be specified in a single call. If includes are not provided, then all files in this container will be included. If includes are provided, then a file must match at least one of the include patterns to be processed. |
| [include](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:include(java.lang.String%5B%5D))(includes) | Adds an ANT style include pattern. This method may be called multiple times to append new patterns and multiple patterns may be specified in a single call. If includes are not provided, then all files in this container will be included. If includes are provided, then a file must match at least one of the include patterns to be processed. |
| [include](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:include(org.gradle.api.specs.Spec))(includeSpec) | Adds an include spec. This method may be called multiple times to append new specs. If includes are not provided, then all files in this container will be included. If includes are provided, then a file must match at least one of the include patterns or specs to be included. |
| [reports](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:reports(org.gradle.api.Action))(configureAction) | Configures the reports to be generated by this task. The contained reports can be configured by name and closures. Example: |
| [source](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:source(java.lang.Object%5B%5D))(sources) | Adds some source to this task. The given source objects will be evaluated as per [Project.files(java.lang.Object[])](http://docs.google.com/dsl/org.gradle.api.Project.html#org.gradle.api.Project:files(java.lang.Object%5B%5D)). |

Script blocks

| Block | Description |
| --- | --- |
| [reports](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:reports(groovy.lang.Closure)) | Configures the reports to be generated by this task. The contained reports can be configured by name and closures. Example: |

Property details

[**FileCollection**](http://docs.google.com/javadoc/org/gradle/api/file/FileCollection.html) checkstyleClasspath

The class path containing the Checkstyle library to be used.

Default with checkstyle plugin:project.configurations.checkstyle

[**FileCollection**](http://docs.google.com/javadoc/org/gradle/api/file/FileCollection.html) classpath

The class path containing the compiled classes for the source files to be analyzed.

Default with checkstyle plugin:*sourceSet*.output

[**TextResource**](http://docs.google.com/dsl/org.gradle.api.resources.TextResource.html) config

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

The Checkstyle configuration to use. Replaces the configFile property.

Default with checkstyle plugin:project.checkstyle.config

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

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

Path to other Checkstyle configuration files.

This path will be exposed as the variable config\_loc in Checkstyle's configuration files.

Default with checkstyle plugin:project.checkstyle.configDir

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

The Checkstyle configuration file to use.

Default with checkstyle plugin:project.checkstyle.configFile

[**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), [**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html)> configProperties

The properties available for use in the configuration file. These are substituted into the configuration file.

Default with checkstyle plugin:project.checkstyle.configProperties

[**Set**](http://download.oracle.com/javase/8/docs/api/java/util/Set.html)<[**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html)> excludes

The set of exclude patterns.

boolean ignoreFailures

Whether this task will ignore failures and continue running the build.

Default with checkstyle plugin:project.checkstyle.ignoreFailures

[**Set**](http://download.oracle.com/javase/8/docs/api/java/util/Set.html)<[**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html)> includes

The set of include patterns.

int maxErrors

The maximum number of errors that are tolerated before breaking the build or setting the failure property.

Default with checkstyle plugin:project.checkstyle.maxErrors

int maxWarnings

The maximum number of warnings that are tolerated before breaking the build or setting the failure property.

Default with checkstyle plugin:project.checkstyle.maxWarnings

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

The reports to be generated by this task.

boolean showViolations

Whether rule violations are to be displayed on the console.

Default with checkstyle plugin:project.checkstyle.showViolations

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

The source for this task, after the include and exclude patterns have been applied. Ignores source files which do not exist.

The sources for this task are relatively relocatable even though it produces output that includes absolute paths. This is a compromise made to ensure that results can be reused between different builds. The downside is that up-to-date results, or results loaded from cache can show different absolute paths than would be produced if the task was executed.

Default with checkstyle plugin:*sourceSet*.allJava

Method details

[**SourceTask**](http://docs.google.com/dsl/org.gradle.api.tasks.SourceTask.html) exclude([**Closure**](http://docs.groovy-lang.org/2.4.15/html/gapi/groovy/lang/Closure.html) excludeSpec)

Adds an exclude spec. This method may be called multiple times to append new specs.The given closure is passed a [FileTreeElement](http://docs.google.com/javadoc/org/gradle/api/file/FileTreeElement.html) as its parameter. The closure should return true or false. Example:

copySpec {  
 from 'source'  
 into 'destination'  
 //an example of excluding files from certain configuration:  
 exclude { it.file in configurations.someConf.files }  
}

If excludes are not provided, then no files will be excluded. If excludes are provided, then files must not match any exclude pattern to be processed.

[**SourceTask**](http://docs.google.com/dsl/org.gradle.api.tasks.SourceTask.html) exclude([**Iterable**](http://download.oracle.com/javase/8/docs/api/java/lang/Iterable.html)<[**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html)> excludes)

Adds an ANT style exclude pattern. This method may be called multiple times to append new patterns and multiple patterns may be specified in a single call. If excludes are not provided, then no files will be excluded. If excludes are provided, then files must not match any exclude pattern to be processed.

[**SourceTask**](http://docs.google.com/dsl/org.gradle.api.tasks.SourceTask.html) exclude([**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html)... excludes)

Adds an ANT style exclude pattern. This method may be called multiple times to append new patterns and multiple patterns may be specified in a single call. If excludes are not provided, then no files will be excluded. If excludes are provided, then files must not match any exclude pattern to be processed.

[**SourceTask**](http://docs.google.com/dsl/org.gradle.api.tasks.SourceTask.html) exclude([**Spec**](http://docs.google.com/javadoc/org/gradle/api/specs/Spec.html)<[**FileTreeElement**](http://docs.google.com/javadoc/org/gradle/api/file/FileTreeElement.html)> excludeSpec)

Adds an exclude spec. This method may be called multiple times to append new specs. If excludes are not provided, then no files will be excluded. If excludes are provided, then files must not match any exclude pattern to be processed.

[**SourceTask**](http://docs.google.com/dsl/org.gradle.api.tasks.SourceTask.html) include([**Closure**](http://docs.groovy-lang.org/2.4.15/html/gapi/groovy/lang/Closure.html) includeSpec)

Adds an include spec. This method may be called multiple times to append new specs. The given closure is passed a [FileTreeElement](http://docs.google.com/javadoc/org/gradle/api/file/FileTreeElement.html) as its parameter. If includes are not provided, then all files in this container will be included. If includes are provided, then a file must match at least one of the include patterns or specs to be included.

[**SourceTask**](http://docs.google.com/dsl/org.gradle.api.tasks.SourceTask.html) include([**Iterable**](http://download.oracle.com/javase/8/docs/api/java/lang/Iterable.html)<[**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html)> includes)

Adds an ANT style include pattern. This method may be called multiple times to append new patterns and multiple patterns may be specified in a single call. If includes are not provided, then all files in this container will be included. If includes are provided, then a file must match at least one of the include patterns to be processed.

[**SourceTask**](http://docs.google.com/dsl/org.gradle.api.tasks.SourceTask.html) include([**String**](http://download.oracle.com/javase/8/docs/api/java/lang/String.html)... includes)

Adds an ANT style include pattern. This method may be called multiple times to append new patterns and multiple patterns may be specified in a single call. If includes are not provided, then all files in this container will be included. If includes are provided, then a file must match at least one of the include patterns to be processed.

[**SourceTask**](http://docs.google.com/dsl/org.gradle.api.tasks.SourceTask.html) include([**Spec**](http://docs.google.com/javadoc/org/gradle/api/specs/Spec.html)<[**FileTreeElement**](http://docs.google.com/javadoc/org/gradle/api/file/FileTreeElement.html)> includeSpec)

Adds an include spec. This method may be called multiple times to append new specs. If includes are not provided, then all files in this container will be included. If includes are provided, then a file must match at least one of the include patterns or specs to be included.

[**CheckstyleReports**](http://docs.google.com/javadoc/org/gradle/api/plugins/quality/CheckstyleReports.html) reports([**Action**](http://docs.google.com/javadoc/org/gradle/api/Action.html)<? super [**CheckstyleReports**](http://docs.google.com/javadoc/org/gradle/api/plugins/quality/CheckstyleReports.html)> configureAction)

Configures the reports to be generated by this task. The contained reports can be configured by name and closures. Example:

checkstyleTask {  
 reports {  
 html {  
 destination "build/codenarc.html"  
 }  
 }  
}

[**SourceTask**](http://docs.google.com/dsl/org.gradle.api.tasks.SourceTask.html) source([**Object**](http://download.oracle.com/javase/8/docs/api/java/lang/Object.html)... sources)

Adds some source to this task. The given source objects will be evaluated as per [Project.files(java.lang.Object[])](http://docs.google.com/dsl/org.gradle.api.Project.html#org.gradle.api.Project:files(java.lang.Object%5B%5D)).

Script block details

reports { }

Configures the reports to be generated by this task. The contained reports can be configured by name and closures. Example:

checkstyleTask {  
 reports {  
 html {  
 destination "build/codenarc.html"  
 }  
 }  
}

Delegates to:[CheckstyleReports](http://docs.google.com/javadoc/org/gradle/api/plugins/quality/CheckstyleReports.html) from [reports](http://docs.google.com/org.gradle.api.plugins.quality.Checkstyle.html#org.gradle.api.plugins.quality.Checkstyle:reports)

**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/)