Jtest Goals Reference for Maven

This section provides information about the Jtest goals for Maven and their parameters. The following goals are available:

- jtest:jtest
- jtest:agent
- jtest:monitor
- cbt:affected-tests

jtest:jtest

Parameters | Parameter Details

Full name: com.parasoft.jtest:jtest-maven-plugin:1.2.13:jtest

This goal should be used as a Maven report. It collects the build data and executes Jtest with the configured parameters (see Configuring the Jtest Plugin for Maven). It is called at the end of the build session and collects the aggregated data of every project.

The goal works as an aggregator, which means it is not tied to a build lifecycle phase. It can be executed as a build plugin or as a reporting plugin (the `site` phase); see Configuring the Jtest Plugin for Maven.

Attributes

- Requires a Maven project to be executed.
- Executes as an aggregator plugin.
- Requires dependency resolution of artifacts in scope: **compile**.
- Since version: 1.0.0

Parameters

Name	Accepted values / data type	Since	Description
compilation	PlexusConfiguration	1.0.4	A description of manual configuration data (see Manual Customization of Compilation Data)
compilations	PlexusConfiguration	1.0.4	A list of descriptions of manual configuration data (see Manual Customization of Compilation Data)
config	String	1.0.0	The name of a built-in, DTP, or user-defined test configuration. **User property is:** `jtest.config`
dataUpdate	String	1.2.9	Allows you to manually update compilation data collected from the build. **User property is**: `jtest.dataUpdate`
exclude	String	1.0.0	Specifies a subset of files or packages from the selected resources that will be excluded from analysis. You can provide the qualified names or use Ant glob patterns to match more filenames. **User property is**: `jtest.exclude`
excludes		1.1.0	A list of exclude parameters. **User property is**: `jtest.excludes`
excludeTestSources	true | false	1.1.0	Excludes test source code from analysis. **Default value is**: `true` **User property is**: `jtest.excludeTestSources`.
fail	true | false	1.0.0	Fails the build if any violation is reported. **Default value is**: `false` **User property is**: `jtest.fail`
forceTestReportsImport	true | false	1.0.0	Ensures that test results are imported from the default location if the Jtest Plugin fails to recognize the existing test plugin (for example, Surefire) in the Maven environment. **Default value is**: `false` **User property is**: `jtest.forceTestReportsImport`
home	Path	1.1.0	Specifies the Jtest installation directory. **User property is**: `jtest.home`

ignoredIds		1.0.4	A list of ignored compilation IDs. **User property is**: `jtest.ignoredids`
include	String	1.0.0	Specifies a subset of files or packages form the selected resources that will be included during analysis. **User property is**: `jtest.include`
includes		1.1.0	A list of include parameters. **User property is**: `jtest.includes`
projectNameTemplate	String	1.0.0	The project name template. This parameter allows you to configure the pattern with the options [groupId], [artifactId], [version], and [basedir]. **User property is**: `jtest.projectNameTemplate`
publish	true | false	1.0.0	Enables reporting results of local analysis to the DTP server. **User property is**: `jtest.publish`
report	Path	1.0.0	Specifies the directory where the report will be created. **User property is**: `jtest.report`
resource	String	1.0.0	Specifies the input scope for analysis. If no resources are specified, Jtest will analyze resources from every built project. **User property is**: `jtest.resource`
resources		1.1.0	A list of resource parameters. **User property is**: `jtest.resources`
settings	Path	1.0.0	An absolute or relative path to the *properties file that includes custom configuration settings. **User property is**: jtest.settings
settingsList		1.0.0	A list of settings parameters. **User property is**: `jtest.settingsList`
showDetails	true | false	1.0.0	Displays detailed progress information. **User property is**: jtest.showdetails
showSettings	true | false	1.0.0	Prints the current settings and customizations. **User property is**: jtest.showsettings
skip	true | false	1.1.0	Allows you to skip the Jtest execution phase. If set to `true`, only the jtest.data.json data file is generated. **Default value is**: false **User property is**: jtest.skip

Parameter Details

compilation

A description of manual configuration data (see Manual Customization of Compilation Data).

- **Type**: org.codehaus.plexus.configuration.PlexusConfiguration
- **Since**: 1.0.4
- **Required**: No

compilations

A list of descriptions of manual configuration data (see Manual Customization of Compilation Data)

- **Type**: org.codehaus.plexus.configuration.PlexusConfiguration
- **Since**: 1.0.4
- **Required**: No

config

The name of a test configuration. The following configuration types are supported:

- built-in test configurations, for example `"builtin://Recommended Rules"`
- dtp test configurations, for example `"dtp://New Config"`
- user-defined test configurations, for example `"user://Your Config"`

User-defined test configurations should be stored in the [INSTALL_DIR]/configs/user directory as *.properties files.

- **Type**: java.lang.String
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.config`

dataUpdate

Allows you to manually update compilation data collected from the build.

- **Supported actions**: prepend, append, and set.
- **Supported data types**: classpath, bootpath, sourcepath, binarypath, sourcecode (set only) and encoding (set only)
- **Scope**: all collected projects data will be updated
- **Type**: java.lang.String
- **Since**: 1.2.9
- **Required**: No
- **User property:** `jtest.dataUpdate`

See Manual Customization of Compilation Data and Compilation Data Model for details.

exclude

Specifies a subset of files or packages from the selected resources that will be excluded from analysis. You can provide the qualified names. You can use glob patterns to match more filenames.

Example 1: The following pattern excludes all files from com.parasoft package and its sub-packages: `/com/parasoft/**`

Example 2: The following pattern excludes all files from the package com.parasoft.jtest, but NOT from its sub-packages: `/com/parasoft/jtest/*`

Example 3: The following pattern excludes all files from the given directory and all subdirectories: `path:/home/user/project/src/test/java/**`

- **Type**: java.lang.String
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.exclude`

excludeTestSources

Excludes test source code from analysis.

- **Type**: java.lang.String
- **Since**: 1.1.0
- **Required**: No
- **Default**: true
- **User property**: jtest.excludeTestSources

excludes

A list of exclude parameters. Can be configured by property since Maven 3.0.3.

- **Type**: java.lang.String[]
- **Since**: 1.1.0
- **Required**: No
- **User property**: `jtest.excludes`

fail

Fails the build if any violation is reported.

- **Type**: java.lang.String
- **Since**: 1.0.0
- **Required**: No
- **Default**: false
- **User property:** `jtest.fail`

forceTestReportsImport

Ensures that test results are imported from the default location if the Jtest Plugin fails to recognize the existing test plugin (for example, Surefire) in the Maven environment.

We recommend using this parameter with the "clean" goal to prevent the Jtest Plugin from importing test results from previous executions.

- **Type**: java.lang.Boolean
- **Since**: 1.0.0
- **Required**: No
- **Default**: false
- **User property**: `jtest.forceTestReportsImport`

home

Jtest installation directory.

- **Type**: java.io.File
- **Since**: 1.1.0
- **Required**: No
- **User property**: `jtest.home`

ignoredIds

A list of ignored compilation IDs. Supported by Maven since 3.0.3.

- **Type**: java.lang.String[]
- **Since**: 1.0.4
- **Required**: No
- **User property**: `jtest.ignoredids`

include

Specifies a subset of files or packages form the selected resources that will be included during analysis. You can use glob patterns to match more filenames.

Example 1: The following pattern includes all files from com.parasoft package and its sub-packages: `/com/parasoft/**`

Example 2: The following pattern includes all files from the package com.parasoft.jtest, but NOT from its sub-packages: `/com/parasoft/jtest/*`

Example 3: The following pattern includes all files from the given directory and all subdirectories: `path:/home/user/project/src/test/java/**`

- **Type**: java.lang.String
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.include`

includes

A list of include parameters. Can be configured by property since Maven 3.0.3.

- **Type**: java.lang.String[]
- **Since**: 1.1.0
- **Required**: No
- **User property**: `jtest.includes`

projectNameTemplate

The project name template. By default, the following pattern is used to create project names in json: :"groupId:artifactId ". This parameter allows you to configure the pattern with the options [groupId], [artifactId], [version] and [basedir], for example, to make a project name compatible with the project name in Eclipse.

If you change the pattern for a project that has already been integrated with DTP, DTP will not recognize your project. It will assume that your project after the change of the pattern is a different project. For details, see the DTP documentation.

- **Type**: java.lang.String
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.projectNameTemplate`

publish

Enables reporting results of local analysis to the DTP server.

- **Type**: java.lang.String
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.publish`

report

Specifies the directory where the report will be created. By dafault it is `${project.build.directory}/jtest`. This parameter is ignored when Jtest is used as a reporting plugin (see Configuring the Jtest Plugin for Maven as a Reporting Plugin). In such case, the report is appended to the `site` in Project Reports section, and the **report** property points to the `${project.build.directory}/site` directory.

- **Type**: java.io.File
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.report`

resource

Specifies the input scope for analysis. If no resources are specified, Jtest will analyze resources from every built project.

Use the following pattern to specify the paths: **${ProjectName}/my/src/dir/my/package/file.java**". You can use the following wildcards:

- ? - matches one character
- * - matches zero or more characters, except the path separator (the **/** character)
- ** - matches entire paths, including the path separator (the **/** character)

The Jtest Plugin for Maven resolves `${ProjectName}` to `${artifactId}:${groupId}`.

- **Type**: java.lang.String
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.resource`

resources

A list of resource parameters. Can be configured by property since Maven 3.0.3.

- **Type**: java.lang.String[]
- **Since**: 1.1.0
- **Required**: No
- **User property:** `jtest.resources`

settings

An absolute or relative path to the *properties file that includes custom configuration settings.

> Some of the setting names may not be consistent with the settings dedicated for the build system plugins. See Configuration Settings for details.

- **Type**: java.io.File
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.settings`

settingsList

A list of settings parameters. Supported by Maven since 3.0.3. Can be configured by property since Maven 3.0.3.

- **Type**: java.io.File[]
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.settingsList`

showDetails

Displays detailed progress information.

- **Type**: java.lang.String
- **Since**: 1.0.0
- **Required**: No
- **User Property**: jtest.showdetails

showSettings

Prints the current settings and customizations along with the information where each of them is configured (e.g. in the jtest.properties file).

- **Type**: java.lang.String
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.showsettings`

skip

Allows you to skip the Jtest execution phase. If set to `true`, only the jtest.data.json data file is generated.

- **Type**: java.lang.String
- **Since**: 1.1.0
- **Required**: No
- **Default**: false
- **User property**: `jtest.skip`

jtest:agent

Required Parameters | Optional Parameters | Parameter Details

Full name: com.parasoft.jtest:jtest-maven-plugin:1.2.13:agent

This goal generates the Jtest Agent settings based on the build modules, and sets the javaagent VM agument in the properties specified with the agentPropertyNames parameter.

By default, it tries to inject `javaagent VmArg` into `maven-surefire-plugin` and `eclipse-test-plugin` through specific properties.

> **Command Line Execution**
>
> Ensure that tests and classes are compiled before executing the jtest:agent goal from command line (the test-compile phase).

Attributes

- Requires a Maven project to be executed.
- Requires dependency resolution of artifacts in scope: **compile.**
- Since version: 1.2.4
- Binds by default to the lifecycle phase: process-test-classes

Required Parameters

Name	Accepted values / data type	Since	Description
autostartAgent	true | false	1.0.7	A switch to automatically enable collecting coverage. **Default value is**: `true` **User property is**: `jtest.autostartAgent`
home	Path	1.1.0	Jtest installation directory. Configuring this parameter is required to run `jtestcli` and to access the Jtest Agent library. **User property is**: `jtest.home`

Optional Parameters

Name	Accepted values / data type	Since	Description
agentPropertyNames	String	1.2.4	A comma-separated list of properties that pass Jtest Agent JVM arguments. **Default value is**: argLine,tycho.testArgLine **User property is**: jtest.agentPropertyNames
agentServerEnabled	true | false	1.2.4	A switch that allows you to enable the Agent server. **Default value is**: false **User property is**: jtest.agentServerEnabled

agentSettings	Path	1.2.2	The path to the location where agent settings will be generated. **Default value is**: ${project.build.directory}/jtest/agent.properties **User property is**: jtest.agentSettings
coverage		1.2.4	A set of parameters that allow you to configure collecting coverage. • **jtest.coverage.skip** - Disables collecting coverage. • **jtest.coverage.includes** - <Glob> patterns to include classes. Coverage information will be collected. • **jtest.coverage.excludes** - <Glob> patterns to exclude classes. Coverage information will not be collected. • **jtest.coverage.testIncludes** - <Glob> patterns to include test classes. Test cases will be assigned with coverage. • **jtest.coverage.testExcludes** - <Glob> patterns to exclude test classes. Test cases will not be assigned with coverage.
ignoredIds		1.0.4	A list of ignored compilation IDs. **User property is**: jtest.ignoredids
projectNameTemplate	String	1.2.4	The project name template. **User property is**: jtest.projectNameTemplate.

Parameter Details

agentPropertyNames

A comma-separated list of properties that pass Jtest Agent JVM arguments.

- **Type**: java.lang.String
- **Since**: 1.2.4
- **Required**: No
- **User Property**: jtest.agentPropertyNames
- **Default**: argLine,tycho.testArgLine

agentServerEnabled

A switch that allows you to enable the Agent server.

- **Type**: boolean
- **Since**: 1.2.4
- **Required**: No
- **User Property**: jtest.agentServerEnabled
- **Default**: false

agentSettings

The path to the location where agent settings will be generated. You can provide either an absolute path or a path relative to the **${project.build.directory}/jtest** directory. The default location is `${project.build.directory}/jtest/agent.properties`. If you provide a path to an already existing file, the file will be overwritten.

- **Type**: java.lang.String
- **Since**: 1.2.2
- **Required**: No
- **User Property**: jtest.agentSettings
- **Default**: agent.properties

autostartAgent

A switch to automatically enable collecting coverage.

- **Type**: boolean
- **Since**: 1.0.7
- **Required**: Yes
- **User Property**: jtest.autostartAgent
- **Default**: true

coverage

A set of parameters that allow you to configure collecting coverage.

- **jtest.coverage.skip** - Disables collecting coverage.

The following parameters specify classes that will be instrumented (coverage will be collected only for instrumented classes):

- **jtest.coverage.includes** - <Glob> patterns to include classes. Coverage information will be collected.
- **jtest.coverage.excludes** - <Glob> patterns to exclude classes. Coverage information will not be collected.

The following parameters specify test classes that will be instrumented (individual instrumented test cases will be assigned with coverage information):

- **jtest.coverage.testIncludes** - <Glob> patterns to include test classes. Test cases will be assigned with coverage.
- **jtest.coverage.testExcludes** - <Glob> patterns to exclude test classes. Test cases will not be assigned with coverage.

- **Type**: com.parasoft.jtest.plugin.CoverageConfig
- **Since**: 1.2.4
- **Required**: No

home

Jtest installation directory. Configuring this parameter is required to run `jtestcli` and to access the Jtest Agent library.

- **Type**: java.io.File
- **Since**: 1.1.0
- **Required**: Yes
- **User Property**: jtest.home

ignoredIds

A list of ignored compilation IDs.

- **Type**: java.lang.String[]
- **Since**: 1.0.4
- **Required**: No
- **User Property**: jtest.ignoredids

projectNameTemplate

The project name template. You can create your own template for creating the project name in json. The following options are available: [groupId], [artifactId], [version], [basedir].

- **Type**: java.lang.String
- **Since**: 1.2.4
- **Required**: No
- **User Property**: jtest.projectNameTemplate

jtest:monitor

Optional Parameters | Parameter Details

Full name: com.parasoft.jtest:jtest-maven-plugin:1.2.13:monitor

This goal should be used as a Maven report. It scans the build to generate settings for the Jtest Java Agent, and creates a monitor package in location specified with the monitorLocation parameter.

The monitor package contains the following:

- static_coverage.xml - the file that contains static coverage information
- agent.jar - the Jtest Java coverage agent jar archive
- agent.properties - the agent settings file that contains scope parameters generated during the build process and other attributes
- agent.sh/agent.bat - the script that generates the Jtest Java agent VM arguments necessary to monitor the application

Command Line Execution

Ensure that tests and classes are compiled before executing the jtest:monitor goal from command line (the compile phase).

Attributes

- Requires a Maven project to be executed.
- Executes as an aggregator plugin.
- Requires dependency resolution of artifacts in scope: **compile.**
- Since version: 1.2.4

Optional Parameters

Name	Accepted values / data type	Since	Description
agentServerEnabled	true | false	1.2.4	A switch that allows you to enable the Agent server. **Default value is**: true **User property is**: jtest.agentServerEnabled
autostartAgent	true | false	1.0.7	A switch to automatically enable collecting coverage. **Default value is**: true **User property is**: jtest.autostartAgent
compilation	PlexusConfiguration	1.0.4	A description of manual configuration data (see Manual Customization of Compilation Data)
compilations	PlexusConfiguration	1.0.4	A list of descriptions of manual configuration data (see Manual Customization of Compilation Data)
coverage		1.2.4	The coverage configuration block. You can nest the following parameters: • **jtest.coverage.includes** - <Glob> patterns to include classes. Coverage information will be collected. • **jtest.coverage.excludes** - <Glob> patterns to exclude classes. Coverage information will not be collected. • **jtest.coverage.skip** - (Boolean) Disables collecting coverage if enabled. Glob patterns are matched to relative paths of `*.class` files from classpath (the path separator: `/`)
dataUpdate	String	1.2.9	Allows you to manually update compilation data collected from the build. **User property is**: `jtest.dataUpdate`
exclude	String	1.0.0	Specifies a subset of files or packages from the selected resources that will be excluded from analysis. You can provide the qualified names, or use Ant glob patterns to match more filenames. **User property is**: `jtest.exclude`
excludes		1.1.0	A list of exclude parameters. **User property is**: `jtest.excludes`
home	Path	1.1.0	Specifies the Jtest installation directory. **User property is**: `jtest.home`
ignoredIds		1.0.4	A list of ignored compilation IDs. **User property is**: `jtest.ignoredids`
include	String	1.0.0	Specifies a subset of files or packages form the selected resources that will be included during analysis. **User property is**: `jtest.include`
includes		1.1.0	A list of include parameters. **User property is**: `jtest.includes`
monitorLocation	Path	1.2.4	The path to the location where the monitor package will be created. **Default value is**: ${project.build.directory}/jtest/monitor/monitor.zip **User property is**: jtest.monitorLocation
projectNameTemplate	String	1.2.4	The project name template. **User property is**: `jtest.projectNameTemplate`
publish	true | false	1.0.0	Enables reporting results of local analysis to the DTP server. **User property is**: `jtest.publish`
resource	String	1.0.0	Specifies the input scope for analysis. If no resources are specified, Jtest will analyze resources from every built project. **User property is**: `jtest.resource`
resources		1.1.0	A list of resource parameters. **User property is**: `jtest.resources`
settings	Path	1.0.0	An absolute or relative path to the *properties file that includes custom configuration settings. **User property is**: `jtest.settings`
settingsList		1.0.0	A list of settings parameters. **User property is**: `jtest.settingsList`

showDetails	true | false	1.0.0	Displays detailed progress information. **User property is**: `jtest.showdetails`
showSettings	true | false	1.0.0	Prints the current settings and customizations. **User property is**: `jtest.showsettings`
skip	true | false	1.1.0	Allows you to skip the Jtest execution phase. If set to `true`, only the jtest.data.json data file is generated. **Default value is**: false **User property is**: jtest.skip

Parameter Details

agentServerEnabled

A switch that allows you to enable the Agent server.

- **Type**: boolean
- **Since**: 1.2.4
- **Required**: No
- **User Property**: jtest.agentServerEnabled
- **Default**: true

autostartAgent

A switch to automatically enable collecting coverage.

- **Type**: boolean
- **Since**: 1.0.7
- **Required**: Yes
- **User Property**: jtest.autostartAgent
- **Default**: true

compilation

A description of manual configuration data (see Manual Customization of Compilation Data).

- **Type**: org.codehaus.plexus.configuration.PlexusConfiguration
- **Since**: 1.0.4
- **Required**: No

compilations

A list of descriptions of manual configuration data (see Manual Customization of Compilation Data)

- **Type**: org.codehaus.plexus.configuration.PlexusConfiguration
- **Since**: 1.0.4
- **Required**: No

coverage

- **jtest.coverage.skip** - Disables collecting coverage.

The following parameters specify classes that will be instrumented (coverage will be collected only for instrumented classes):

- **jtest.coverage.includes** - <Glob> patterns to include classes. Coverage information will be collected.
- **jtest.coverage.excludes** - <Glob> patterns to exclude classes. Coverage information will not be collected.

Glob patterns are matched to relative paths of `*.class` files from classpath (the path separator: `/`)

- **Type**: com.parasoft.jtest.plugin.CoverageConfig
- **Since**: 1.2.4
- **Required**: No

dataUpdate

Allows you to manually update compilation data collected from the build.

- **Supported actions**: prepend, append, and set.
- **Supported data types**: classpath, bootpath, sourcepath, binarypath, sourcecode (set only) and encoding (set only)
- **Scope**: all collected projects data will be updated

- **Type**: java.lang.String
- **Since**: 1.2.9
- **Required**: No
- **User property:**`jtest.dataUpdate`

See Manual Customization of Compilation Data and Compilation Data Model for details.

exclude

Specifies a subset of files or packages from the selected resources that will be excluded from analysis. You can provide the qualified names. You can use glob patterns to match more filenames.

Example 1: The following pattern excludes all files from com.parasoft package and its sub-packages: `/com/parasoft/**`

Example 2: The following pattern excludes all files from the package com.parasoft.jtest, but NOT from its sub-packages: `/com/parasoft/jtest/*`

Example 3: The following pattern excludes all files from the given directory and all subdirectories: `path:/home/user/project/src/test/java/**`

- **Type**: java.lang.String
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.exclude`

excludes

A list of exclude parameters. Can be configured by property since Maven 3.0.3.

- **Type**: java.lang.String[]
- **Since**: 1.1.0
- **Required**: No
- **User property**: `jtest.excludes`

home

Jtest installation directory.

- **Type**: java.io.File
- **Since**: 1.1.0
- **Required**: No
- **User property**: `jtest.home`

ignoredIds

A list of ignored compilation IDs. Supported by Maven since 3.0.3.

- **Type**: java.lang.String[]
- **Since**: 1.0.4
- **Required**: No
- **User property**: `jtest.ignoredids`

include

Specifies a subset of files or packages form the selected resources that will be included during analysis. You can use glob patterns to match more filenames.

Example 1: The following pattern includes all files from com.parasoft package and its sub-packages: `/com/parasoft/**`

Example 2: The following pattern includes all files from the package com.parasoft.jtest, but NOT from its sub-packages: `/com/parasoft/jtest/*`

Example 3: The following pattern includes all files from the given directory and all subdirectories: `path:/home/user/project/src/test/java/**`

- **Type**: java.lang.String
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.include`

includes

A list of include parameters. Can be configured by property since Maven 3.0.3.

- **Type**: java.lang.String[]
- **Since**: 1.1.0
- **Required**: No

- **User property**: `jtest.includes`

monitorLocation

The path to the location where the monitor package will be created.

- **Type**: java.io.File
- **Since**: 1.2.4
- **Required**: No
- **User property**: `jtest.monitorLocation`
- **Default**: `${project.build.directory}/jtest/monitor/monitor.zip`

projectNameTemplate

The project name template. By default, the following pattern is used to create project names in json: :"groupId:artifactId ". This parameter allows you to configure the pattern with the options [groupId], [artifactId], [version] and [basedir], for example, to make a project name compatible with the project name in Eclipse.

> If you change the pattern for a project that has already been integrated with DTP, DTP will not recognize your project. It will assume that your project after the change of the pattern is a different project. For details, see the DTP documentation.

- **Type**: java.lang.String
- **Since**: 1.2.4
- **Required**: No
- **User Property**: jtest.projectNameTemplate

publish

Enables reporting results of local analysis to the DTP server.

- **Type**: java.lang.String
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.publish`

resource

Specifies the input scope for analysis. If no resources are specified, Jtest will analyze resources from every built project.

Use the following pattern to specify the paths: **${ProjectName}/my/src/dir/my/package/file.java**". You can use the following wildcards:

- ? - matches one character
- * - matches zero or more characters, except the path separator (the **/** character)
- ** - matches entire paths, including the path separator (the **/** character)

ℹ The Jtest Plugin for Maven resolves `${ProjectName}` to `${artifactId}:${groupId}`.

- **Type**: java.lang.String
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.resource`

resources

A list of resource parameters. Can be configured by property since Maven 3.0.3.

- **Type**: java.lang.String[]
- **Since**: 1.1.0
- **Required**: No
- **User property:** `jtest.resources`

settings

An absolute or relative path to the *properties file that includes custom configuration settings.

> Some of the setting names may not be consistent with the settings dedicated for the build system plugins. See Configuration Settings for details.

- **Type**: java.io.File
- **Since**: 1.0.0
- **Required**: No

- **User property**: `jtest.settings`

settingsList

A list of settings parameters. Supported by Maven since 3.0.3. Can be configured by property since Maven 3.0.3.

- **Type**: java.io.File[]
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.settingsList`

showDetails

Displays detailed progress information.

- **Type**: java.lang.String
- **Since**: 1.0.0
- **Required**: No
- **User Property**: jtest.showdetails

showSettings

Prints the current settings and customizations along with the information where each of them is configured (e.g. in the jtest.properties file).

- **Type**: java.lang.String
- **Since**: 1.0.0
- **Required**: No
- **User property**: `jtest.showsettings`

skip

Allows you to skip the Jtest execution phase. If set to `true`, only the jtest.data.json data file is generated.

- **Type**: java.lang.String
- **Since**: 1.1.0
- **Required**: No
- **Default**: false
- **User property**: `jtest.skip`

cbt:affected-tests

Required Parameters | Optional Parameters

This goal is provided by the test impact analysis plugin shipped with Jtest and allows you to perform test impact analysis of your project (see Test Impact Analysis).

Required Parameters

Name	User Property	Accepted values / data type	Description
coverageFile	parasoft.coverage.file	Path	The path to the coverage.xml file generated by Jtest during analysis. The file contains the coverage data collected during the run. The default is: `target/jtest/coverage.xml`
testFile	parasoft.test.file	Path	The path to the report.xml file generated by Jtest during analysis. The file contains information about unit tests executed during the run. The default is: `target/jtest/report.xml`
jtestHome	jtest.home	Path	The path to the Jtest installation directory. It is not required if the Jtest installatation directory has already been configured with the Maven settings or the JTEST_HOME environment variable.

Optional Parameters

Name	User Property	Accepted values / data type	Description

runFailingTests	parasoft.runFailingTests	true | false	Enables including all failed tests form the previous run in the current test run, even if they were not affected by the changes. The default is: `false`
runModifiedTests	parasoft.runModifiedTests	true | false	Enables including new and modified tests (even if they were not affected by production code changes) in the test run. The default is: `true`
settingsFiles	jtest.settings	Path	The path to the .properties file that includes custom configuration settings. You can use a comma-separated list to provide multiple files that will be merged before execution. The settings configured in the jtest.properties file located in the Jtest installation directory are automatically included.