

Analysis Parameters

Table of Contents

- Mandatory Parameters
 - Server
 - Project Configuration
- Optional Parameters
 - Project identity
 - Authentication
 - Web Services
 - Project Configuration
 - Exclusions / Inclusions
 - Duplications
 - Analysis Logging
 - Deprecated

Parameters to configure project analysis can be set in multiple places. Here is the hierarchy of parameters:

- Global analysis parameters, defined in the UI, apply to all the projects (From the top bar, go to Administration > Configuration > General Settings)
- Project analysis parameters, defined in the UI, override global parameters (At a project level, go to Administration > General Settings)
- Project analysis parameters, defined in a project analysis configuration file or an analyzer configuration file, override the ones defined in the UI
- Analysis / Command line parameters, defined when launching an analysis, override project analysis parameters

Note that only parameters set through the UI are stored in the database.

For example, if you override the `sonar.exclusions` parameter via command line for a specific project, it will not be stored in the database. Local analyses in Eclipse, for example, would still be executed with the exclusions defined in the UI and therefore stored in the DB.

Note that the list of parameters below is not exhaustive. The property keys shown in the interface, at both global and project levels, can also be set as analysis parameters.

Mandatory Parameters

Server

Key	Description	Default value
<code>sonar.host.url</code>	Server URL	http://localhost:9000

Project Configuration

Key	Description	Default value
<code>sonar.projectKey</code>	The project key that is unique for each project. Allowed characters are: letters, numbers, '-', '_', ':' and '.', with at least one non-digit. When using Maven, it is automatically set to <code><groupId>:<artifactId></code> .	
<code>sonar.sources</code>	Comma-separated paths to directories containing source files. Compatible with Maven. If not set, the source code is retrieved from the default Maven source code location.	

Optional Parameters

Project identity

Key	Description	Default value
<code>sonar.projectName</code>	Name of the project that will be displayed on the web interface. Set through <code><name></code> when using Maven.	<ul style="list-style-type: none">If there is no name defined yet, use the project keyIf there is already a name in the DB, don't overwrite;
<code>sonar.projectVersion</code>	The project version. Set through <code><version></code> when using Maven.	"not provided"

Authentication

If [Anyone](#) does not have permission to perform analyses, you'll need to supply the credentials of a user with Execute Analysis permission for the analysis to run under.

Key	Description	Default value
<code>sonar.login</code>	The login or authentication token of a SonarQube user with Execute Analysis permission.	
<code>sonar.password</code>	The password that goes with the <code>sonar.login</code> username. This should be left blank if an authentication token is being used.	

Web Services

Key	Description	Default value
<code>sonar.ws.timeout</code>	Maximum time to wait for the response of a Web Service call (in seconds). Modifying this value from the default is useful only when you're experiencing timeouts during analysis while waiting for the server to respond to Web Service calls.	60

Project Configuration

Key	Description	Default value
<code>sonar.projectDescription</code>	The project description. Not compatible with Maven, which uses the <code><description></code> attribute.	
<code>sonar.links.homepage</code>	Project home page Not compatible with Maven, which uses the <code><url></code> attribute.	
<code>sonar.links.ci</code>	Continuous integration Not compatible with Maven, which uses the <code><ciManagement><url></code> attribute.	
<code>sonar.links.issue</code>	Issue tracker Not compatible with Maven, which uses the <code><issueManagement><url></code> attribute.	
<code>sonar.links.scm</code>	Project sources Not compatible with Maven, which uses the <code><scm><url></code> attribute.	

sonar.links.scm_dev	<p>Developer connection Not compatible with Maven, which uses the <scm><developerConnection> attribute.</p>	
sonar.tests	<p>Comma-separated paths to directories containing tests. Not compatible with Maven, which retrieves test from the default location for Java Maven projects.</p>	
sonar.sourceEncoding	<p>Set the source file encoding.</p> <p>Encoding of source files. Example of values: UTF-8, MacRoman, Shift_JIS. This property can be replaced by the standard property <code>project.build.sourceEncoding</code> in Maven projects.</p> <p>The list of available encodings depends on your JVM. See http://docs.oracle.com/javase/1.5.0/docs/guide/intl/encoding.doc.html (column : <i>Canonical Name for java.nio API</i>)</p>	System encoding
sonar.externalIssuesReportPaths	<p>Comma-delimited list of paths to Generic Issue reports.</p>	

<p><code>sonar.projectDate</code></p>	<p>Assign a date to the analysis.</p> <p>Note: This parameter is applicable to a few, special use cases, rather than being an "every day" parameter:</p> <ul style="list-style-type: none"> • When analyzing a new project, you may want to retroactively create some history for the project in order to get some information on quality trends over the last few versions. • When moving from one database engine to another, it is highly recommended (even mandatory) to start from a fresh new database schema. In doing so, you will lose the entire history for all your projects. Which is why you may want to feed the new SonarQube database with some historical data. <p>To answer those use cases, you can use the <code>sonar.projectDate</code> property. The format is <code>yy</code> <code>yy-MM-dd</code>, for example: <code>2010-12-01</code>.</p> <p>The process is the following:</p> <ul style="list-style-type: none"> • Retrieve the oldest version of your application's source that you wish to populate into the history (from a specific tag, whatever). • Run a SonarQube analysis on this project by setting the <code>sonar.projectDate</code> property. Example: <code>sonar-scanner -Dsonar.projectDate=2010-12-01</code> • Retrieve the next version of the source code of your application, update the <code>sonar.projectDate</code> property, and run another analysis. And so on for all the versions of your application you're interested in. <p>Since you cannot perform an analysis dated prior to the most recent one in the database, you must analyze your versions in chronological order, oldest first.</p> <p> Note: You may need to adjust your housekeeping settings if you wish to create a long-running history.</p>	<p>Current date</p>
<p><code>sonar.projectBaseDir</code></p>	<p>Use this property when you need analysis to take place in a directory other than the one from which it starts. E.G. analysis begins from <code>jenkins/jobs/myjob/workspace</code> but the files to be analyzed are in <code>ftpdrop/cobol/project1</code>. The path may be relative or absolute.</p> <p>Specify not the the source directory, but some parent of the source directory. The value specified here becomes the new "analysis directory", and other paths are then specified as though the analysis were starting from the new <code>sonar.projectBaseDir</code>.</p> <p>Note that the analysis process will need write permissions in this directory; it is where the <code>sonar.working.directory</code> will be created.</p>	

<code>sonar.working.directory</code>	<p>Set the working directory for an analysis triggered with the SonarQube Scanner or the SonarQube Ant Task (versions greater than 2.0). This property is not compatible with the SonarScanner for MSBuild.</p> <p>Path must be relative and unique for each project.</p> <p> Beware: the specified folder is deleted before each analysis.</p>	<code>.sonar</code>
<code>sonar.scm.provider</code>	This property can be used to explicitly tell SonarQube which SCM plugin should be used to grab SCM data on the project (in case auto-detection does not work). The value of this property is always lowercase and depends on the plugin (ex. "tfvc" for the TFVC plugin). Check the documentation page of each plugin to know more.	
<code>sonar.scm.forceReloadAll</code>	By default, blame information is only retrieved for changed files. Set this property to <code>true</code> to load blame information for all files. This can be useful if you feel that some SCM data is outdated but SonarQube does not get the latest information from the SCM engine.	<code>false</code>

Exclusions / Inclusions

See [Narrowing the Focus](#) to:

- Exclude files from analysis
- Prevent some files from being checked for duplications
- Prevent some files from being taken into account for code coverage by unit tests and integration tests
- Ignore issues on certain components and against certain coding rules

Key	Description	Default value
<code>sonar.inclusions</code>	Comma-delimited list of file path patterns to be included in analysis. When set, <i>only</i> files matching the paths set here will be included in analysis.	
<code>sonar.exclusions</code>	Comma-delimited list of file path patterns to be excluded from analysis.	
<code>sonar.coverage.exclusions</code>	Comma-delimited list of file path patterns to be excluded from coverage calculations	
<code>sonar.test.exclusions</code>	Comma-delimited list of test file path patterns to be excluded from analysis.	
<code>sonar.test.inclusions</code>	Comma-delimited list of test file path patterns to be included in analysis. When set, <i>only</i> test files matching the paths set here will be included in analysis.	
<code>sonar.issue.ignore.allfile</code>	Files containing text matching this regular expression will be ignored by analysis.	
<code>sonar.cpd.exclusions</code>	Comma-delimited list of file path patterns to be excluded from duplication detection	

Duplications

Key	Description	Default value
<code>sonar.cpd.exclusions</code>	(see the Exclusions/Inclusions section)	
<code>sonar.cpd.\${language}.minimumTokens</code>	A piece of code is considered duplicated as soon as there are at least 100 duplicated tokens in a row (override with <code>sonar.cpd.\${language}.minimumTokens</code>) spread on at least 10 lines of code (override with <code>sonar.cpd.\${language}.minimumLines</code>). For Java projects, a piece of code is considered as duplicated when there is a series of at least 10 statements in a row, regardless of the number of tokens and lines. This threshold cannot be overridden.	100
<code>sonar.cpd.\${language}.minimumLines</code>		10

Analysis Logging

Key	Description	Default value
<code>sonar.log.level</code>	Control the quantity / level of logs produced during an analysis. DEBUG: Display INFO logs + more details at DEBUG level. Similar to <code>sonar.verbose=true</code> TRACE: Display DEBUG logs + the timings of all Elasticsearch queries and Web API calls executed by the SonarQube Scanner.	INFO
<code>sonar.verbose</code>	Add more detail to both client and server-side analysis logs. <ul style="list-style-type: none"> Activates DEBUG mode for the scanner. This is a shortcut of <code>sonar.log.level=DEBUG</code>. Adds client-side environment variables and system properties to server-side log of analysis report processing. NOTE There is the potential for this setting to expose sensitive information such as passwords if they are stored as server-side environment variables.	false
<code>sonar.showProfiling</code>	Display logs to see where the analyzer spends time. This parameter is generating a file containing these timing infos in <code><workingDir>/profiling/<moduleKey>-profiler.xml</code> where <code><workingDir></code> is: <ul style="list-style-type: none"> <code>.sonar/profiling/</code> when analysis is run with Sonar Scanner <code>target/sonar/profiling/</code> when Maven is used 	false
<code>sonar.scanner.dumpToFile</code>	Outputs to the specified file the full list of properties passed to the scanner API as a means to debug analysis.	
<code>sonar.scanner.metadataFilePath</code>	Allows to specify the location where the scanner writes the <code>report-task.txt</code> file containing among others the <code>ceTaskId</code>	Its default value is the value of <code>sonar.working.directory</code>

Deprecated

DEPRECATED These parameters are listed here for completeness but are deprecated and should not be used in new analyses.

Key	Description	Default
DEPRECATED FROM SQ 6.7 sonar.branch	<p>Deprecated since SonarQube 6.7. Please consider using the Developer Edition for branch features.</p> <p>Manage SCM branches. ⚠️ Two branches of the same project are considered to be different projects in SonarQube. As a consequence issues found in a project A in a branch B1 are not linked to issues found for this project A in a branch B2. Currently, there is no way to resolve automatically issues of B2 when they are resolved in B1 as again A-B1 & A-B2 are considered as separated project.</p>	
DEPRECATED FROM SQ 4.5 sonar.language	Set the language of the source code to analyze. Browse the Plugin Library page to get the list of all available languages. If not set, a multi-language analysis will be triggered.	
DEPRECATED FROM SQ 4.5 sonar.profile	Deprecated since SQ 4.5 LTS (see SONAR-5370 - Deprecate usage of "sonar.profile" as an analysis parameter CLOSED) and should not be used .	Default profile for the given language
DEPRECATED FROM SQ 6.6 sonar.analysis.mode	This parameter is set to preview as part of PR analysis.	