

Analysis Parameters

Table of Contents

- [Mandatory Parameters](#)
 - [Server](#)
 - [Project Configuration](#)
- [Optional Parameters](#)
 - [Authentication](#)
 - [Web Services](#)
 - [Project Configuration](#)
 - [Exclusions / Inclusions](#)
 - [Analysis Logging](#)
 - [Parameters specific to Preview analyses](#)

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 Settings > General Settings)
- Project analysis parameters, defined in the UI, override global parameters (At a project level, go to Configuration > 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.projectName</code>	Name of the project that will be displayed on the web interface. Set through <code><name></code> when using Maven.	
<code>sonar.projectVersion</code>	The project version. Set through <code><version></code> when using Maven.	
<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

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
sonar.login	The login or authentication token of a SonarQube user with Execute Analysis permission.	
sonar.password	The password that goes with the sonar.login username. This should be left blank if an authentication token is being used.	

Web Services

Key	Description	Default value
sonar.ws.timeout	Maximum time to wait for the response of a Web Service call (in seconds)	60

Project Configuration

Key	Description	Default value
sonar.projectDescription	The project description. Not compatible with Maven, which uses the <description> attribute.	
sonar.links.homepage	Project home page Not compatible with Maven, which uses the <url> attribute.	
sonar.links.ci	Continuous integration Not compatible with Maven, which uses the <ciManagement><url> attribute.	
sonar.links.issue	Issue tracker Not compatible with Maven, which uses the <issueManagement><url> attribute.	
sonar.links.scm	Project sources Not compatible with Maven, which uses the <scm><url> attribute.	
sonar.links.scm_dev	Developer connection Not compatible with Maven, which uses the <scm><developerConnection> attribute.	
sonar.tests	Comma-separated paths to directories containing tests. Not compatible with Maven, which retrieves test from the default location for Java Maven projects.	
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.	
sonar.sourceEncoding	Set the source file encoding. 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. 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>)	System encoding

sonar.projectDate	<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 sonar.projectDate property. The format is <code>yyyy-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 a 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 sonar.projectDate property. Example: <code>sonar-scanner -Dsonar.projectDate=2010-12-01</code> Retrieve the next version of the source code of your application, update the sonar.projectDate 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>	Current date
sonar.branch	<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> <p>If you are a user of Developer Cockpit, please see "Limitation" section in the Developer Cockpit Installation and Usage</p>	
sonar.profile	<p>This property is deprecated since SQ 4.5 LTS (see SONAR-5370 - Deprecate usage of "sonar.profile" as an analysis parameter CLOSED) and should not be used.</p>	Default profile for the given language
sonar.projectBaseDir	<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>	
sonar.working.directory	<p>Set the working directory for an analysis triggered with the SonarQube Scanner or the SonarQube Ant Task (versions greater than 2.0).</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>
sonar.scm.provider	<p>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.</p>	
sonar.scm.forceReloadAll	<p>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 is you feel that some SCM data is outdated but SonarQube does not get the latest information from the SCM engine.</p>	<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
sonar.import_unknown_files	If set to true, all files are imported - with respect to inclusions and exclusions, even if there is no matching language plugin installed.	false

Analysis Logging

Key	Description	Default value
sonar.log.level	Control the quantity / level of logs produced during an analysis. DEBUG: Display INFO logs + more details at DEBUG level. Similar to <i>sonar.verbose=true</i> TRACE: Display DEBUG logs + all the SQL queries + their timings executed by the analyzer. Similar to <i>sonar.showProfiling=true</i>	INFO
sonar.showProfiling	<i>Deprecated since 5.1, replaced by sonar.log.level=TRACE DEBUG</i> Display logs to see where the analyzer spends time. This parameter is generating a file containing these timing infos in <workingDir>/profiling/<moduleKey>-profiler.xml where <workingDir> is: <ul style="list-style-type: none"> • .sonar/profiling/ when analysis is run with Sonar Scanner • target/sonar/profiling/ when Maven is used 	false
sonar.verbose	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 sonar.log.level=DEBUG. • 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

Parameters specific to Preview analyses

Those parameters are listed here for completeness of this documentation page - but regular users should not have to worry about them because they mainly target integration use cases.

Key	Description	Default value
sonar.analysis.mode	Set to <code>preview</code> to get quick feedback about any issues that you might have introduced before pushing your modifications to the main code base.	
sonar.report.export.path	Path to which a JSON file with all the issues found during analysis should be output. This path is relative to the working directory (see "sonar.working.directory"). By default, no file is generated.	