



Analyzing with SonarQube Scanner for Maven

Created by Olivier Gaudin, last modified by Alexandre Gigleux on Sep 06, 2017

By [SonarSource](#) – GNU LGPL 3 – [Issue Tracker](#) – [Sources](#)

SonarQube Scanner for Maven 3.3

Table of Contents

- [Features](#)
- [Compatibility](#)
- [Prerequisites](#)
- [Initial Setup](#)
 - [Global Settings](#)
- [Analyzing a Maven Project](#)
 - [Configuring the SonarQube Analysis](#)
 - [Excluding a module from SonarQube analysis](#)
 - [Sample Project](#)
 - [How to Fix Version of Maven Plugin](#)
 - [Troubleshooting](#)

Features

This analyzer is recommended to launch analysis on Java Maven project.

Compatibility

Maven Version	2.x	3.x
Compatibility		

From maven-sonar-plugin 2.7, SonarQube < 4.5 is no longer supported.

If using SonarQube instance prior to 4.5, you should use maven-sonar-plugin 2.6.

From maven-sonar-plugin 3.1, Maven < 3.0 is no longer supported.

If using Maven prior to 3.0, you should use maven-sonar-plugin 3.0.2.

Prerequisites

- Maven 3.x
- SonarQube is already [installed](#)
- At least the minimal version of Java supported by your SonarQube server is in use (Java 8 for latest LTS)
- The language plugins for each of the languages you wish to analyze are installed
- You have read [Analyzing Code Source](#).

Initial Setup

Global Settings

Edit the [settings.xml file](#), located in `$MAVEN_HOME/conf` or `~/.m2`, to set the plugin prefix and optionally the SonarQube server URL.

Example:

```
<settings>
  <pluginGroups>
    <pluginGroup>org.sonarsource.scanner.maven</pluginGroup>
  </pluginGroups>
  <profiles>
    <profile>
      <id>sonar</id>
      <activation>
        <activeByDefault>true</activeByDefault>
      </activation>
      <properties>
        <!-- Optional URL to server. Default value is http://localhost:9000 -->
        <sonar.host.url>
          http://myserver:9000
        </sonar.host.url>
      </properties>
    </profile>
  </profiles>
</settings>
```

Analyzing a Maven Project

Analyzing a Maven project consists of running a Maven goal: `sonar:sonar` in the directory where the `pom.xml` file sits.

```
mvn clean verify sonar:sonar

# In some situation you may want to run sonar:sonar goal as a dedicated step. Be sure to use install
mvn clean install
mvn sonar:sonar

# Specify the version of sonar-maven-plugin instead of using the latest. See also 'How to Fix Version'
mvn org.sonarsource.scanner.maven:sonar-maven-plugin:3.3.0.603:sonar
```

To get coverage information, you'll need to generate the coverage report before the analysis. See [Code Coverage by Unit Tests for Java Project](#) and [Code Coverage by Integration Tests for Java Project](#) for more information.

Configuring the SonarQube Analysis

Analysis parameters are listed on the [Analysis Parameters](#) page. You have to configure them in the `<properties>` section of your `pom.xml` like this:

```
<properties>
  <sonar.exclusions> [...] </sonar.exclusions>
</properties>
```

Security

Any user who's granted *Execute Analysis* permission can run an analysis.

If the *Anyone* group is not granted *Execute Analysis* permission or if the SonarQube instance is secured (the `sonar.forceAuthentication` property is set to `true`), the analysis token of a user with *Execute Analysis* permission must be provided through the `sonar.login` property. Example: `sonar-scanner -Dsonar.login=[my analysis token]`

Excluding a module from SonarQube analysis

You can either:

- define property `<sonar.skip>true</sonar.skip>` in the `pom.xml` of the module you want to exclude
- use build profiles to exclude some module (like for integration tests)
- use [Advanced Reactor Options](#) (such as `"-pl"`). For example `mvn sonar:sonar -pl !module2`

Sample Project

To help you get started, a simple project sample is available here: <https://github.com/SonarSource/sonar-scanning-examples/tree/master/sonarqube-scanner-maven>


How to Fix Version of Maven Plugin

It is recommended to [lock down versions of Maven plugins](#):

Project analyzed with Maven 3

```
<build>
  <pluginManagement>
    <plugins>
      <plugin>
        <groupId>org.sonarsource.scanner.maven</groupId>
        <artifactId>sonar-maven-plugin</artifactId>
        <version>3.3.0.603</version>
      </plugin>
    </plugins>
  </pluginManagement>
</build>
```

Troubleshooting

 If you get an java.lang.OutOfMemoryError, you can set the MAVEN_OPTS environment variable, like this in *nix environments:

```
export MAVEN_OPTS="-Xmx512m"
```

On Windows environments, avoid the double-quotes, since they get misinterpreted.

```
set MAVEN_OPTS=-Xmx512m
```