

# Analyzing with SonarQube Scanner

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

## Download SonarQube Scanner 3.3

Compatible with SonarQube 5.6+ (LTS)

[Linux 64 bit](#)

[Windows 64 bit](#)

[Mac OS X 64 bit](#)

[Any\\*](#)

\*This package expects that a JVM is already installed on the system - with same Java requirements as the SonarQube server.

## Table of Contents

- [Features](#)
- [Installation](#)
- [Use](#)
- [Troubleshooting](#)
- [Going Further](#)

## Features

The SonarQube Scanner is recommended as the default launcher to analyze a project with SonarQube.

## Installation

1. Expand the downloaded file into the directory of your choice. We'll refer to it as *<install\_directory>* in the next steps.
2. Update the global settings to point to your SonarQube server by editing *<install\_directory>/conf/sonar-scanner.properties*:

```
#----- Default SonarQube server
#sonar.host.url=http://localhost:9000
```

3. Add the *<install\_directory>/bin* directory to your path.
4. You can verify your installation by opening a new shell and executing the command `sonar-scanner -h` (on Windows platform the command is `sonar-scanner.bat -h`). You should get output like this:

```
usage: sonar-scanner [options]

Options:
  -D,--define <arg>      Define property
  -h,--help               Display help information
  -v,--version           Display version information
  -X,--debug             Produce execution debug output
```

If you need more debug information you can add the `sonar.verbose` property by adding the command line parameter `-Dsonar.verbose=true`.

## Use

Create a configuration file in the root directory of the project: `sonar-project.properties`

## sonar-project.properties

```
# must be unique in a given SonarQube instance
sonar.projectKey=my:project
# this is the name and version displayed in the SonarQube UI. Was mandatory prior to SonarQube 6.1.
sonar.projectName=My project
sonar.projectVersion=1.0

# Path is relative to the sonar-project.properties file. Replace "\" by "/" on Windows.
# This property is optional if sonar.modules is set.
sonar.sources=.

# Encoding of the source code. Default is default system encoding
#sonar.sourceEncoding=UTF-8
```

Run the following command from the project base directory to launch the analysis:

```
sonar-scanner
```

## Project Samples

To help you get started, simple project samples are available for most languages on github. They can be [browsed](#) or [downloaded](#). You'll find them filed under *projects/languages*.

## 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]`

## Troubleshooting

### Java heap space error or `java.lang.OutOfMemoryError`,

Increase the memory via the `SONAR_SCANNER_OPTS` environment variable:

```
export SONAR_SCANNER_OPTS="-Xmx512m"
```

On Windows environments, avoid the double-quotes, since they get misinterpreted and combine the two parameters into a single one.

```
set SONAR_SCANNER_OPTS=-Xmx512m
```

### Unsupported major.minor version

Upgrade the version of Java being used for analysis or use one of the native package (that embed its own Java runtime). SonarQube 5.6+ requires Java 8.

### Property missing: 'sonar.cs.analyzer.projectOutPaths'. No protobuf files will be loaded for this project.

Scanner CLI is not able to analyze .NET projects. Please, use [Scanner for MSBuild](#). If you are running Scanner for MSBuild, ensure that you are not hitting [a known limitation](#).

## Going Further

- [Advanced SonarQube Scanner Usages](#)

