Analyzing with SonarQube Scanner for MSBuild

By SonarSource – MIT – Issue Tracker – Sources

SonarScanner for MSBuild 4.6.2.2108 – Compatible with SonarQube 6.7+ (LTS)
Download for .NET Framework 4.6+
Download for .NET Core 2.0+
Download as a .NET Core Global Tool

Table of Contents

Features

The SonarScanner for MSBuild is the recommended way to launch a SonarQube or SonarCloud analysis for projects/solutions using MSBuild or dotnet command as build tool. It is the result of a collaboration between SonarSource and Microsoft.

SonarScanner for MSBuild is distributed as a standalone command line executable and as build steps for VSTS/TFS and a plugin for Jenkins.

It supports .Net Core multi-platform projects and it can be used on non-Windows platforms.

Installation and Compatibility

Usage

There are two versions of the SonarScanner for MSBuild.

The first version is based on the “classic” .NET Framework. To use it, execute the following commands from the root folder of your project:

```
SonarScanner.MSBuild.exe begin /k:"project-key"
MSBuild.exe <path to solution.sln> /t:Rebuild
SonarScanner.MSBuild.exe end
```

Note: On macOS or Linux, you can also use “mono <path to SonarScanner.MSBuild.exe>”.

The second version is based on .NET Core which has a very similar usage:

```
dotnet <path to SonarScanner.MSBuild.dll> begin /k:"project-key"
dotnet build <path to solution.sln>
dotnet <path to SonarScanner.MSBuild.dll> end
```

The .NET Core version can also be used as a .NET Core Global Tool.

```
dotnet tool install --global dotnet-sonarscanner
dotnet sonarscanner begin /k:"project-key"
dotnet build <path to solution.sln>
dotnet sonarscanner end
```

Notes:

- The .NET Core version of the scanner does not support TFS XAML builds. Apart from that, the two versions of scanner have the same capabilities and command line arguments.
- Single .NET Core project files (csproj or vbproj) could be built and successfully analyzed only if a `<ProjectGuid>unique guid</ProjectGuid>` element is added in the csproj or vbproj XML. The `<ProjectGuid>` element is not required if you build a solution (sln) containing that project.

Detailed Explanations

Begin Step

The begin step is executed when you add the `begin` command line argument. It hooks into the MSBuild pipeline, downloads SonarQube quality profiles, settings and prepares your project for the analysis.

Command Line Parameters

Parameter	Description
/k:<project-key>	[required] Specifies the key of the analyzed project in SonarQube
/n:<project name>	[optional] Specifies the name of the analyzed project in SonarQube. Adding this argument will overwrite the project name in SonarQube if it already exists.
/v:<version>	[recommended] Specifies the version of your project.
/d:<analysis-parameter>=<value>	[optional] Specifies additional SonarQube analysis parameter, you could add this argument multiple times. The most commonly used parameters are: • /d:sonar.login=<username> or <token> [optional] Specifies the username or access token to authenticate with to SonarQube. If this argument is added to the begin step, it must also be added on the end step. • /d:sonar.password=<password> - [optional] Specifies the password for the SonarQube username in the `sonar.login` argument. This argument is not needed if you use authentication token. If this argument is added to the begin step, it must also be added on the end step. • /d:sonar.verbose=true - [optional] Sets the logging verbosity to detailed. Add this argument before sending logs for troubleshooting. For detailed information about all available parameters, see Analysis Parameters

⚠️ The "begin" step will modify your build like this:

- all existing code analyzers that are referenced by your projects will be disabled and only analyzers from SonarQube plugins will be executed SQ <= 7.3
- the active CodeAnalysisRuleSet will be updated to match the SonarQube quality profile
- WarningsAsErrors will be turned off

If your build process cannot tolerate these changes we recommend creating a second build job for SonarQube analysis.

Building your Project

Between the "begin" and "end" steps, you need to build your project, execute tests and generate code coverage data. This part is specific to your needs and it is not detailed here.

End Step

The end step is executed when you add the "end" command line argument. It cleans the MSBuild hooks, collects the analysis data generated by the build, the test results, the code coverage and then uploads everything to SonarQube.

Command Line Parameters

There are only two additional arguments that are allowed for the end step:

Parameter	Description
/d:sonar.login=<username> or <token>	[optional] This argument is required if it is added to the begin step.
/d:sonar.password=<password>	[optional] This argument is required if it is added to the begin step and not required if you are using <token>

Known Limitations

- MSBuild versions older than 14 are not supported.
- Web Application projects are supported. Legacy Web Site projects are not.
- Projects targeting multiple frameworks and using preprocessor directives could have slightly inaccurate metrics (lines of code, complexity, etc.) because the metrics are calculated only from the first of the built targets.

Going Further

- Compatibility
- Install the SonarScanner for MSBuild
- Additional Analysis Parameters
- Excluding Artifacts from the Analysis
- Miscellaneous Advanced Usages

4.5.0.1761