Miscellaneous Advanced Usages

Detection of Test Projects

SonarQube analyzes test projects differently from non-test projects, so it is important to correctly classify test projects.

By default, the SonarScanner for MSBuild will detect as test project:

1. MSTest unit test projects, thanks to the presence of a well-known project type GUID in .csproj file of such projects.
2. Projects with names ending in "Test" or "Tests". This behavior can be changed by providing the parameter "`sonar.msbuild.testProjectPattern`" to the begin step (regex follows .NET Regular Expression in a case-sensitive way with the default value ".*Tests?\.(cs|vb)proj$"). This regex is applied against the fullname of the .csproj or .vbproj hence why it's recommended to keep at the end of your custom regex "\.(cs|vb)proj$".

To manually classify a project as a test project, mark it with `<SonarQubeTestProject>true</SonarQubeTestProject>`:

.csproj

```
<PropertyGroup>
  <!-- Mark the project as being a test project -->
  <SonarQubeTestProject>true</SonarQubeTestProject>
</PropertyGroup>
```

Concurrent Analyses on the Same Build Machine

Concurrent analyses (i.e. parallel analysis of two solutions on the same build machine using a unique service account) are not supported by default by the Scanner for MSBuild.

You can enable this feature following the steps described here after:

1. Locate the folder containing the Scanner for MSBuild
2. Go in the `Targets` folder and copy the folder **SonarQube.Integration.ImportBefore.targets**
3. Paste it under your build tool global **ImportBefore** folder (if the folder doesn't exist you can create it).
    a. For MSBuild, the path is ***<MSBUILD_INSTALL_DIR>\<Version>\Microsoft.Common.targets\ImportBefore*** where <MSBUILD_INSTALL_DIR> is:
        i. For v14, default path is: *C:\Program Files (x86)\MSBuild\14.0\Microsoft.Common.Targets\ImportBefore*
        ii. For v15, default path is: *C:\Program Files (x86)\Microsoft Visual Studio\2017\Community\MSBuild\15.0\Microsoft.Common.targets\ImportBefore* (for VS Community Edition)
    b. For dotnet, the path is **<DOTNET_SDK_INSTALL_DIR>\15.0*Microsoft.Common.targets\ImportBefore*** where <DOTNET_SDK_INSTALL_DIR> can be found using the "`dotnet --info` and looking for the *Base Path* property.

> **Performance impact**
> Note that the performance impact of this global installation for project not analyzed on SonarQube/SonarCloud is negligible as this target is only a bootstrapper and will bail out nearly instantaneously when the `.sonarqube` folder is not found under the folder being built.

Using SonarScanner for MSBuild with a Proxy

On build machines that connect to the Internet through a proxy server you might experience difficulties connecting to SonarCloud or your own SonarQube instance.

To instruct the Java VM to use the system proxy settings, you need to set the following environment variable before running the SonarScanner for MSBuild:

```
SONAR_SCANNER_OPTS = "-Djava.net.useSystemProxies=true"
```

To instruct the Java VM to use specific proxy settings or when there is no system-wide configuration use the following value:

```
SONAR_SCANNER_OPTS = "-Dhttp.proxyHost=yourProxyHost
-Dhttp.proxyPort=yourProxyPort"
```

Where *yourProxyHost* and *yourProxyPort* are the hostname and the port of your proxy server. There are additional proxy settings for https, authentication and exclusions that could be passed to the Java VM, for full reference visit the following article: https://docs.oracle.com/javase/8/docs/technotes/guides/net/proxies.html