

Inspections (ReSharper)

The Inspections (ReSharper) runner allows you to use the benefits of [JetBrains ReSharper code quality analysis](#) feature right in TeamCity using the bundled JetBrains ReSharper Command Line Tools. The usage of the tools within TeamCity does not require any additional licensing for ReSharper.

ReSharper analyzes your C#, VB.NET, XAML, XML, ASP.NET, ASP.NET MVC, JavaScript, HTML, CSS code and allows you to:

- Find probable bugs
- Eliminate errors and code smells
- Detect performance issues
- Improve the code structure and maintainability
- Ensure the code conforms to guidelines, standards and specifications

If you want to run ReSharper inspections using a specific ReSharper version (e.g. to ensure it matches the version you have installed in Visual Studio), you can install another version of the tools and change the default version to be used using the [Administration | Tools](#) page.

This page contains reference information about the Inspections (.Net) Build Runner fields:

- [Sources to Analyze](#)
- [Environment Requirements](#)
- [JetBrains ReSharper Command Line Tools Settings](#)
- [InspectCode Options](#)
- [Build Failure Conditions](#)
- [Build before analyze](#)

You can also refer to [ReSharper documentation](#) for more details.



To run inspections for your project, you must have a ReSharper inspection profile for .NET projects.

Sources to Analyze

Option	Description
Solution file path	The path to .sln file created by Microsoft Visual Studio 2005 or later. The specified path should be relative to the checkout directory.
Projects filter	Specify project name wildcards to analyze only a part of the solution. Leave blank to analyze the whole solution. Separate wildcards with new lines. Example: <pre>JetBrains.CommandLine.* *.Common *.Tests.*</pre>

Environment Requirements




In order to launch inspection analysis, you should have .NET Framework 4.0 (or higher) installed on an agent where builds will run.

Option	Description
Target Frameworks	This option allows you to handle the Visual Studio Multi-Targeting feature. Agent requirement will be created for every checked item. .Net Framework versions 2.0 - 4.7.2 are supported. .NET Frameworks client profiles are not supported as target frameworks.

JetBrains ReSharper Command Line Tools Settings

Option	Description
R# CLT Home Directory	Select the ReSharper Command Line Tools version. You can check the installed JetBrains ReSharper Command Line Tools versions on the Administration Tools page. If you want to run ReSharper duplicates using a specific ReSharper version (e.g. to ensure it matches the version you have installed in Visual Studio), you can use this page to install another version of the tools and can change the default version to be used.
InspectCode Platform	Select the platform bitness of the InspectCode tool. To find code issues in C++ projects, use the x86 platform.

InspectCode Options

Option	Description
Custom settings profile path	The path to the file containing ReSharper settings created with JetBrains ReSharper 6.1 or later. The specified path should be relative to the checkout directory. If specified, this settings layer has the top priority, so it overrides ReSharper build-in settings. By default, build-in ReSharper settings layers are applied. For additional information about ReSharper settings system, visit ReSharper Web Help and JetBrains .NET Tools Blog
Enable debug output	Check this option to include debug messages in the build log and publish the file with additional logs (dot net-tools-inspectcode.log) as a hidden artifact.
Additional inspectCode.exe arguments:	Specify newline-separated command line parameters to add to calling inspectCode.exe. <div style="border: 1px solid #f0e68c; padding: 5px;"> Only xml reports are supported by the runner. To get the output xml report, specify the path to the output file here via the <code>-o</code> or <code>-output</code> additional command line arguments. The paths relative to the build checkout directory as well as absolute paths are supported.</div>

Build Failure Conditions

If a build has too many inspection errors or warnings, you can configure it to fail by setting a [build failure condition](#).

Build before analyze

In order to have adequate inspections execution results, you may need to build your solution before running analysis. This pre-step is especially actual when you use (implicitly or explicitly) code generation in your project.