Get List of Zero Reference Codes in Visual Studio

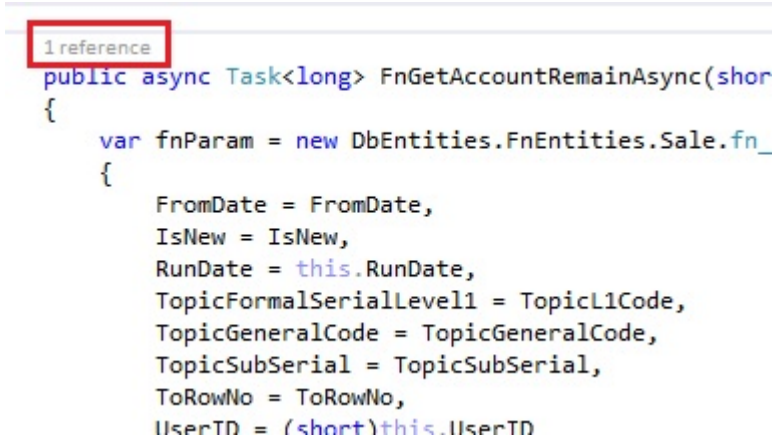
Asked 4 years, 2 months ago Active 10 months ago Viewed 50k times

109

37

In visual studio 2013 the number of references of a special Code(method, property, field,...) is shown by **Code Lens**. I want to get unused **(zero reference)** Codes in visual studio. Is there any way to get them?

I mean below reference:



c# visual-studio-2013 codelens

edited Jun 22 '15 at 7:50

asked Jun 22 '15 at 7:41

Nima Rostami

917 3 11 20

could you please explain more about it? – Nima Rostami Jun 22 '15 at 7:46

4 I think he wants a list of all methods which are not referenced, rather than to bring the number of references of that particular method to zero. – Jurgen Camilleri Jun 22 '15 at 7:46

1 yes I want to find all unused codes contain Methods, Properties, etc. – Nima Rostami Jun 22 '15 at 7:49

2 Years later, the screenshot continues to be misleading. – Sinjai Nov 1 '17 at 23:56

2 Answers

149

Probably the best and easiest way to achieve what you are after is to use the build-in code analysis tool with Visual Studio to find and take you directly to dead code and unused members.

To this effect, I simply created a new code analysis ruleset file (Via **File->New->File**, making sure **General** in the left pane was selected and scrolling down to find **Code Analysis Rule Set**, giving it a filename, then searching for and selecting the below rules). See below for the contents of the ruleset file that you can simply copy, and paste into a new file with the extension .ruleset to use.

Given a ruleset file, one can right click on a project file in the **Solution Explorer** panel, and select **Properties**. In the project properties windows, click on the **Code Analysis** tab in the left panel, and then click **Open** to browse to the .ruleset file's location. If you go to the properties of a solution file (as opposed to a project file), you can set the code analysis file for each project in the solution in one place (under **Code Analysis Settings**, and using the drop-down there to select the ruleset file. NOTE: You must have previously have browsed to the ruleset file for it to show up in the drop-down in this properties window, however).

Then you simply run the code analysis on the projects/solution (Via **Analyze->Run Code Analysis On Solution** -OR- **Alt+F11**) and it will come back as warnings, any unreferenced methods or unused members it finds. It will even find methods that are referenced by a method, whom itself has no references elsewhere.

Be careful however, as one of the ways code analysis for dead code can steer you wrong, is if the reference is 'hidden' by only ever calling the method via delegates, and of course, reflection.

The rules to detect dead code, specifically, are:

- Private methods that are not called from any other code (CA1811)
- Unused local variables (CA1804)
- Unused private fields (CA1823)
- Unused parameters (CA1801)
- Internal classes that are not instantiated from any other code (CA1812).

below XML, paste it into notepad++, save somewhere with the extension **.ruleset**, browse for and use as explained above:

```
<?xml version="1.0" encoding="utf-8"?>
<RuleSet Name="Dead Code Rules" Description=" " ToolsVersion="12.0">
  <Rules AnalyzerId="Microsoft.Analyzers.ManagedCodeAnalysis"
RuleNamespace="Microsoft.Rules.Managed">
    <Rule Id="CA1801" Action="Warning" />
    <Rule Id="CA1804" Action="Warning" />
    <Rule Id="CA1811" Action="Warning" />
    <Rule Id="CA1812" Action="Warning" />
    <Rule Id="CA1823" Action="Warning" />
  </Rules>
  <Rules AnalyzerId="Microsoft.Analyzers.NativeCodeAnalysis"
RuleNamespace="Microsoft.Rules.Native">
    <Rule Id="C6259" Action="Warning" />
  </Rules>
</RuleSet>
```

Hope this helps you, and don't forget to select the best answer.

edited Jan 2 '18 at 23:35

Brian Webster
21.3k 43 132 209

answered Jun 1 '16 at 22:21

Adam White
2,151 1 15 15

26 I don't think this completely answers the question. The main difference is CodeLens will tell you that a PUBLIC method has zero references in a entire solution. This is the key. FxCop, R#, and your method for great for anything not public. – Scott Wylie Jun 23 '16 at 22:55

@ScottWylie - I disagree. I Just tried the above solution and it did not flag un-referenced public methods. CodeLens was exceptional at flagging dead code, unreferenced local stuff and unused variables. I think this gets most people exactly what they want w/o using a 3rd party tool. – mike Nov 30 '17 at 6:05

4 @mike Consider this: If you want to find dead code with a mass operation you tend to not care about private/protected members, because these are local problems. For example, I am migrating a 500k+ LoC project with 100+ projects and 10+ solutions to a repository pattern architecture. After migrating a component, I need to know which old interfaces I can delete. Some IDEs like Eclipse have tools for exactly that. Greyed-out local methods are simply not my concern, I'd like a list of PUBLIC classes/interfaces where code lens would tell me "0". – Oliver Schimmer Apr 16 '18 at 20:57

Here's a manual way to accomplish this that I've used for finding unused Classes that are marked public.

2. Build to find all the errors
3. For each error in the build, use your source control to restore the file for the referenced class.
4. Repeat for every error until the build succeeds.
5. Any remaining files that haven't been restored are likely candidates for removal.
6. (optional) Rename the classes in the above files and do one more build to find errors.
7. Do one last Search for the name of the class you want to remove to confirm there aren't any instances of it being used in reflection or magic strings.
8. Remove the identified unused class files.
9. Repeat for each solution project you want to clean.

Note: If you don't follow the one class per file rule, this will require a lot more work. Also, any API service endpoints you will need to verify that it is not being used by any external projects.

answered Nov 19 '18 at 22:34

Ulfius
199 1 12