Using and developing Roslyn Analyzers

Fons Sonnemans

@fonssonnemans

Fons Sonnemans

- Software Development Consultant
  - Programming Languages
    - Clipper, Smalltalk, Visual Basic, C#
  - Platforms
    - Windows Forms, ASP.NET, XAML (Silverlight, WPF, Windows Phone, Windows 10, Blend)
  - Databases
    - MS SQL Server, Oracle
  - Role
    - Trainer, Coach, Advisor, Architect, Designer, Windows App Developer

- www.reflectionit.nl/training
- www.reflectionit.nl/apps

MVP Microsoft® Most Valuable Professional

Using Roslyn Analyzers, Code Fixes and Refactorings

Analyzers, Code Fixes and Refactorings

Analyzers

- Analyzers identify Errors, Warnings or Messages in your code as you type, i.e. without having to wait for a build.
  - Error: red squiggly line `Person p =`
  - Warning: green squiggly line `Main(string[] args)`
  - Message/suggestion: 3 dots `{ this.Gpa = gpa; }`

```
0 references
11  static void Main(string[] args) {
12
13      if (args.Length > 2)
14          Console.WriteLine("Yes");
15
16      Console.WriteLine("Done");
17  }
```

126 %

Error List

Entire Solution | 0 Errors | 1 Warning | 0 Messages | Build + IntelliSense

Code	Description	Project	File	Line	Suppression State	Tool
IDE0011	Add braces to 'if' statement.	ConsoleApp29	Program.cs	13	Active	Compiler

Code Fix

- Provides fixes for Analyzer Errors, Warnings or Messages
  - Sometimes provide ‘Fix all occurrences’ (batching)

```
10  static void Main(string[] args) {
11
12      if (args.Length > 2)
13
14
15
16
17  }
18
19
20
```

Add braces
Suppress IDE0011

IDE0011 Add braces to 'if' statement.

```
...
if (args.Length > 2)
if (args.Length > 2) {
    Console.WriteLine("Yes");
}
...
```

Preview changes

Fix all occurrences in: Document | Project | Solution

126 %

Error List

Entire Solution | 0 Errors | 1 Warning | 0 Messages | Build + IntelliSense

Code	Description
IDE0011	Add braces to 'if' statement.

Refactoring

- Refactoring is the process of modifying code in order to make it easier to maintain, understand, and extend, but without changing its behavior.

```
public override string ToString() {
    return string.Format("Employee Name = {0}, Salary = {1}", this.Name.ToString(), this.Salary.ToString());
```

Use expression body for methods

Convert to interpolated string

```
...
public override string ToString() {
    return string.Format("Employee Name = {0}, Salary = {1}", this.Name.ToString(), this.Salary.ToString());
    return $"Employee Name = {this.Name.ToString()}, Salary = {this.Salary.ToString()}";
}
...
```

Preview changes

C# Code Styles & Naming – Generate .editorconfig file

Options

Search Options (Ctrl+E)

- Text Editor
  - General
  - Advanced
  - File Extension
  - All Languages
  - Android XML
  - Basic
  - C#
    - General
    - Scroll Bars
    - Tabs
    - Advanced
    - Code Style
      - General
      - Formatting
      - Naming
    - IntelliSense
  - C#_LSP
  - C#_LSP
  - C/C++
  - CSS
  - Dockerfile
  - EditorConfig
  - F#
  - HTML
  - HTML (Web Forms)
  - JavaScript/TypeScript
  - JSON

Your .editorconfig file might override the local settings configured on this page which only apply to your machine. To configure these settings to travel with your solution use EditorConfig files. More info Learn more

Generate .editorconfig file from settings

Description	Preference	Severity
'var' preferences:		
For built-in types	Prefer 'var'	Suggestion
When variable type is apparent	Prefer 'var'	Suggestion
Elsewhere	Prefer 'var'	Suggestion
Code block preferences:		
Prefer braces	Yes	Warning
Prefer auto properties	Yes	Warning
Prefer simple 'using' statement	Yes	Suggestion

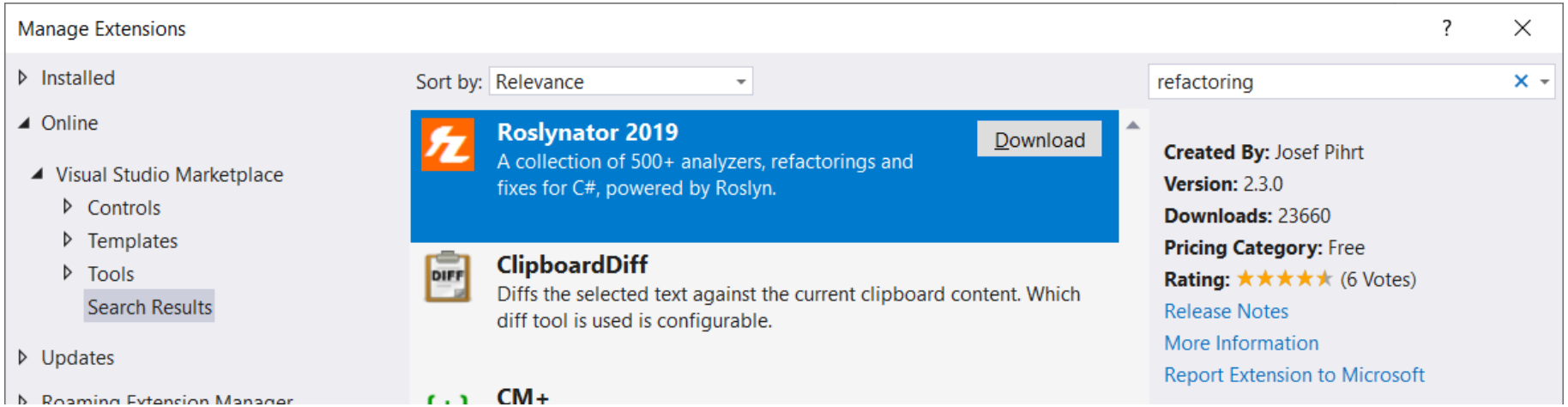
```
// Prefer:
if (test) {
    Console.WriteLine("Text");
}

// Over:
if (test)
    Console.WriteLine("Text");
```

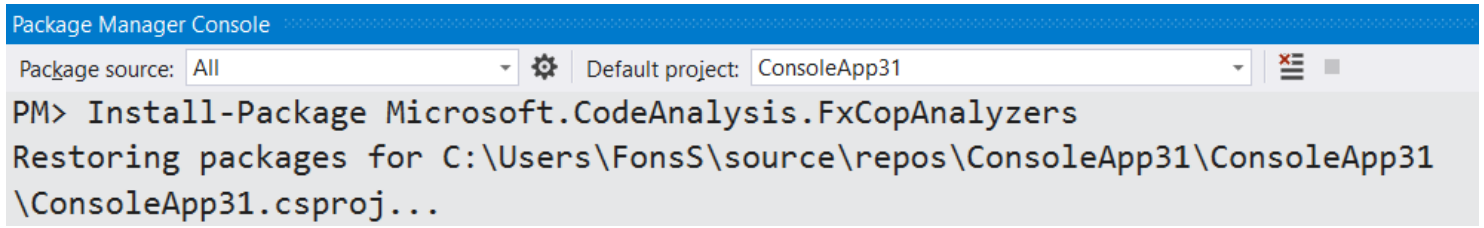
OK Cancel

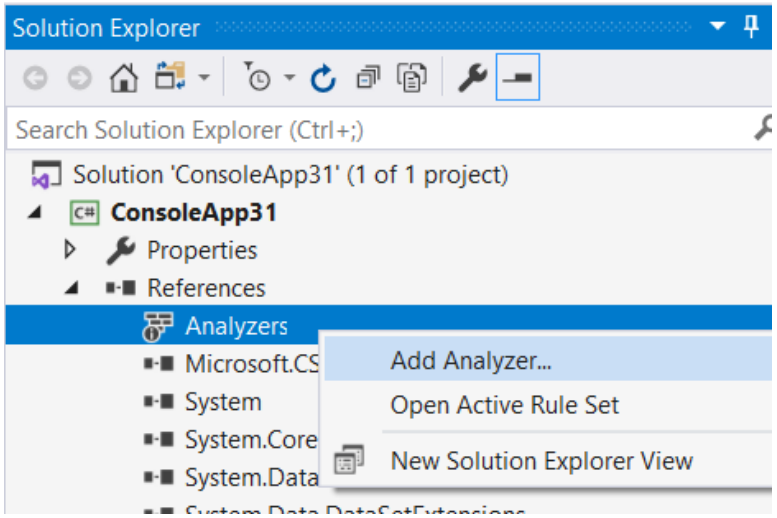
Deployment

- Visual Studio Extension (VSIX)

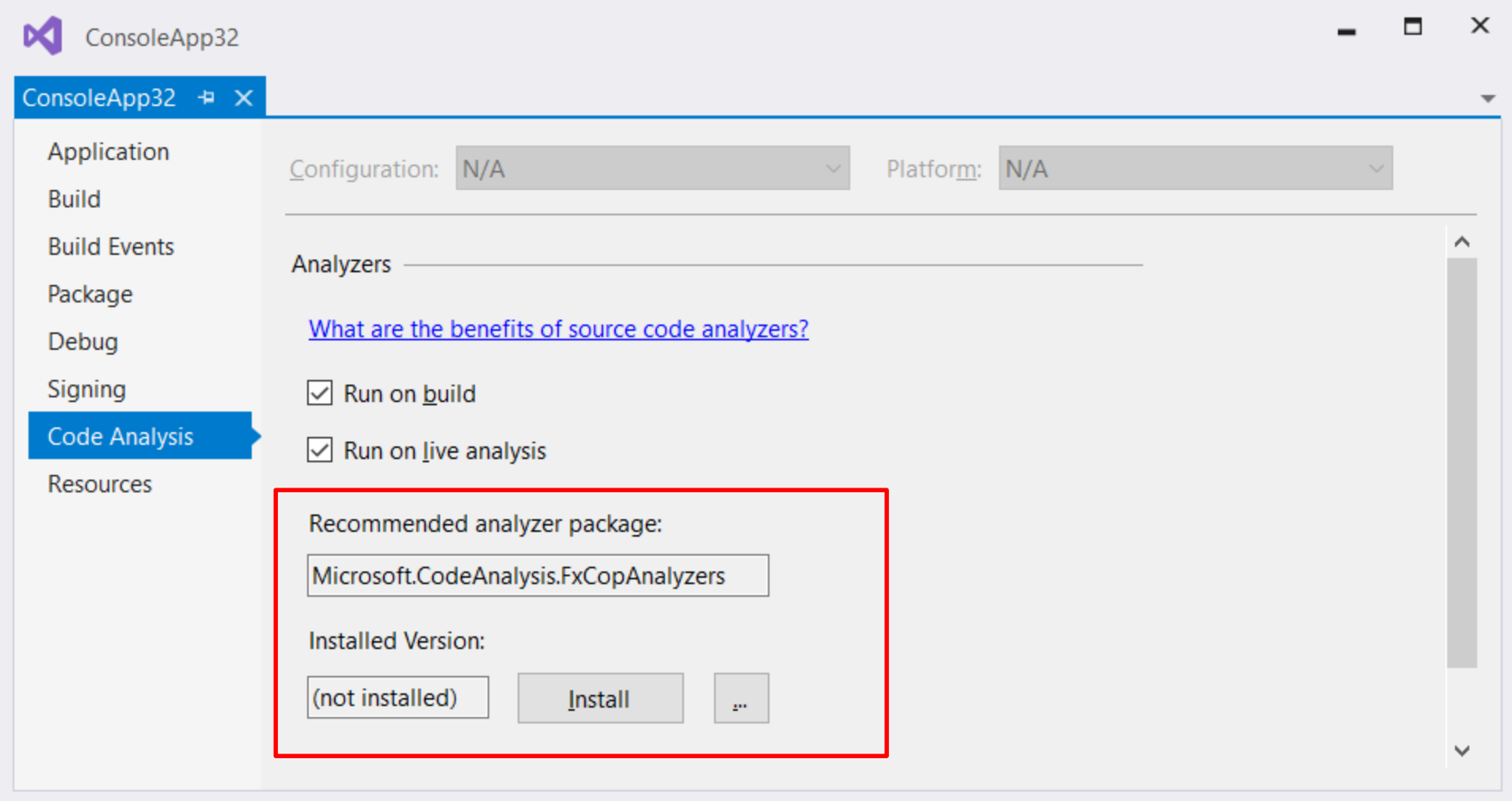


- Project Reference (NuGet Package or Assembly)





VS2019 – Code Analysis



Microsoft.CodeAnalysis.FxCopAnalyzers

ConsoleApp72 - Microsoft Visual Studio Preview

PREVIEW

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Debug | Any CPU | Start

Program.cs

ConsoleApp72 | ConsoleApp72.Program | Main(string[] args)

```
7   namespace ConsoleApp72 {
        0 references
8       class Program {
            0 references
9           static void Main(string[] args) {
10          }
11      }
12
        0 references
13      public class Test {
14
15          public delegate void MyNestedDelegate();
16      }
17  }
18
```

120 %

Error List

Entire Solution | 0 Errors | 3 Warnings | 0 Messages | Build + IntelliSense | Search Error List

Code	Description	Project
CA1801	Parameter args of method Main is never used. Remove the parameter or use it in the method body.	ConsoleApp72
CA1052	Type 'Test' is a static holder type but is neither static nor NotInheritable	ConsoleApp72
CA1034	Do not nest type MyNestedDelegate. Alternatively, change its accessibility so that it is not externally visible.	ConsoleApp72

Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'ConsoleApp72' (1 project)
  - ConsoleApp72
    - Properties
    - References
      - Analyzers
        - Microsoft.CodeQuality.Analyzers
        - Microsoft.CodeQuality.CSharp.Analyzers
        - Microsoft.NetCore.Analyzers
        - Microsoft.NetCore.CSharp.Analyzers
        - Microsoft.NetFramework.Analyzers
        - Microsoft.NetFramework.CSharp.Analyzers
        - Text.CSharp.Analyzers
        - Microsoft.CodeAnalysis.FxCopAnalyzers
      - Microsoft.CSharp
      - System
      - System.Core
      - System.Data
      - System.Data.DataSetExtensions
      - System.Net.Http
      - System.Xml
      - System.Xml.Linq
    - App.config

Solution Explorer | Team Explorer

Package Manager Console

Ready | Add to Source Control

Configure or **Suppress** issues

```
3   namespace ConsoleApp32 {
        0 references
4       class Program {
5
6           private static int _data = 12;
7
8       id Main(string[] args) {
9       le.WriteLine(_data);
10
11
12
```

- Add readonly modifier
- Introduce constant
- Encapsulate field: '_data' (and use property)
- Encapsulate field: '_data' (but still use field)
- Convert to binary
- Convert to hex
- Configure or Suppress issues
  - Configure IDE0044 code style
  - Configure IDE0044 severity
  - Suppress IDE0044
    - in Source
    - in Suppression File
    - in Source (attribute)

153 % | 0 | 1 | Ln: 6 | Ch: 31 | SPC | CRLF

Error List

Entire Solution | 0 Errors | 1 Warning | 1 Message | Build + Intel

Code	Description
IDE0044	Make field readonly
IDE0060	Remove unused parameter 'args'

IDE0044 Make field readonly

```
...
#pragma warning disable IDE0044 // Add readonly modifier
        private static int _data = 12;
#pragma warning restore IDE0044 // Add readonly modifier
...
```

Preview changes

Fix all occurrences in: Document | Project | Solution

Configure code style

- Stored in **.editorconfig** file

```
class Program {

    private static int _data = 12;

id Main(string[] args) {
le.WriteLine(_data);
```

Add readonly modifier
Introduce constant
Encapsulate field: '_data' (and use property)
Encapsulate field: '_data' (but still use field)
Convert to binary
Convert to hex
Configure or Suppress issues
Configure IDE0044 code style
Configure IDE0044 severity
Suppress IDE0044
true
false

IDE0044 Make field readonly

```
...
# Field preferences
dotnet_style_readonly_field = true:warning
dotnet_style_readonly_field = false:warning

...
```

Preview changes

153 %
0
1

Error List
Entire Solution
0 Errors
1 Warning
1 Message
Build + IntelliSense

Code	Description
IDE0044	Make field readonly
IDE0060	Remove unused parameter 'args'

Visual Studio 2017 Rulesets

- **Deprecated** in VS2019, **replaced** by **.editorconfig**

ConsoleApp328 - Microsoft Visual Studio

ConsoleApp328.ruleset* | Program.cs

Group by: Analyzer ID

ID	Name	Action
CodeCracker.CSharp		Multiple
CC0001	You should use 'var' whenever possible.	Warning
CC0002	Invalid argument name	Warning
CC0003	Your catch should include an Exception	Warning
CC0004	Catch block cannot be empty	Warning
CC0005	Empty Object Initializer	Warning
CC0006	Use foreach	Warning
CC0007	Return Condition directly	Warning
CC0008	Use object initializer	Warning
CC0009	Use object initializer	Warning
CC0010	Your regex expression is incorrect	Error
CC0011	You should remove the 'Where' invocation when it is possible.	Warning
CC0012	Your throw does nothing	Warning
CC0013	Use ternary operator	Info
CC0014	Use ternary operator	Info
CC0015	Unnecessary Parenthesis	Warning
CC0017	Use auto property	Info
CC0018	Use the existence operator	Info
CC0019	Use 'switch'	Info
CC0020	You should remove the lambda expression when it only invo...	Hidden
CC0021	Use nameof	Warning
CC0022	Should dispose object	None
CC0023	Unsealed Attribute	Warning
CC0024	Don't throw exceptions inside static constructors.	Warning
CC0025	Remove Empty Finalizers	Warning
CC0026	Call Extension Method As Extension	Info
CC0029	Disposables Should Call Suppress Finalize	Warning
CC0030	Make Local Variable Constant.	Info
CC0031	Check for null before calling a delegate	Warning

Solution Explorer: Solution 'ConsoleApp328' (1 project) — ConsoleApp328 — Properties — References

ConsoleApp328* | ConsoleApp328.ruleset | Program.cs

Configuration: Active (Debug) Platform: Active (Any CPU)

Enable Code Analysis on Build

Suppress results from generated code (managed only)

Rule Set

Run this rule set: My Custom Ruleset

Description:
These rules focus on the most critical problems in your code, including potential security holes, application crashes, and other important logic and design errors. It is recommended to include this rule set in any custom rule set you create for your projects.

Path:
C:\Users\FonsS\source\repos\ConsoleApp328\ConsoleApp328\ConsoleApp328.ruleset

Open | Learn more about rule sets

Solution Explorer: Solution 'ConsoleApp328' (1 project) — ConsoleApp328 — Properties — References — App.config — ConsoleApp328.ruleset — Program.cs

StyleCop.Analyzers

ConsoleApplication20 - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Architecture Test Analyze Window Help

Debug | Any CPU | Start

Object Browser | Program.cs

ConsoleApplication20 | ConsoleApplication20.Program | Main(string[] args)

```
1  using System;
2  using System.Collections.Generic;
3  using System.IO;
4  using System.Linq;
5  using System.Text;
6  using System.Threading.Tasks;
7
8  namespace ConsoleApplication20 {
       0 references
9      class Program {
10
11
12
13         }
14     }
15
16 }
```

Declare accessibility

SA1400 Element 'Program' must declare an access modifier

```
...
namespace ConsoleApplication20 {
    class Program {
    internal class Program {
...
```

Preview changes

Fix all occurrences in: Document | Project | Solution

119 %

Error List - Current Project (ConsoleApplication20)

Current Project | 0 Errors | 25 Warnings | 0 Messages | Build + IntelliSense | Search Error List

Code	Description	Project	File
SA1200	Using directive must appear within a namespace declaration	ConsoleApplication20	Program
SA1200	Using directive must appear within a namespace declaration	ConsoleApplication20	Program
SA1200	Using directive must appear within a namespace declaration	ConsoleApplication20	Program
SA1500	Curly brackets for multi-line statements must not share line	ConsoleApplication20	Program
SA1400	Element 'Program' must declare an access modifier	ConsoleApplication20	Program
SA1500	Curly brackets for multi-line statements must not share line	ConsoleApplication20	Program
SA1505	An opening curly bracket must not be followed by a blank line.	ConsoleApplication20	Program

Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'ConsoleApplication20' (1 project)
  - ConsoleApplication20
    - Properties
    - References
      - Analyzers
        - Newtonsoft.Json
        - StyleCop.Analyzers
      - Microsoft.CSharp
      - System
      - System.Core
      - System.Data
      - System.Data.DataSetExtensions
      - System.Net.Http
      - System.Xml
      - System.Xml.Linq
    - App.config
    - packages.config
    - Program.cs

Solution Explorer | Team Explorer

Properties

- https://blog.submain.com/stylecop-detailed-guide/

WpfAnalyzers

- https://github.com/DotNetAnalyzers/WpfAnalyzers

DotNetAnalyzers/WpfA

GitHub, Inc. [US] https://github.com/DotNetAnalyzers/WpfAnalyzers

- 2.x versions are for C#7.

WPF0001	Backing field for a DependencyP
WPF0002	Backing field for a DependencyP
WPF0003	CLR property for a DependencyP
WPF0004	CLR method for a DependencyP
WPF0005	Name of PropertyChangedCallb
WPF0006	Name of CoerceValueCallback sh
WPF0007	Name of ValidateValueCallback s
WPF0010	Default value type must match r
WPF0011	Containing type should be used
WPF0012	CLR property type should match
WPF0013	CLR accessor for attached prope
WPF0014	SetValue must use registered typ
WPF0015	Registered owner type must inh

```
0 references
public sealed class MoneyConverter : IValueConverter {
```

Add [ValueConversion(typeof(object), typeof(string))].
Add default field.
Add default field with docs.
Add default property.
Add default property with docs.
Generate constructor 'MoneyConverter()'
Suppress WPF0070
Suppress WPF0071

WPF0070 Add default field to converter.

```
public sealed class MoneyConverter : IValueConverter {
    /// <summary> Gets the default instance </summary>
    public static readonly MoneyConverter Default = new MoneyConverter();
```

Preview changes
Fix all occurrences in: Document

```
rtBack(object value, Type targetType, object p
arse(value.ToString(), NumberStyles.Currency,
        return money;
    } else {
        return DependencyProperty.UnsetValue;
```

ror List

Entire Solution | 0 Errors | 0 Warnings | 3 Messages | Build + IntelliSense

Code	Description	Project	File
IDE0009	Add 'this' or 'Me' qualification.	WpfApp2	MainWindow.xaml
WPF0070	Add default field to converter.	WpfApp2	Class1.cs
WPF0071	Add ValueConversion attribute.	WpfApp2	Class1.cs

ClrHeapAllocationAnalyzer – Boxing & Unboxing

ConfigureAawait - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Architecture Test Web Essentials Analyze Window Help

Debug | Any CPU | Start

MainWindow.xaml.cs | Employee.cs* | NuGet: ConfigureAwait

ConfigureAwait | AsyncWpfDemo.Employee | ToString()

```
using System;

namespace AsyncWpfDemo {

    4 references
    class Employee {

        1 reference
        public int ID { get; set; }
        1 reference
        public string Name { get; set; }
        1 reference
        public decimal Salary { get; set; }
        2
        0 references
        public override string ToString() {
            return string.Format("{0} - {1} - {2:C0}", this.ID, this.Name, this.Salary);
            //return string.Format("{0} - {1} - {2}",
            //                     this.ID.ToString(), this.Name, this.Salary.ToString("C0"));
        }
    }

}
```

87 %

Solution Explorer

Search Solution Explorer (Ctrl+;)

- References
  - Analyzers
    - ClrHeapAllocationAnalyzer
      - HeapAnalyzerBoxingRule: Valu
      - HeapAnalyzerClosureCapture
      - HeapAnalyzerClosureSource
      - HeapAnalyzerDelegateOnStru
      - HeapAnalyzerEnumeratorAllo
      - HeapAnalyzerExplicitNewAno
      - HeapAnalyzerExplicitNewArra

Solution Explorer | Team Explorer

Properties

Error List

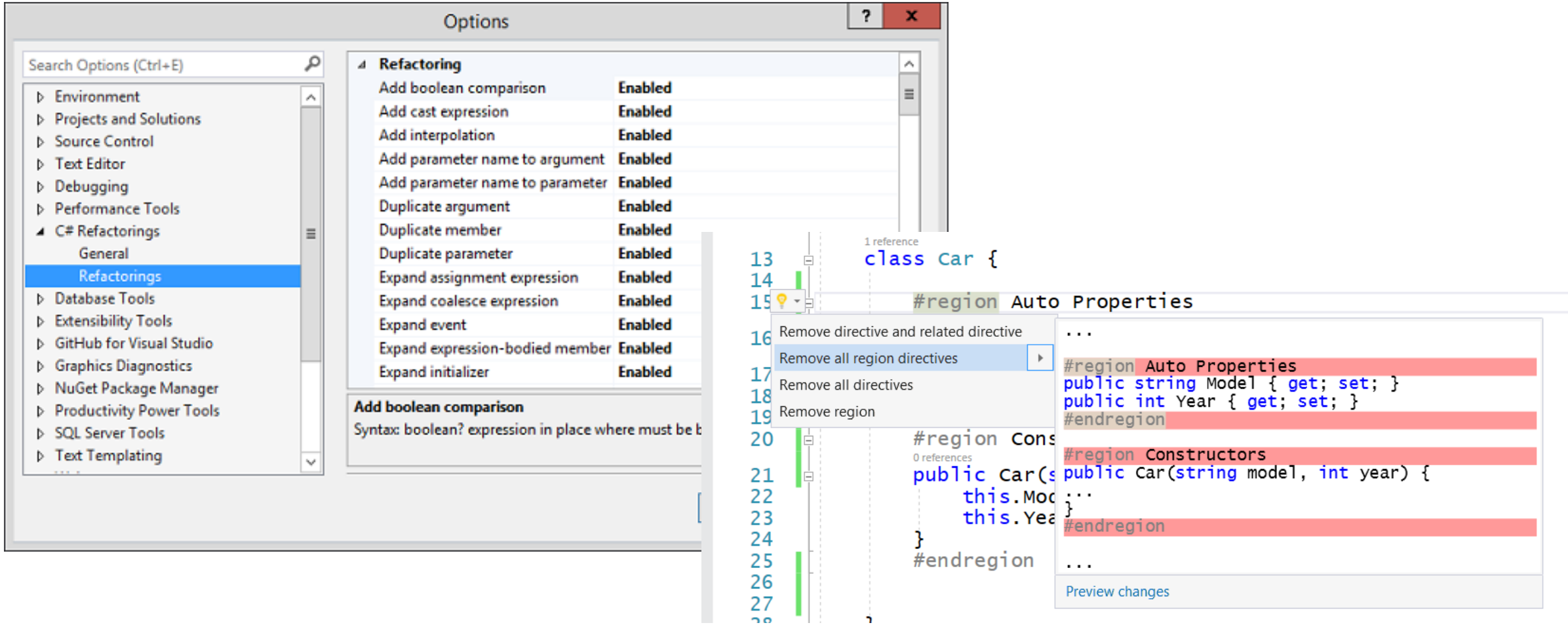
Entire Solution | 0 Errors | 19 Warnings | 0 of 3 Messages | Build + IntelliSense | Search Error List

Code	Description	File	Project	Line	S..
HeapAnalyzerBoxingRu	Value type to reference type conversion causes boxing at call site (here), and unboxing at the callee-site. Consider using generics if applicable	Employee.cs	ConfigureAwait	12	Active
HeapAnalyzerBoxingRu	Value type to reference type conversion causes boxing at call site (here), and unboxing at the callee-site. Consider using generics if applicable	Employee.cs	ConfigureAwait	12	Active
RIT0001	Private field 'Url' does not start with an underscore and camelCasing	MainWindow.xaml.cs	ConfigureAwait	29	Active

Ready | Ln 18 | Col 2 | Ch 2 | INS | Publish

Roslynator

- A collection of 500+ analyzers, refactorings and fixes for C#, powered by Roslyn.
  - https://github.com/JosefPihrt/Roslynator



ReflectionIT.Analyzer ☺

https://github.com/sonnemaf/ReflectionIT.Analyzer/tree/master/ReflectionI GitHub, Inc. [US] ReflectionIT.Analyzer/Reflecti...

ReflectionIT.Analyzer / ReflectionIT.Analyzer / ReflectionIT.Analyzer / **Analyzers** /

Fons Sonnemans local variable camelCase naming fix + ExplicitTypecast fix — Latest commit `82b0028` on Mar 17

..		
AsyncMethodNameSuffix	Add project files.	2 months ago
ExplicitTypecast	local variable camelCase naming fix + ExplicitTypecast fix	a month ago
ForClosures	Add project files.	2 months ago
LocalVariable	local variable camelCase naming fix + ExplicitTypecast fix	a month ago
MissingModifiers	Add project files.	2 months ago
NonPrivateField	Add project files.	2 months ago
PrivateField	local variable camelCase naming fix + ExplicitTypecast fix	a month ago
DependencyProperty.cs	Add project files.	2 months ago
DiagnosticAnalyzerCategories.cs	Add project files.	2 months ago

Refactorings & Code Analyzers Links

- **https://github.com/dotnet/roslyn-analyzers (FxCop)**
- **https://github.com/DotNetAnalyzers/StyleCopAnalyzers**
- **https://github.com/code-cracker/code-cracker**
- **https://github.com/JosefPihrt/Roslynator**
- **https://github.com/mjsabby/RoslynClrHeapAllocationAnalyzer**
- **https://github.com/sonnemaf/ReflectionIT.Analyzer**
- **https://github.com/DotNetAnalyzers/WpfAnalyzers**
- **https://github.com/dotnet-security-guard/roslyn-security-guard**

- https://github.com/Vannevelj/VSDiagnostics
- https://github.com/DustinCampbell/CSharpEssentials
- https://dotnet-security-guard.github.io/index.htm
- https://github.com/SergeyTeplyakov/ExceptionAnalyzer
- https://github.com/Wintellect/Wintellect.Analyzers
- https://github.com/olohmann/AsyncAwaitAnalyzer
- http://vsrefactoringessentials.com/
- https://github.com/meziantou/Meziantou.Analyzer

Developing Roslyn Analyzers, Code Fixes and Refactorings

Visual Studio Installer (2017 + 2019)

Modifying — Visual Studio Professional 2017 — 15.8.8

Workloads | Individual components | Language packs | Installation locations

Mobile development with JavaScript
Build Android, iOS and UWP apps using Tools for Apache Cordova.

Mobile development with C++
Build cross-platform applications for iOS, Android or Windows using C++.

Game development with C++
Use the full power of C++ to build professional games powered by DirectX, Unreal, or Cocos2d.

Other Toolsets (3)

Visual Studio extension development
Create add-ons and extensions for Visual Studio, including new commands, code analyzers and tool windows.

Linux development with C++
Create and debug applications running in a Linux environment.

.NET Core cross-platform development
Build cross-platform applications using .NET Core, ASP.NET Core, HTML/JavaScript, and Containers including Docker...

Installation details

- .NET Core cross-platform development
- Visual Studio extension development *

Included
- Visual Studio SDK
- Visual Studio extension development prerequisi...

Optional
- .NET profiling tools
- Text Template Transformation
- PreEmptive Protection - Dotfuscator
- Visual C++ runtime for UWP
- Developer Analytics tools
- Class Designer
- Visual C++ ATL for x86 and x64
- Visual C++ MFC for x86 and x64
- Visual Studio C++ core features
- VC++ 2017 version 15.8 v14.15 latest v141 tools
- .NET Compiler Platform SDK

Individual components *
- Static analysis tools
- .NET Framework 4.7.1 SDK

Location
C:\Program Files (x86)\Microsoft Visual Studio\2017\Professional Change...

Total space required 224 KB

By continuing, you agree to the license for the Visual Studio edition you selected. We also offer the ability to download other software with Visual Studio. This software is licensed separately, as set out in the 3rd Party Notices or in its accompanying license. By continuing, you also agree to those licenses.

Install while downloading | Modify

Install | More

2.0.55.1005

Extra Extensibility Templates - VS2017

New Project

Android
Apple TV
Apple Watch
Cloud
Cross-Platform
Extensibility
iOS Extensions
iPhone & iPad
Test
WCF
Other Languages

Not finding what you are looking for?
Open Visual Studio Installer

Sort by: Default

Search (Ctrl+E)

Stand-Alone Code Analysis Tool	Visual C#
Analyzer with Code Fix (.NET Standard)	Visual C#
C# Item Template	Visual C#
C# Project Template	Visual C#
Code Refactoring (.NET Standard)	Visu
VSIX Project	Visu

Type: Visual C#

Create a C# analyzer that comes with a code fix and generates diagnostics. The analyzer can be deployed as either a NuGet package or a VSIX extension.

Name: Analyzer1
Location: C:\Users\Fons Sonnemans\Source\Repos
Solution name: Analyzer1
Framework: .NET Framework 4.7.2

New Project

Android
Apple TV
Apple Watch
Cloud
Cross-Platform
Extensibility
iOS Extensions
iPhone & iPad
Test
WCF
Other Languages

Not finding what you are looking for?
Open Visual Studio Installer

Sort by: Default

Search (Ctrl+E)

Stand-Alone Code Analysis Tool	Visual C#
Analyzer with Code Fix (.NET Standard)	Visual C#
C# Item Template	Visual C#
C# Project Template	Visual C#
Code Refactoring (.NET Standard)	Visual C#
VSIX Project	Visual C#

Type: Visual C#

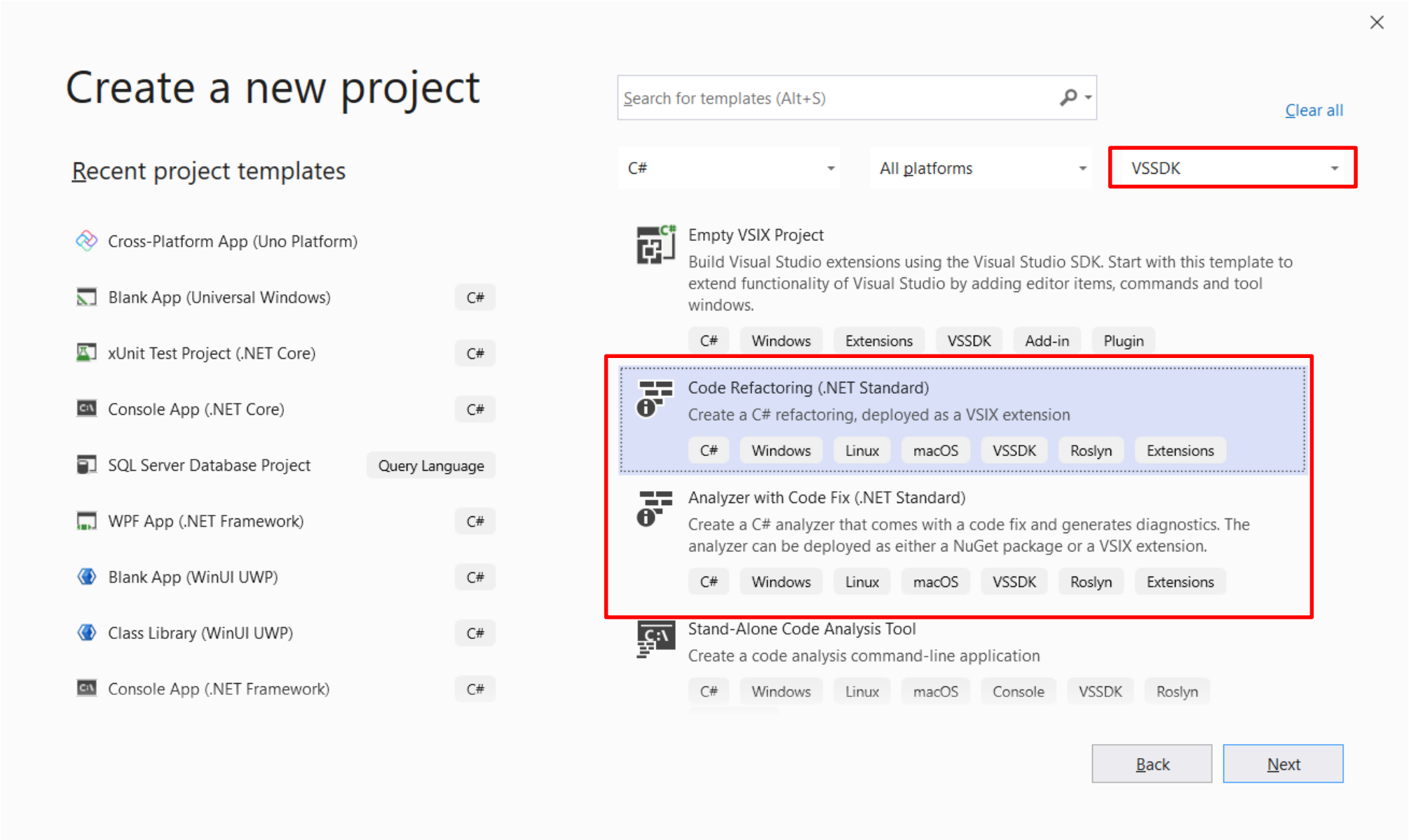
Create a C# refactoring, deployed as a VSIX extension

Name: CodeRefactoring1
Location: C:\Users\Fons Sonnemans\Source\Repos
Solution name: CodeRefactoring1
Framework: .NET Framework 4.7.2

Browse...
☑ Create directory for solution
☐ Add to Source Control

OK Cancel

Extra Extensibility Templates - VS2019



Syntax Visualizer

ConsoleApplication8 - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Architecture Test Analyze Window Help

Debug | Any CPU | Start

Syntax Visualizer

Syntax Tree | Legend

- OpenBraceToken [217..218)
- LocalDeclarationStatement [234..244)
- IfStatement [260..326)
  - IfKeyword [260..262)
  - OpenParenToken [263..264)
  - GreaterThanExpression [264..269)
  - CloseParenToken [269..270)
  - Block [271..326)
- IfStatement [342..374)
  - IfKeyword [342..344)
  - OpenParenToken [345..346)
  - GreaterThanExpression [346..351)
  - CloseParenToken [351..352)
  - ExpressionStatement [353..374)
- CloseBraceToken [386..387)
- CloseBraceToken [393..394)
- CloseBraceToken [396..397)

Properties

Type	IfStatementSyntax
Kind	IfStatement
ContainsDiagnostics	False
ContainsDirectives	False

Program.cs — ConsoleApplication8 — ConsoleApplication8.Program — Main(string[] args)

```
1   using System;
2   using System.Collections.Generic;
3   using System.Linq;
4   using System.Text;
5   using System.Threading.Tasks;
6
7   namespace ConsoleApplication8 {
        0 references
8       class Program {
            0 references
9           static void Main(string[] args) {
10
11              var a = 5;
12
13              if (a > 2) {
14                  Console.WriteLine(a);
15              }
16
17              if (a > 2) Console.WriteLine(a);
18
19          }
20      }
21  }
22
```

119 %

Solution Explorer

Search Solution Explor

- Solution 'ConsoleApplic
- ConsoleApplication8
  - Properties
  - References
  - App.config
  - Program.cs

Solution Ex... | Team Explo...

Properties

Ready

Syntax and Semantic APIs

Syntax Tree

High-fidelity representation of source code

Semantic Model

Answers semantic questions about syntax.

- Can require a compilation
- Can be slower and use more memory

IOperation

In-progress evolution of Roslyn APIs

Provides semantic information

Abstracts over common syntactic shapes

Common representation for C#/VB

Syntax Generator

Can generate source code for both C# and VB

Creating Refactoring

Demo

Create ‘Code Refactoring’ project

Create a new project

Search for templates (Alt+S)

Clear all

Recent project templates

- Code Refactoring (.NET Standard) — C#
- Console App (.NET Core) — C#
- Blank App (Universal Windows) — C#
- xUnit Test Project (.NET Core) — C#
- SQL Server Database Project — Query Language
- WPF App (.NET Framework) — C#
- Blank App (WinUI UWP) — C#
- Class Library (WinUI UWP) — C#
- Console App (.NET Framework) — C#

C# | All platforms | VSSDK

VSIX Project
Build Visual Studio extensions using the Visual Studio SDK. Start with this template to create an extension project with an async package. Further extend functionality of Visual Studio by adding editor items, commands and tool windows.
C# Windows Extensions VSSDK Add-in Plugin

Empty VSIX Project
Build Visual Studio extensions using the Visual Studio SDK. Start with this template to extend functionality of Visual Studio by adding editor items, commands and tool windows.
C# Windows Extensions VSSDK Add-in Plugin

Code Refactoring (.NET Standard)
Create a C# refactoring, deployed as a VSIX extension
C# Windows Linux macOS VSSDK Roslyn Extensions

Analyzer with Code Fix (.NET Standard)
Create a C# analyzer that comes with a code fix and generates diagnostics. The analyzer can be deployed as either a NuGet package or a VSIX extension.

Next

Update NuGet Packages & Framework: .NET Standard 2.0

CodeRefactoring2 - NuGet: CodeRefactoring2

NuGet: CodeRefactoring2

Browse | Installed | Updates 2

NuGet Package Manager: CodeRefactoring2

Search (Ctrl+L) | Include prerelease | Package source: nuget.org

Select all packages | Update

Microsoft.CodeAnalysis.Analyzers by Microsoft — v2.6.2 / v2.9.8

Analyzers for consumers of Microsoft.CodeAnalysis NuGet package, i.e. extensions and applications built on top of .NET Compiler Platform (Roslyn). Thi...

Microsoft.CodeAnalysis.CSharp.Workspaces by Microsoft — v2.10.0 / v3.4.0

.NET Compiler Platform ("Roslyn") support for analyzing C# projects and solutions.

Microsoft.CodeAnaly nuget.org

Installed: 2.6.2 | Uninstall

Version: Latest stable 2.9.8 | Update

Options

Description

Analyzers for consumers of Microsoft.CodeAnalysis NuGet package, i.e. extensions and applications built on top of .NET Compiler Platform (Roslyn). This package is included as a development dependency of Microsoft.CodeAnalysis NuGet package and does not need to be installed separately if you are referencing Microsoft.CodeAnalysis NuGet package.

Version: 2.9.8

Author(s): Microsoft

“Reverse type name” Refactoring - F5

```
10  using Microsoft.CodeAnalysis.CSharp.Syntax;
11  using Microsoft.CodeAnalysis.Rename;
12  using Microsoft.CodeAnalysis.Text;
13
14  namespace CodeRefactoring2 {
15      [ExportCodeRefactoringProvider(LanguageNames.CSharp, Name = nameof(CodeRefactoring2CodeRefactoringProvider))]
        1 reference
16      internal class CodeRefactoring2CodeRefactoringProvider : CodeRefactoringProvider {
            0 references
17          public sealed override async Task ComputeRefactoringsAsync(CodeRefactoringContext context) {
18              // TODO: Replace the following code with your own analysis, generating a CodeAction for each refacto
19
20              var root = await context.Document.GetSyntaxRootAsync(context.CancellationToken).ConfigureAwait(false
21
22              // Find the node at the selection.
23              var node = root.FindNode(context.Span);
24
25              // Only offer a refactoring if the selected node is a type declaratic
26              if (!(node is TypeDeclarationSyntax typeDecl)) {
27                  return;
28              }
29
30              // For any type declaration node, create a code action to reverse the identifier text.
31              var action = CodeAction.Create("Reverse type name", c => ReverseTypeNameAsync(context.Document, typel
32
33              // Register this code action.
34              context.RegisterRefactoring(action);
35          }
36
            1 reference
37          private async Task<Solution> ReverseTypeNameAsync(Document document, TypeDeclarationSyntax typeDecl, Can
38              // Produce a reversed version of the type declaration's identifier token.
39              var identifierToken = typeDecl.Identifier;
40              var newName = new string(identifierToken.Text.ToCharArray().Reverse().ToArray());
41
42              // Get the symbol representing the type to be renamed.
```

(awaitable) System.Runtime.CompilerServices.ConfiguredTaskAwaitable<SyntaxNode> Task<SyntaxNode>.ConfigureAwait(bool continueOnCapturedContext) (+ 1 overload)
Configures an awaiter System await this Task<TResult>.
Usage:
SyntaxNode x = await ConfigureAwait(...);

“Reverse type name” Refactoring - F5

File Edit View Project Build Debug Architecture Test Analyze Tools Extensions Window Help Search (C

Debug | Any CPU | ConsoleApp34

Toolbox

Program.cs*

ConsoleApp34 | ConsoleApp34.Program

```
1   using System;
2
3   namespace ConsoleApp34 {
4
        0 references
5       class Program {
```

- Add modifier
- Add accessibility modifiers
- Sort Members
- Reverse type name
- Move to namespace...
- Generate overrides...
- Generate constructor 'Program()'
- Suppress or Configure issues

```
...

class Program {
class margorP {
    static void Main(string[] args) {
...
```

Preview changes

Creating Analyzers and/or Code Fix

Create ‘Analyzer with Code Fix’ Project

Create a new project

Search for templates (Alt+S)

Clear all

Recent project templates

- Code Refactoring (.NET Standard) — C#
- Console App (.NET Core) — C#
- Blank App (Universal Windows) — C#
- xUnit Test Project (.NET Core) — C#
- SQL Server Database Project — Query Language
- WPF App (.NET Framework) — C#
- Blank App (WinUI UWP) — C#
- Class Library (WinUI UWP) — C#
- Console App (.NET Framework) — C#

C# | All platforms | VSSDK

Empty VSIX Project
Build Visual Studio extensions using the Visual Studio SDK. Start with this template to extend functionality of Visual Studio by adding editor items, commands and tool windows.
C# Windows Extensions VSSDK Add-in Plugin

Code Refactoring (.NET Standard)
Create a C# refactoring, deployed as a VSIX extension
C# Windows Linux macOS VSSDK Roslyn Extensions

Analyzer with Code Fix (.NET Standard)
Create a C# analyzer that comes with a code fix and generates diagnostics. The analyzer can be deployed as either a NuGet package or a VSIX extension.
C# Windows Linux macOS VSSDK Roslyn Extensions

Stand-Alone Code Analysis Tool
Create a code analysis command-line application
C# Windows Linux macOS Console VSSDK Roslyn

Next

Update NuGet Packages

File Edit View Project Build Debug Architecture Test Analyze Tools Extensions Window Help Search (Ctrl+Q) Analyzer12

Debug Any CPU Analyzer12.Vsix Live Share PREVIEW

NuGet: Analyzer12 | Resources.resx | Analyzer12Analyzer.cs | Analyzer12CodeFixProvider.cs

Browse Installed Updates 2

NuGet Package Manager: Analyzer12

Search (Ctrl+L) Include prerelease Package source: nuget.org

Select all packages Update

Microsoft.CodeAnalysis.Analyzers by Micro v2.6.2
Analyzers for consumers of Microsoft.CodeAnalysis NuGet package, i.e. extensions and applications bu... v2.9.8

Microsoft.CodeAnalysis.CSharp.Workspa v2.10.0
.NET Compiler Platform ("Roslyn") support for analyzing C# projects and solutions. v3.4.0

Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'Analyzer12' (3 of 3 projects)
  - Analyzer12
    - Dependencies
    - bin
    - tools
    - Analyzer12Analyzer.cs
    - Analyzer12CodeFixProvider.cs
    - Resources.resx
  - Analyzer12.Test
    - Dependencies
    - Helpers
    - Verifiers
    - Analyzer12UnitTests.cs
  - **Analyzer12.Vsix**
    - References
    - source.extension.vsixmanifest

Solution Explorer Team Explorer Notifications

SQL Server Object Explorer

Ready Add to Source Control

Analyzer with Code Fix

Analyzer2 - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Architecture Test Analyze Window Help

Debug | Any CPU | Start

DiagnosticAnalyzer.cs

Analyzer2 | Analyzer2.Analyzer2Analyzer | Initialize(AnalysisContext context)

```
using Microsoft.CodeAnalysis.Diagnostics;

namespace Analyzer2 {
    [DiagnosticAnalyzer(LanguageNames.CSharp)]
    2 references
    public class Analyzer2Analyzer : DiagnosticAnalyzer {
        public const string DiagnosticId = "Analyzer2";

        // You can change these strings in the Resources.resx file. If you do not want your analyzer to be l
        private static readonly LocalizableString Title = new LocalizableResourceString(nameof(Resources.Ana
        private static readonly LocalizableString MessageFormat = new LocalizableResourceString(nameof(Resou
        private static readonly LocalizableString Description = new LocalizableResourceString(nameof(Resourc
        private const string Category = "Naming";

        private static DiagnosticDescriptor Rule = new DiagnosticDescriptor(DiagnosticId, Title, MessageForm

        1 reference
        public override ImmutableArray<DiagnosticDescriptor> SupportedDiagnostics { get { return ImmutableAr
            2
        0 references
        public override void Initialize(AnalysisContext context) {
            // TODO: Consider registering other actions that act on syntax instead of or in addition to symb
            context.RegisterSymbolAction(AnalyzeSymbol, SymbolKind.NamedType);
        }

        1 reference
        private static void AnalyzeSymbol(SymbolAnalysisContext context) {
            // TODO: Replace the following code with your own analysis, generating Diagnostic objects for an
            var namedTypeSymbol = (INamedTypeSymbol)context.Symbol;

            // Find just those named type symbols with names containing lowercase letters.
            if (namedTypeSymbol.Name.ToCharArray().Any(char.IsLower)) {
                // For all such symbols, produce a diagnostic.
                var diagnostic = Diagnostic.Create(Rule, namedTypeSymbol.Locations[0], namedTypeSymbol.Name)

                context.ReportDiagnostic(diagnostic);
```

Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'Analyzer2' (3 projects)
  - Analyzer2 (Portable)
    - Properties
    - References
    - tools
    - CodeFixProvider.cs
    - Diagnostic.nuspec
    - DiagnosticAnalyzer.cs
    - packages.config
    - ReadMe.txt
    - Resources.resx
  - Analyzer2.Test
    - Properties
    - References
    - Helpers
    - Verifiers
    - packages.config
    - UnitTests.cs
  - Analyzer2.Vsix
    - References
    - source.extension.vsixmanifest

Solution Explorer | Team Explorer

Item(s) Saved | Ln 27 | Col 13 | Ch 13 | INS

Analyzer with Code Fix - F5

ConsoleApplication19 - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Architecture Test Analyze Window Help

Debug | Any CPU | Start

Program.cs

ConsoleApplication19 | ConsoleApplication19.Program | Main(string[] args)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace ConsoleApplication19
{
    0 references
    class Program
    }
}
```

Make uppercase
Suppress Analyzer2
Reverse type name

Analyzer2 Type name 'Program' contains lowercase letters

```
...
{
    class Program
    class PROGRAM
    {
...
```

Preview changes

Fix all occurrences in: Document | Project | Solution

Solution Explorer

Search Solution Explorer (Ctrl+;)

Solution 'ConsoleApplication19' (1 project)
ConsoleApplication19
Properties
References
App.config
Program.cs

Solution Explorer | Team Explorer

Properties

Error List

Entire Solution | 0 Errors | 1 Warning | 0 Messages | Build + IntelliSense | Search Error List

Code	Description	Project	File	Line
Analyzer2	Type name 'Program' contains lowercase letters	ConsoleApplication19	Program.cs	9

Ready | Ln 9 | Col 15 | Ch 15 | INS

ProtectedConstructorInAbstractClassAnalyzer

- **CA1012** - Abstract types should not have constructors
  - https://docs.microsoft.com/en-us/visualstudio/code-quality/ca1012-abstract-types-should-not-have-constructors

```
1 reference
abstract class Shape {

    0 references
    public Shape() {

    }
}
```

Fix

```
1 reference
abstract class Shape {

    0 references
    protected Shape() {

    }
}
```

ProtectedConstructorInAbstractClassAnalyzer

```
[DiagnosticAnalyzer(LanguageNames.CSharp)]
public class ProtectedConstructorInAbstractClassAnalyzer : DiagnosticAnalyzer {

    public const string DiagnosticId = "CA1012";

    private static readonly LocalizableString _title = new LocalizableResourceString(nameof(Resources.AnalyzerTitle),
                                                        Resources.ResourceManager, typeof(Resources));
    private static readonly LocalizableString _messageFormat = new LocalizableResourceString(nameof(Resources.AnalyzerMessageFormat),
                                                                Resources.ResourceManager, typeof(Resources));
    private static readonly LocalizableString _description = new LocalizableResourceString(nameof(Resources.AnalyzerDescription),
                                                              Resources.ResourceManager, typeof(Resources));
    private const string Category = "Common Practices and Code Improvements";

    private static readonly DiagnosticDescriptor _rule = new DiagnosticDescriptor(DiagnosticId, _title, _messageFormat, Category,
                                            DiagnosticSeverity.Warning, isEnabledByDefault: true, description: _description);

    public override ImmutableArray<DiagnosticDescriptor> SupportedDiagnostics { get { return ImmutableArray.Create(_rule); } }

    public override void Initialize(AnalysisContext context) {
        context.EnableConcurrentExecution();
        context.ConfigureGeneratedCodeAnalysis(GeneratedCodeAnalysisFlags.None);
        context.RegisterSymbolAction(AnalyzeSymbol, SymbolKind.Method);
    }
```

ProtectedConstructorInAbstractClassAnalyzer

```
    private static void AnalyzeSymbol(SymbolAnalysisContext context) {
        var methodSymbol = (IMethodSymbol)context.Symbol;

        // Find public and internal constructor in abstract class
        if (methodSymbol.MethodKind == MethodKind.Constructor &&
            (methodSymbol.DeclaredAccessibility == Accessibility.Public ||
             methodSymbol.DeclaredAccessibility == Accessibility.Internal) &&
            (methodSymbol.ContainingType?.IsAbstract ?? false)) {

            // For all such symbols, produce a diagnostic.
            var diagnostic = Diagnostic.Create(_rule, methodSymbol.Locations[0], methodSymbol.ContainingType.Name);

            context.ReportDiagnostic(diagnostic);
        }
    }
}
```

NakedIfAnalyzer - DiagnosticAnalyzer

Analyzer1 - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Debug | Any CPU | Analyzer1.Vsix | Current Instance [Visual Studio Professional 2017]

Resources.resx | ProtectedConstruct...sCodeFixProvider.cs | ProtectedConstruct...actClassAnalyzer.cs

Strings | Add Resource | Remove Resource | Access Modifier: Internal

	Name	Value	Comment
▶	AnalyzerDescription	Constructor in Abstract class should be Protected	An optional longer localizable description of the
	AnalyzerMessageFormat	The constructor in the abstract class '{0}' must be protected	The format-able message the diagnostic displays.
	AnalyzerTitle	Constructor in Abstract class should be Protected	The title of the diagnostic.
*			

Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'Analyzer1' (3 projects)
  - Analyzer1
    - Dependencies
    - bin
    - tools
    - ProtectedConstructorInAbstractCla
    - ProtectedConstructorInAbstractCla
    - Resources.resx
  - Analyzer1.Test
    - Dependencies
    - Helpers
    - Verifiers
    - ProtectedConstructorInAbstractCla

Solution Explorer | Team Explorer

Properties

Resources.resx File Properties

Advanced

Build Action: Embedded resource

Build Action

How the file relates to the build and deploym...

Error List | Output

This item does not support previewing | Add to Source Control

ProtectedConstructorInAbstractClassCodeFixProvider

```
[ExportCodeFixProvider(LanguageNames.CSharp, Name = nameof(ProtectedConstructorInAbstractClassCodeFixProvider)), Shared]
public class ProtectedConstructorInAbstractClassCodeFixProvider : CodeFixProvider {

    private const string _title = "Make protected";

    public sealed override ImmutableArray<string> FixableDiagnosticIds {
        get { return ImmutableArray.Create(ProtectedConstructorInAbstractClassAnalyzer.DiagnosticId); }
    }

    public sealed override FixAllProvider GetFixAllProvider() => WellKnownFixAllProviders.BatchFixer;

    public sealed override async Task RegisterCodeFixesAsync(CodeFixContext context) {
        var root = await context.Document.GetSyntaxRootAsync(context.CancellationToken).ConfigureAwait(false);

        // TODO: Replace the following code with your own analysis, generating a CodeAction for each fix to suggest
        var diagnostic = context.Diagnostics.First();
        var diagnosticSpan = diagnostic.Location.SourceSpan;

        // Find the type declaration identified by the diagnostic.
        var constructor = root.FindToken(diagnosticSpan.Start).Parent.AncestorsAndSelf().OfType<ConstructorDeclarationSyntax>().First();

        // Register a code action that will invoke the fix.
        context.RegisterCodeFix(
            CodeAction.Create(title: _title,  createChangedDocument: c => MakeProtectedAsync(context.Document, constructor, c),
                equivalenceKey: _title),
            diagnostic);
    }
```

ProtectedConstructorInAbstractClassCodeFixProvider

```
    private async Task<Document> MakeProtectedAsync(Document document, ConstructorDeclarationSyntax constructor,
                                                    CancellationToken cancellationToken) {

        var generator = SyntaxGenerator.GetGenerator(document);
        var newStatement = generator.WithAccessibility(constructor, Accessibility.Protected);

        // Replace old (public/internal ctor) with the new (protected ctor)
        var oldRoot = await document.GetSyntaxRootAsync(cancellationToken).ConfigureAwait(false);
        var newRoot = oldRoot.ReplaceNode(constructor, newStatement);

        return document.WithSyntaxRoot(newRoot);
    }

}
```

Debug – F5

ConsoleApp64 - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Debug | Any CPU | ConsoleApp64

Toolbox

Demo.cs | Program.cs

ConsoleApp64 | ConsoleApp64.Demo | Demo()

```
1  namespace ConsoleApp64 {
       1 reference
2      abstract class Demo {
3
           0 references
4          public Demo() {
5
6
7
8      }
9
```

Make protected

Suppress CA1012

CA1012 The constructor in the abstract class 'Demo' must be protected

```
...
abstract class Demo {
    public Demo() {
    protected Demo() {
...
```

Preview changes

Fix all occurrences in: Document | Project | Solution

121 %

Error List

Entire Solution | 0 Errors | 1 Warning | 0 Messages | Build + IntelliSense | Search Error List

Code	Description	Project	File	Line
CA1012	The constructor in the abstract class 'Demo' must be protected	ConsoleApp64	Demo.cs	4

Generate NuGet Package

Analyzer1 - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Debug | Any CPU | Analyzer1.Vsix | Current Instance [Visual Studio Professional 2017]

Analyzer1 | Resources.resx | ProtectedConstruct...sCodeFixProvider.cs | ProtectedConstruct...actClassAnalyzer.cs

- Application
- Build
- Build Events
- **Package**
- Debug
- Signing
- Resources

Configuration: N/A Platform: N/A

☑ Generate NuGet package on build

☐ Require license acceptance

Package id:	Analyzer1
Package version:	1.0.0
Authors:	Fons Sonnemans
Company:	Reflection IT
Product:	Analyzer1
Description:	Analyzer1
Copyright:	Copyright
License URL:	http://LICENSE_URL_HERE_OR_DELETE_THIS_LINE
Project URL:	http://PROJECT_URL_HERE_OR_DELETE_THIS_LINE
Icon URL:	http://ICON_URL_HERE_OR_DELETE_THIS_LINE
Repository URL:	http://REPOSITORY_URL_HERE_OR_DELETE_THIS_LINE

Solution Explorer

Search Solution Explorer (Ctrl+;)

- Solution 'Analyzer1' (3 projects)
  - Analyzer1
    - Dependencies
    - bin
      - Debug
        - netstandard1.3
        - Analyzer1.1.0.0.nupkg
    - obj
    - tools
    - ProtectedConstructorInAbstractClassA
    - ProtectedConstructorInAbstractClassC
    - Resources.resx
  - Analyzer1.Test
    - Dependencies

Solution Explorer | Team Explorer

Properties

Toolbox | Test Explorer | Notifications | Diagnostic Tools

Error List | Output

Build succeeded | Add to Source Control

Publish NuGet Package?

NuGet Gallery | Upload

https://www.nuget.org/packages/manage/upload

nuget Packages Upload Statistics Documentation Downloads Blog sonnemaf

Search for packages...

Packages > Upload

Your package file will be uploaded and hosted on the NuGet Gallery server (https://www.nuget.org).

Upload

Browse or Drop files to select a package (.nupkg) or symbols package (.snupkg)... Browse...

Contact

Got questions about NuGet or the NuGet

Status

Find out the service status of NuGet.org

FAQ

Read the Frequently Asked Questions

Create Extension (.vsix)

Analyzer1 - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Debug | Any CPU | Analyzer1.Vsix | Current Instance [Visual Studio Professional 2017]

source.extension.vsixmanifest [Design] | Analyzer1 | Resources.resx | ProtectedConstruct...sCodeFixProvider.cs

Product Name: Analyzer1

Product ID: Analyzer1.4d29177a-d130-428a-8238-9bad335214d7

Author: FonsS

Version: 1.0

- Metadata
- Install Targets
- Assets
- Dependencies
- Prerequisites

Description: This is a sample diagnostic extension for the .NET Compiler Platform ("Roslyn").

Language: English (United States)

License: Browse...

Icon: Browse...

Preview Image: Browse...

Tags:

Release Notes: Browse...

Solution Explorer

Search Solution Explorer (Ctrl+;)

- obj
- tools
- ProtectedConstructorInAbstractClassA
- ProtectedConstructorInAbstractClassC
- Resources.resx
- Analyzer1.Test
  - Dependencies
  - Helpers
  - Verifiers
  - ProtectedConstructorInAbstractClassU
- Analyzer1.Vsix
  - References
  - source.extension.vsixmanifest

Solution Explorer | Team Explorer

Properties

Error List | Output

Ready | Add to Source Control

Unit Test

Analyzer1 - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Test Analyze Window Help

Debug | Any CPU | Analyzer1.Vsix | Current Instance [Visual Studio Professional 2017]

Test Explorer

Run All | Run... | Playlist : All

- Analyzer1 (2 tests)
  - Analyzer1.Test (2) 2 sec
    - Analyzer1.Test (2) 2 sec
      - Protected... (2) 2 sec
        - TestInternal 6 ms
        - TestPublic 2 sec

Group Summary Copy All

Grouped by Hierarchy: ProtectedC

Duration: 0:00:02.6483083

2 Tests Passed

ProtectedConstructo...actClassUnitTest.cs | source.extension.vsixmanifest [Design] | Analyzer1

Analyzer1.Test | Analyzer1.Test.ProtectedConstructorInAb | TestPublic()

```
        //Diagnostic and CodeFix both triggered and checked fo
        [TestMethod]
        public void TestPublic() {
            var test = @"
using System;

namespace ConsoleApplication1
{
    abstract class Demo
    {
        public Demo()
        {
        }
    }
}";
            var expected = new DiagnosticResult {
                Id = "ProtectedConstructorInAbstractClassAnaly
                Message = $"The constructor in the abstract cl
                Severity = DiagnosticSeverity.Warning,
                Locations =
                    new[] {
                        new DiagnosticResultLocation("Test
                    }
            };

            VerifyCSharpDiagnostic(test, expected);
```

104 %

Solution Explorer

Search Solution Explorer (Ctrl+;)

- obj
- tools
- ProtectedConstructorInAbstractClassA
- ProtectedConstructorInAbstractClassC
- Resources.resx
- Analyzer1.Test
  - Dependencies
  - Helpers
  - Verifiers
  - ProtectedConstructorInAbstractClassU
- Analyzer1.Vsix
  - References
  - source.extension.vsixmanifest

Solution Explorer | Team Explorer

Properties

Error List | Output

Build succeeded

Add to Source Control

Closure

Links

- C# - Adding a Code Fix to Your Roslyn Analyzer
  - https://msdn.microsoft.com/en-us/magazine/dn904670.aspx
- Introduction to building code analyzers and code fixes with Roslyn
  - https://channel9.msdn.com/Events/TechDays/Techdays-2016-The-Netherlands/Introduction-to-building-code-analyzers-and-code-fixes-with-Roslyn
- Writing a language-agnostic Roslyn Analyzer
  - https://www.meziantou.net/writing-a-roslyn-analyzer.htm
  - https://www.meziantou.net/writing-a-language-agnostic-roslyn-analyzer-using-ioperation.htm
- The Power of Roslyn: Improving Your Productivity with Live Code Analyzers
  - https://channel9.msdn.com/Events/Ignite/New-Zealand-2016/M344
- Learn Roslyn Now: Part 7 Introducing the Semantic Model
  - https://joshvarty.com/2014/10/30/learn-roslyn-now-part-7-introducing-the-semantic-model/
- http://roslynquoter.azurewebsites.net/
- http://sourceroslyn.io

@fonssonnemans

fons.sonnemans@reflectionit.nl

fonssonnemans

reflectionit.nl/blog

github.com/sonnemaf

Copyright