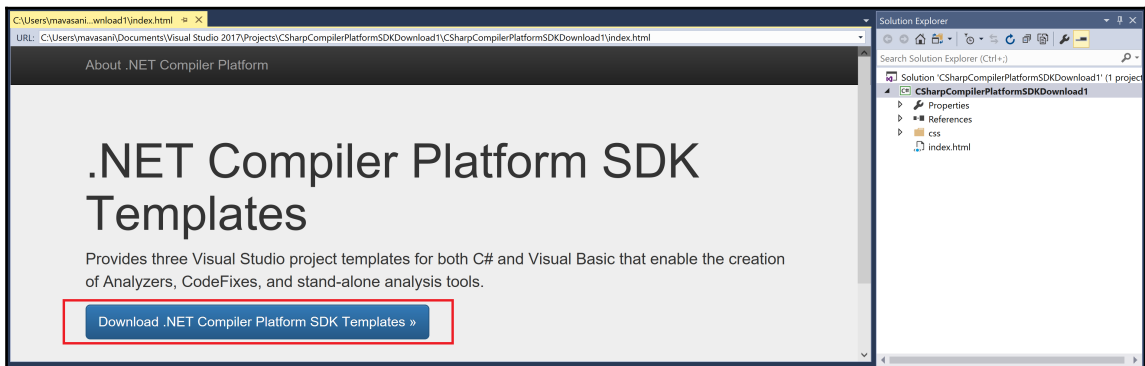
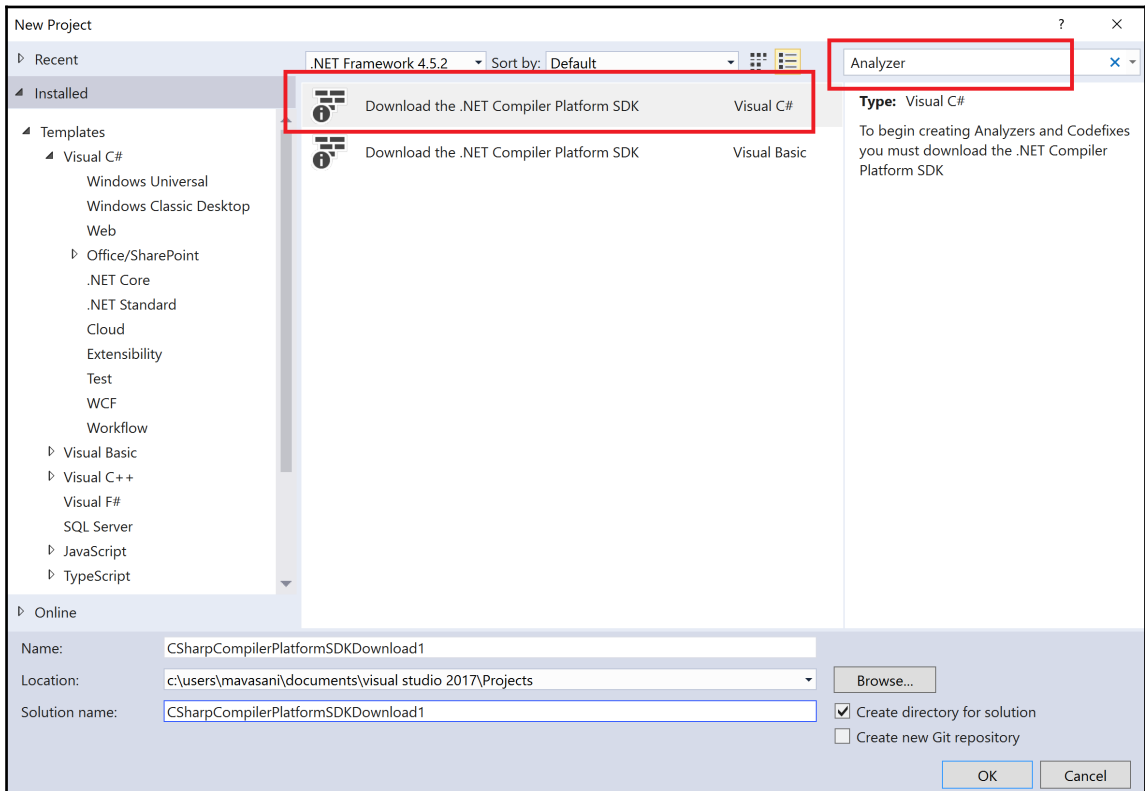
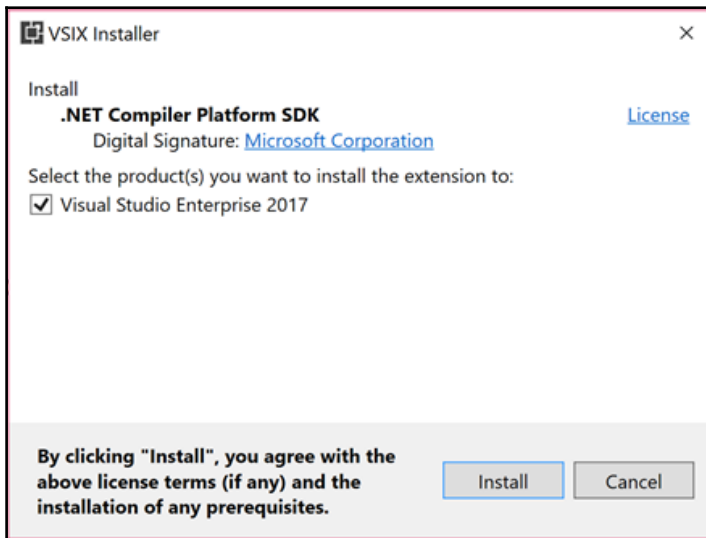
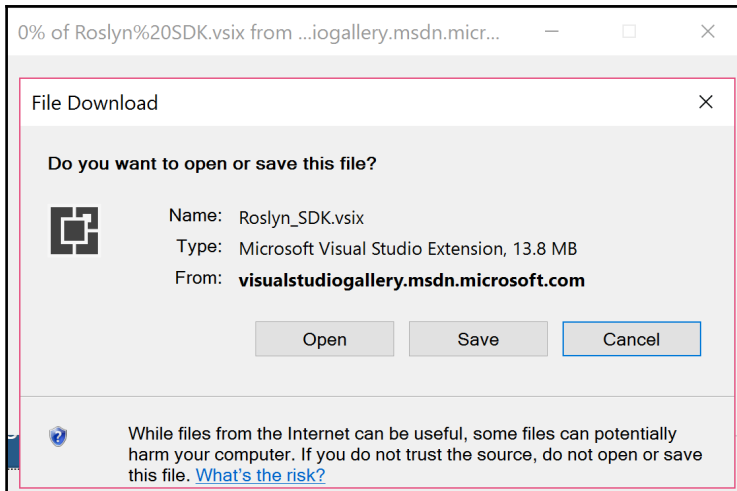
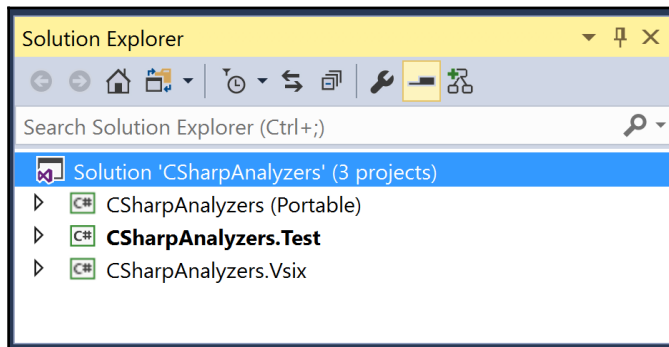
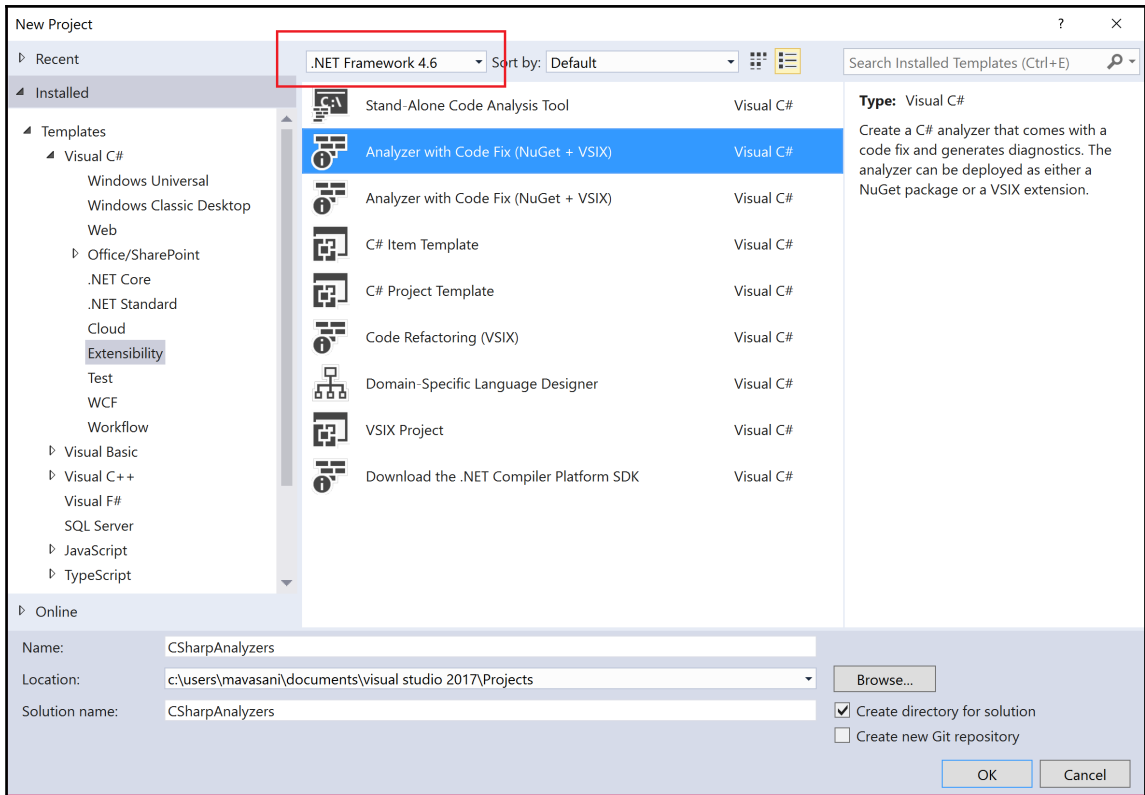


Chapter 1: Writing Diagnostic Analyzers







```

DiagnosticAnalyzer.cs
CSharpAnalyzers
CSharpAnalyzers.CSharpAnalyzersAnalyzer
AnalyzeSymbol(SymbolAnalysisContext context)
27 public override ImmutableArray<DiagnosticDescriptor> SupportedDiagnostics { get { return ImmutableArra
28
29 --references
30 public override void Initialize(AnalysisContext context)
31 {
32 // TODO: Consider registering other actions that act on syntax instead of or in addition to symbol
33 // See https://github.com/dotnet/roslyn/blob/master/docs/analyzers/Analyzer%20Actions%20Semantics.
34 context.RegisterSymbolAction(AnalyzeSymbol, SymbolKind.NamedType);
35 }
36
37 --references
38 private static void AnalyzeSymbol(SymbolAnalysisContext context)
39 {
40 // TODO: Replace the following code with your own analysis, generating Diagnostic objects for any
41 var namedTypeSymbol = (INamedTypeSymbol)context.Symbol;
42
43 // Find just those named type symbols with names containing lowercase letters.
44 if (namedTypeSymbol.Name.ToCharArray().Any(char.IsLower))
45 {
46 // For all such symbols, produce a diagnostic.
47 var diagnostic = Diagnostic.Create(Rule, namedTypeSymbol.Locations[0], namedTypeSymbol.Name);
48 context.ReportDiagnostic(diagnostic);
49 }
50 }

```

```

Class1.cs
ClassLibrary1
ClassLibrary.Class1
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace ClassLibrary
8 {
9     0 references
10     public class Class1
11     {
12     }
13 }

```

class ClassLibrary.Class1
Type name 'Class1' contains lowercase letters
Show potential fixes (Alt+Enter or Ctrl+.)

Error List

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

Code	Description	Project	File	Line	Suppression S...
CA1852	Type name 'Class1' contains lowercase letters	ClassLibrary1	Class1.cs	9	Active

Resources.resx

Strings Add Resource Remove Resource Access Modifier: Internal

Name	Value	Comment
AnalyzerDescription	Do not declare members with same name as any outer type.	An optional longer localizable description of the diagnostic.
AnalyzerMessageFormat	Symbol '{0}' has the same name as one of its outer type, considering renaming the type or the symbol.	The format-able message the diagnostic displays.
AnalyzerTitle	Do not declare members with same name as any outer type.	The title of the diagnostic.

Class1.cs

```

1 namespace ClassLibrary
2 {
3     public class OuterClass
4     {
5         public class NestedClass
6         {
7             public class InnerClass
8             {
9                 public class NestedClass
10                {
11                }
12            }
13        }
14    }
15 }

```

class ClassLibrary.OuterClass.NestedClass.InnerClass.NestedClass

Symbol 'NestedClass' has the same name as one of its outer type, considering renaming the type or the symbol.

Show potential fixes (Alt+Enter or Ctrl+.)

100 %

Error List

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

Code	Description	Project	File	Line	Suppression S...
CSSharpAnalyzers	Symbol 'NestedClass' has the same name as one of its outer type, considering renaming the type or the symbol.	ClassLibrary	Class1.cs	9	Active

Resources.resx

Name	Value	Comment
AnalyzerDescription	Declare explicit type for local declarations.	An optional longer localizable description of the diagnostic.
AnalyzerMessageFormat	Local '0' is implicitly typed. Consider specifying its type explicitly in the declaration.	The format-able message the diagnostic displays.
AnalyzerTitle	Declare explicit type for local declarations.	The title of the diagnostic.

Class1.cs

```

1 namespace ClassLibrary
2 {
3     public class Class1
4     {
5         public void M(int param1, Class1 param2)
6         {
7             // Explicitly typed variables - do not flag.
8             int local1 = param1;
9             Class1 local2 = param2;
10
11            // Implicitly typed variable with error type - do not flag.
12            var local3 = UndefinedMethod();
13
14            // Implicitly typed variable with special type - do not flag.
15            var local4 = param1;
16
17            // Implicitly typed variable with user defined type - flag.
18            var local5 = param2;
19        }
20    }
21 }

```

M(int param1, Class1 param2)

100 %

Error List

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

Code	Description	Project	File	Line	Suppression S...
CS0103	The name 'UndefinedMethod' does not exist in the current context	ClassLibrary	Class1.cs	12	Active
CSSharpAnalyzers	Local 'local5' is implicitly typed. Consider specifying its type explicitly in the declaration.	ClassLibrary	Class1.cs	18	Active

Resources.resx | DiagnosticAnalyzer.cs | UnitTests.cs

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

Name	Value	Comment
AnalyzerDescription	Enclose statement within curly braces	An optional longer localizable description of the diagnostic.
AnalyzerMessageFormat	'{ brace expected	The format-able message the diagnostic displays.
AnalyzerTitle	Enclose statement within curly braces	The title of the diagnostic.

Class1.cs | ClassLibrary | ClassLibrary.Class1 | Method(bool flag, int value)

```

3 | public class Class1
4 | {
5 |     void Method(bool flag, int value)
6 |     {
7 |         while (flag)
8 |             if (value > 0)
9 |                 System.Console.WriteLine(value);
10 |
11 |     }
12 | }
13 |

```

100%

Error List

Entire Solution | 0 Errors | 2 Warnings | 0 Messages | Build + IntelliSense | Search Error List

Code	Description	Project	File	Line	Suppression S...
CSharpAnalyzer	'{ brace expected	ClassLibrary	Class1.cs	8	Active
CSharpAnalyzer	'{ brace expected	ClassLibrary	Class1.cs	9	Active

Class1.cs

```

3  public class Class1
4  {
5      void Method(bool flag, int value)
6      {
7          while (flag)
8              if (value > 0)
9                  {
10                     System.Console.WriteLine(value);
11                 }
12         }
13     }
14 }
15

```

100 %

Error List

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

Code	Description	Project	File	Line	Suppression S...
CSSharpAnalyzer	'!' brace expected	ClassLibrary	Class1.cs	8	Active

Strings Add Resource Remove Resource Access Modifier: Internal

Name	Value	Comment
AnalyzerDescription	Remove unused parameters.	An optional longer localizable description of the diagnostic.
AnalyzerMessageFormat	Parameter '{0}' is unused in the method '{1}'.	The format-able message the diagnostic displays.
AnalyzerTitle	Remove unused parameters.	The title of the diagnostic.

Class1.cs

```

1  namespace ClassLibrary
2  {
3      public class Class1
4      {
5          void M(int param1, ref int param2, int param3, params int[] param4)
6          {
7              int local1 = param1;
8              param2 = 0;
9          }
10     }
11 }

```

100 %

Error List

Entire Solution 0 Errors 2 Warnings 0 Messages Build + IntelliSense Search Error List

Code	Description	Project	File	Line	Suppression S...
CS00101	Parameter 'param3' is unused in the method 'M'.	ClassLibrary	Class1.cs	5	Active
CS00101	Parameter 'param4' is unused in the method 'M'.	ClassLibrary	Class1.cs	5	Active

Class1.cs

```

1 namespace ClassLibrary
2 {
3     0 references
4     public class Class1
5     {
6         0 references
7         void M(int param1, ref int param2, int param3)
8         {
9             int local1 = param1 + param3;
10            param2 = 0;
11        }
12    }
13 }

```

100 %

Error List

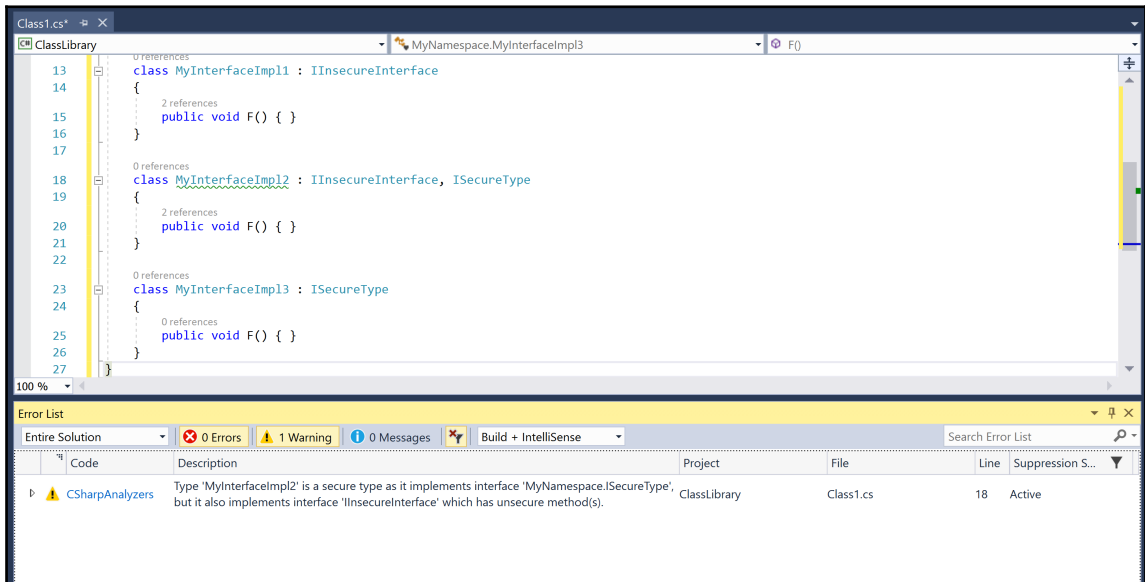
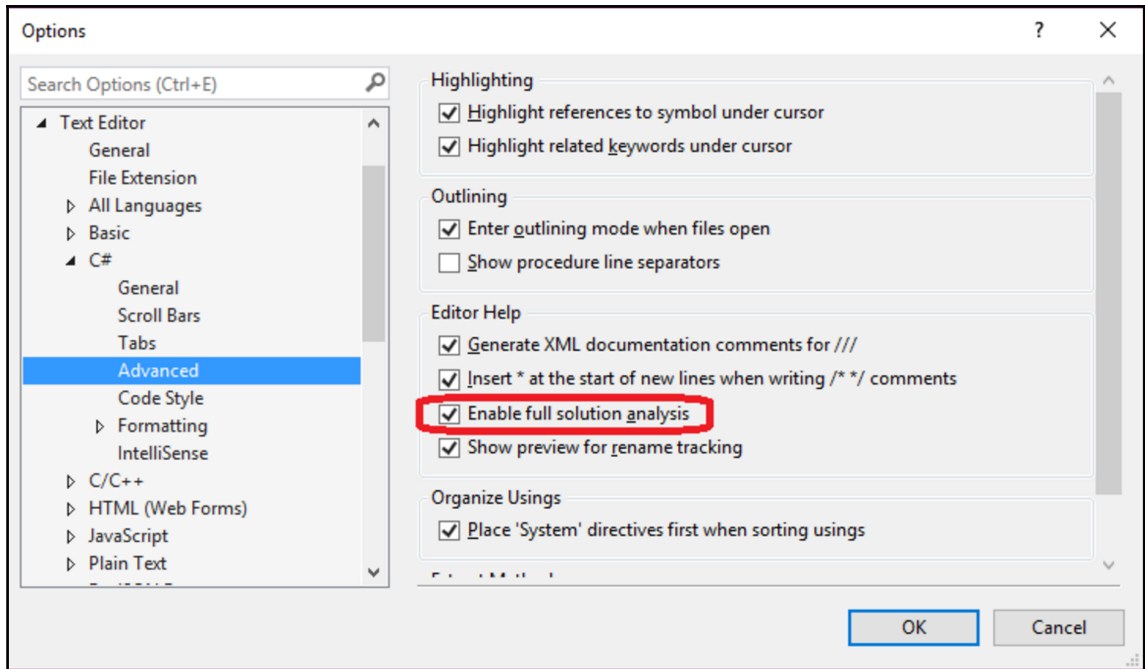
Entire Solution 0 Errors 0 Warnings 0 Messages Build + IntelliSense Search Error List

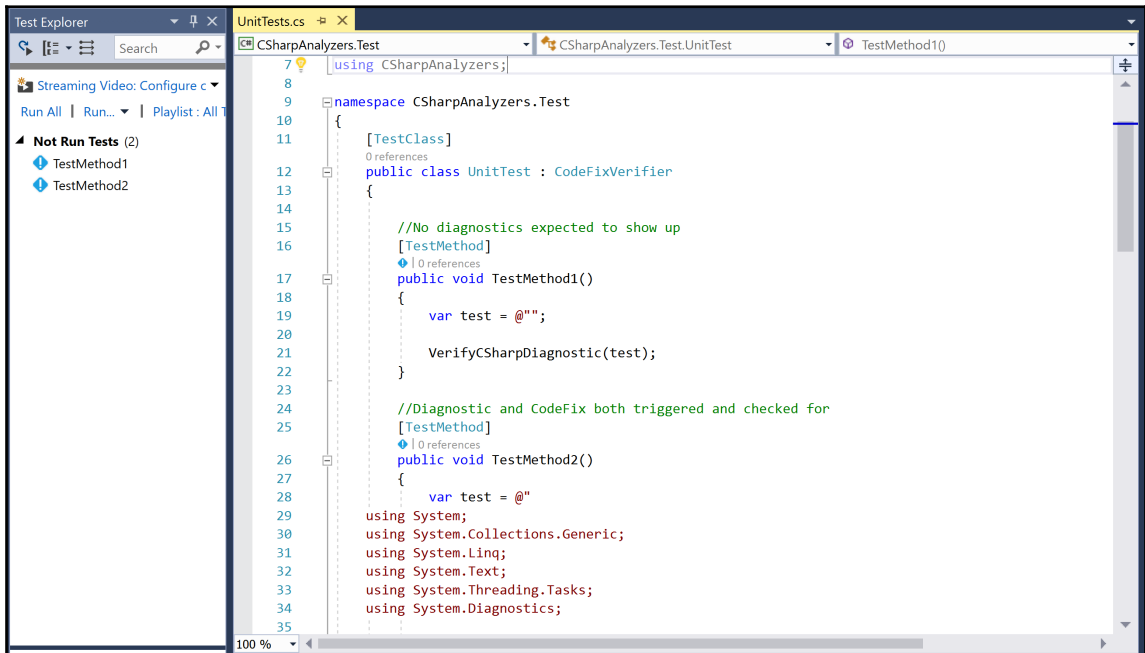
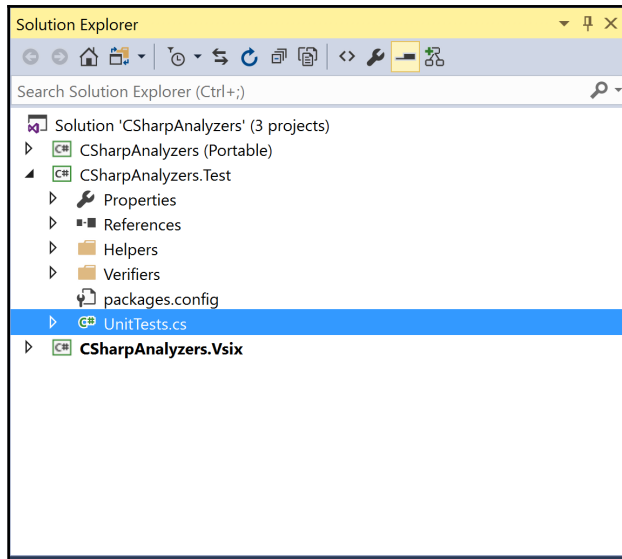
Code	Description	Project	File	Line	Suppression S...
------	-------------	---------	------	------	------------------

Resources.resx

Strings Add Resource Remove Resource Access Modifier: Internal

Name	Value	Comment
AnalyzerDescription	Secure types must not implement interfaces with insecure methods.	An optional longer localizable description of the diagnostic.
AnalyzerMessageFormat	Type '{0}' is a secure type as it implements interface '{1}', but it also implements interface '{2}' which has insecure method(s).	The format-able message the diagnostic displays.
AnalyzerTitle	Secure types must not implement interfaces with insecure methods.	The title of the diagnostic.





Test Explorer

- Failed Tests (1)
 - TestMethod1 1 sec
- Passed Tests (1)
 - TestMethod2 3 sec

TestMethod1 Copy All

Source: UnitTests.cs line 18

Test Failed - TestMethod1

Message: Assert.IsTrue failed. Mismatch between number of diagnostics returned, expected "0" actual "1"

Diagnostics:
 // Test0.cs(1,7): warning CSharpAnalyzers: Type name 'Class1' contains lowercase letters
 GetCSharpResultAt(1, 7, CSharpAnalyzersAnalyzer.CSharpAnalyzers)

Elapsed time: 0:00:01.8705795

Stack Trace:

- DiagnosticVerifier.VerifyDiagnosticResults(IEnumerable`1 actual, DiagnosticVerifier.VerifyDiagnostics(String[]) sources, String language, DiagnosticVerifier.VerifyCSharpDiagnostic(String source, DiagnosticUnitTest.TestMethod1)

UnitTests.cs

```

using CSharpAnalyzers;

namespace CSharpAnalyzers.Test
{
    [TestClass]
    public class UnitTest : CodeFixVerifier
    {
        //No diagnostics expected to show up
        [TestMethod]
        public void TestMethod1()
        {
            var test = @"class Class1 { }";

            VerifyCSharpDiagnostic(test);
        }

        //Diagnostic and CodeFix both triggered and checked for
        [TestMethod]
        public void TestMethod2()
        {
            var test = @"
  
```

Output

```

----- Discover test started -----
----- Discover test finished: 2 found (0:00:00.1659955) -----
----- Run test started -----
----- Run test finished: 1 run (0:00:02.0870936) -----
  
```

Test Explorer

- Failed Tests (1)
 - TestMethod1 1 sec
- Passed Tests (1)
 - TestMethod2 3 sec

TestMethod1 Copy All

Source: UnitTests.cs line 18

Test Failed - TestMethod1

Message: Assert.IsTrue failed. Expected diagnostic to start at column "15" was actually at column "7"

Diagnostics:
 // Test0.cs(1,7): warning CSharpAnalyzers: Type name 'Class1' contains lowercase letters
 GetCSharpResultAt(1, 7, CSharpAnalyzersAnalyzer.CSharpAnalyzers)

UnitTests.cs

```

//No diagnostics expected to show up
[TestMethod]
public void TestMethod1()
{
    var test = @"class Class1 { }";

    var expected = new DiagnosticResult
    {
        Id = "CSharpAnalyzers",
        Message = String.Format("Type name '{0}' contains lowercase letter", "Class1"),
        Severity = DiagnosticSeverity.Warning,
        Locations =
            new[] {
                new DiagnosticResultLocation("Test0.cs", 11, 15)
            }
    };

    VerifyCSharpDiagnostic(test, expected);
}
  
```

Output

```

----- Discover test started -----
----- Discover test finished: 2 found (0:00:00.1600076) -----
----- Run test started -----
----- Run test finished: 1 run (0:00:02.0870936) -----
  
```

Test Explorer

- Streaming Video: Configure continuous integration
- Run All | Run... | Playlist: All Tests
- Failed Tests (1)
 - TestMethod1 2 sec
- Passed Tests (1)
 - TestMethod2 3 sec

TestMethod1 Copy All

Source: UnitTests.cs line 18

Test Failed - TestMethod1

Message: Assert.IsTrue failed. Mismatch between number of diagnostics returned, expected "1" actual "0"

Diagnostics: NONE

Elapsed time: 0:00:02.0896255

StackTrace:

```
DiagnosticVerifier.VerifyDiagnosticResults(IEnumerable<1 actual
DiagnosticVerifier.VerifyDiagnostics(String[] sources, String lan
DiagnosticVerifier.VerifyCSharpDiagnostic(String source, Diagn
UnitTest.TestMethod1()
```

```

14
15 //No diagnostics expected to show up
16 [TestMethod]
17 public void TestMethod1()
18 {
19     var test = @"class CLASS1 { }";
20
21
22     var expected = new DiagnosticResult
23     {
24         Id = "CSharpAnalyzers",
25         Message = String.Format("Type name '{0}' contains lowercase letter
26         Severity = DiagnosticSeverity.Warning,
27         Locations =
28             new[] {
29                 new DiagnosticResultLocation("Test0.cs", 11, 7)
30             }
31     };
32
33     VerifyCSharpDiagnostic(test, expected);
34 }
35
36 //Diagnostic and CodeFix both triggered and checked for

```

Output

Show output from: Tests

```

----- Discover test started -----
===== Discover test finished: 2 found (0:00:00.2360111) =====
----- Run test started -----
===== Run test finished: 1 run (0:00:02.5811169) =====

```

DiagnosticVerifier.cs

```

9 namespace TestHelper
10 {
11     /// <summary>
12     /// Superclass of all Unit Tests for DiagnosticAnalyzers
13     /// </summary>
14     public abstract partial class DiagnosticVerifier
15     {
16         [To be implemented by Test classes]
17
18         Verifier wrappers
19
20         Actual comparisons and verifications
21
22         Formatting Diagnostics
23     }
24 }
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71

```

DiagnosticVerifier.Helper.cs

```

10 namespace TestHelper
11 {
12     /// <summary>
13     /// Class for turning strings into documents and getting
14     /// All methods are static
15     /// </summary>
16     public abstract partial class DiagnosticVerifier
17     {
18         private static readonly MetadataReference CorlibRefer
19         private static readonly MetadataReference SystemCoreR
20         private static readonly MetadataReference CSharpSymbo
21         private static readonly MetadataReference CodeAnalyssi
22
23         Internal static string DefaultFilePathPrefix = "Test"
24         Internal static string CSharpDefaultFileExt = ".cs";
25         Internal static string VisualBasicDefaultExt = ".vb";
26         Internal static string TestProjectName = "TestProject
27
28         get Diagnostics
29
30         Set up compilation and documents
31     }
32 }
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71

```

Solution Explorer

- Solution 'CSharpAnalyzers' (3 projects)
 - CSharpAnalyzers (Portable)
 - CSharpAnalyzers.Test
 - Properties
 - References
 - Helpers
 - CodeFixVerifier.Helper.cs
 - DiagnosticResult.cs
 - DiagnosticVerifier.Helper.cs
 - Verifiers
 - CodeFixVerifier.cs
 - DiagnosticVerifier.cs
 - packages.config
 - UnitTests.cs
 - CSharpAnalyzers.Vsix

```
Diagnostic.nuspec  X
1 <?xml version="1.0"?>
2 <package xmlns="http://schemas.microsoft.com/packaging/2011/08/nuspec.xsd">
3   <metadata>
4     <id>CSharpAnalyzers</id>
5     <version>1.0.0.0</version>
6     <title>CSharpAnalyzers</title>
7     <authors>mavasani</authors>
8     <owners>mavasani</owners>
9     <licenseUrl>http://LICENSE_URL_HERE_OR_DELETE_THIS_LINE</licenseUrl>
10    <projectUrl>http://PROJECT_URL_HERE_OR_DELETE_THIS_LINE</projectUrl>
11    <iconUrl>http://ICON_URL_HERE_OR_DELETE_THIS_LINE</iconUrl>
12    <requireLicenseAcceptance>>false</requireLicenseAcceptance>
13    <description>CSharpAnalyzers</description>
14    <releaseNotes>Summary of changes made in this release of the package.</releaseNotes>
15    <copyright>Copyright</copyright>
16    <tags>CSharpAnalyzers, analyzers</tags>
17    <frameworkAssemblies>
18      <frameworkAssembly assemblyName="System" targetFramework="" />
19    </frameworkAssemblies>
20    <developmentDependency>>true</developmentDependency>
21  </metadata>
22  <!-- The convention for analyzers is to put language agnostic dlls in analyzers\portable50 and language specific analyzers in eit
23  <files>
24    <file src="*.dll" target="analyzers\dotnet\cs" exclude="**\Microsoft.CodeAnalysis.*;**\System.Collections.Immutable.*;**\System
25    <file src="tools\*.ps1" target="tools\" />
26  </files>
27 </package>
```

source.extension.vsixmanifest [Design] X

Product Name:	CSharpAnalyzers	Author:	mavasani
Product ID:	CSharpAnalyzers..69444f8f-dc5f-4884-90fc-6a16a0b30294	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)

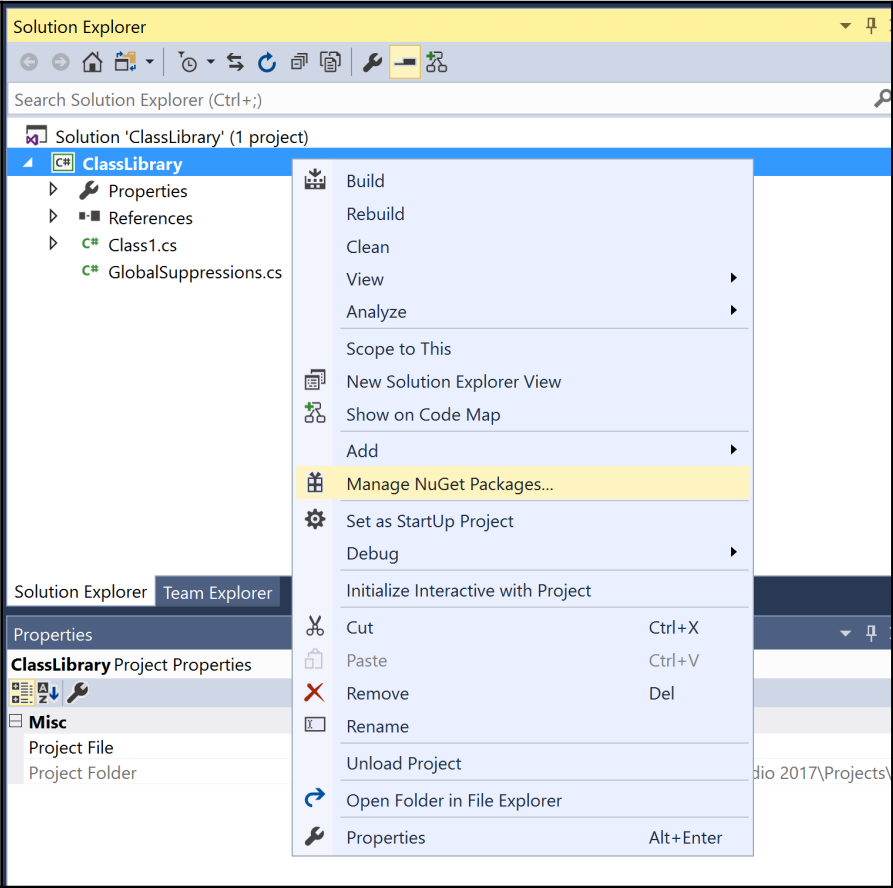
License: Browse...

Icon: Browse...

Preview Image: Browse...

Tags:





Chapter 2: Consuming Diagnostic Analyzers in .NET Projects



NuGet: ClassLibrary [X] Class1.cs

Browse Installed Updates NuGet Package Manager: ClassLibrary

Search (Ctrl+E) [X] [Refresh] [Include prerelease] Package source: nuget.org [Settings]

-  **Newtonsoft.Json** by James Newton-King, **50.7M** downloads v10.0.1-beta1
Prerelease
Json.NET is a popular high-performance JSON framework for .NET
-  **EntityFramework** by Microsoft, **27.5M** downloads v6.1.3
Entity Framework is Microsoft's recommended data access technology for n...
-  **NUnit** by Charlie Poole, **8.59M** downloads v3.6.1
NUnit is a unit-testing framework for all .NET languages with a strong TDD f...
-  **jQuery** by jQuery Foundation, Inc., **32.4M** downloads v3.1.1
jQuery is a new kind of JavaScript Library.





Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

Do not show this again

NuGet: ClassLibrary [X] Class1.cs

Browse Installed Updates NuGet Package Manager: ClassLibrary

Tags:"analyzers" [X] [Refresh] [Include prerelease] Package source: nuget.org [Settings]

-  **Microsoft.CodeAnalysis.Analyzers** by Microsoft, **3.25M** dow v1.2.0-beta2
Prerelease
Analyzers for .NET Compiler Platform ("Roslyn")
-  **System.Runtime.Analyzers** by Microsoft, **167K** downloads v1.2.0-beta2
Prerelease
System.Runtime Analyzers
-  **System.Runtime.InteropServices.Analyzers** by Microsoft, v1.2.0-beta2
Prerelease
System.Runtime.InteropServices Analyzers
-  **Microsoft.Net.RoslynDiagnostics** by Microsoft, **84.5K** downl v1.2.0-beta2
Prerelease
Diagnostic Analyzers consumed by Roslyn.sln (https://github.com/d...

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

Do not show this again

NuGet: ClassLibrary Class1.cs

Browse Installed Updates

Tags: "analyzers" Include prerelease Package source: nuget.org

Microsoft.CodeAnalysis.Analyzers by Microsoft, 3.2 v1.2.0-beta2
Prerelease Analyzers for .NET Compiler Platform ("Roslyn")

System.Runtime.Analyzers by Microsoft, 167K downloads v1.2.0-beta2
Prerelease System.Runtime Analyzers

System.Runtime.InteropServices.Analyzers by Microsoft, v1.2.0-beta2
Prerelease System.Runtime.InteropServices Analyzers

Microsoft.Net.RoslynDiagnostics by Microsoft, 84.5 v1.2.0-beta2
Prerelease Diagnostic Analyzers consumed by Roslyn.sln (https://github.com/...)

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.
 Do not show this again

System.Runtime.Analyzers

Version: Latest prerelease 1.2.0-beta2 Install

Latest prerelease 1.2.0-beta2

Latest stable 1.1.0

Options: 1.2.0-beta2, 1.2.0-beta1, 1.1.0

Description: System.Runtime 1.1.0-beta1-20150812-01

Version: 1.0.1, 1.0.0-rc3, 1.0.0-rc2

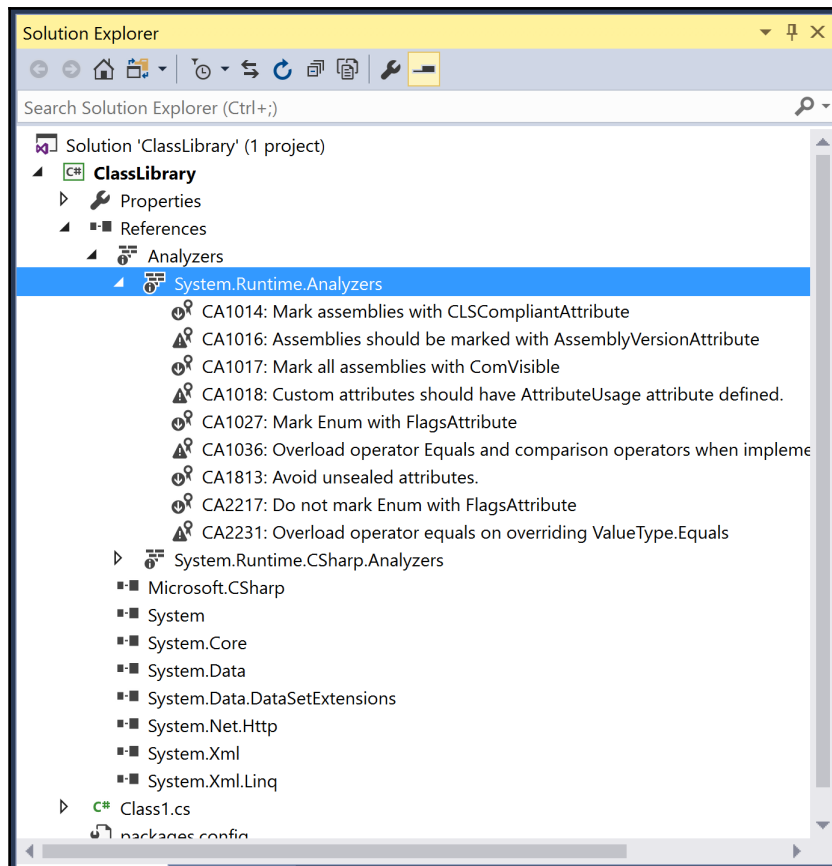
Author(s):

License: <http://go.microsoft.com/fwlink/?LinkId=529443>

Date published: Wednesday, May 18, 2016 (5/18/2016)

Project URL: <https://github.com/dotnet/roslyn-analyzers>

Report Abuse: <https://www.nuget.org/packages/System.Runtime.Analyzers/1.2.0-beta2/ReportAbuse>



Code Editor: `namespace ClassLibrary`
`{`
`0 references`
`public class MyAttribute : System.Attribute`
`{`
`}`

Solution Explorer: ClassLibrary (1 project)
References
Analyzers
System.Runtime.Analyzers
CA1014: Mark assemblies with CLSCompliantAttribute
CA1016: Assemblies should be marked with AssemblyVersionAttribute
CA1018: Custom attributes should have AttributeUsage attribute defined.
CA1027: Mark Enum with FlagsAttribute
CA1036: Overload operator Equals and comparison operators when imple
CA1813: Avoid unsealed attributes.
CA2217: Do not mark Enum with FlagsAttribute
CA2231: Overload operator equals on overriding ValueType.Equals
System.Runtime.CSharp.Analyzers

Error List: 0 Errors, 1 Warning, 0 Messages
Build + IntelliSense
CA1018 Specify AttributeUsage attribute on 'MyAttribute' attribute class. ClassLibrary Class1.cs 3 Active

CA1018 Diagnostic Properties
Category: Design
Default severity: Warning
Description: Specify AttributeUsage attribute on '00' attribute
Effective severity: Warning
Enabled by default: True
Help link: <http://msdn.microsoft.com/library/ms182158.as>
ID: CA1018
Message: Specify AttributeUsage attribute on '00' attribute
Tags: Telemetry
Title: Custom attributes should have AttributeUsage a
Category

```
Administrator: Developer Command Prompt for VS 2017  
C:\Users\mavasani\Documents\Visual Studio 2017\Projects\ClassLibrary\ClassLibrary>msbuild ClassLibrary.csproj /v:m  
Microsoft (R) Build Engine version 15.1.548.43366  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Class1.cs(3,18): warning CA1018: Specify AttributeUsage attribute on 'MyAttribute' attribute class. [C:\Users\mavasani\Documents\Visual Studio 2017\Projects\ClassLibrary\ClassLibrary\ClassLibrary.csproj]  
ClassLibrary -> C:\Users\mavasani\Documents\Visual Studio 2017\Projects\ClassLibrary\ClassLibrary\bin\Debug\ClassLibrary.dll  
C:\Users\mavasani\Documents\Visual Studio 2017\Projects\ClassLibrary\ClassLibrary>
```

Extensions and Updates

Sort by: Most Popular

Search Visual Studio Marketplace (Ctrl+)

- Installed
 - Online
 - Visual Studio Marketplace
 - Controls
 - Templates
 - Tools
 - Updates (1)
 - Roaming Extension Manager

File Icons
Adds icons for files that are not recognized by Solution Explorer

Syntax Highlighting Pack
Adds syntax highlighting and snippet support for a wide variety of programming languages such as Clojure, Go, Jade, Lua, Swift, Ruby...

GitHub Extension for Visual Studio
A Visual Studio Extension that brings the GitHub Flow into Visual Studio.

Package Security Alerts
Identifies npm and Bower packages that contains known security vulnerabilities to ensure your project is always using the most secur...

Markdown Editor
A full featured Markdown editor with live preview and syntax highlighting. Supports GitHub flavored Markdown.

HTML Snippet Pack
A snippet pack to make you more productive working with HTML.

Web Accessibility Checker
The easiest way to perform accessibility checks on any ASP.NET web application. Fully customizable and support all the major internation...

Editor Enhancements
Provides additional features such as HTML and URL encodings.

1 2 3 4 5

Created by: Mads Kristensen
Version: 2.5.164
Downloads: 179403
Rating: ★★★★★ (9 Votes)
[More Information](#)
[Report Extension to Microsoft](#)

Scheduled For Install: None
Scheduled For Update: None
Scheduled For Uninstall: None

[Change your Extensions and Updates settings](#)

Close

Extensions and Updates

Sort by: Relevance

Search:

- Installed
- Online
 - Visual Studio Marketplace
 - Controls
 - Templates
 - Tools
 - Search Results
 - Updates (1)
 - Roaming Extension Manager

Roslynator 2017
A collection of 170+ analyzers and 170+ refactorings for C#, powered by Roslyn.

Roslyn Security Guard
Roslyn analyzers that aim to help security audit on .NET applications.

Hot Commands for Visual Studio
A collection of commands, analyzers, and refactorings for enhanced productivity in Visual Studio IDE

Var Replacer C#
This C# analyzer plugin will help you replace "var" with the actual type name. Use the light bulb to change it. If you want to have "var"...

Visual Studio Project System Extensibility Preview
Provides templates for defining new project types (with unique project file extensions) or modifying existing project types in Visual...

Roslynator Refactorings 2017
A collection of 170+ refactorings for C#, powered by Roslyn.

Refactoring Essentials for Visual Studio...
Free refactorings for C# and Visual Basic (VB) - plus more!
[Download](#)

SonarLint for Visual Studio
SonarLint provides on-the-fly feedback to developers on new bugs.

Created by: IC#Code
Version: 5.0.0.0
Downloads: 151
Rating: ★★★★★ (0 Votes)
[More Information](#)
[Report Extension to Microsoft](#)

Scheduled For Install: None
Scheduled For Update: None
Scheduled For Uninstall: None

1

[Change your Extensions and Updates settings](#)

Close

Extensions and Updates

Sort by: Relevance

Installed

- Online
- Visual Studio Marketplace
 - Controls
 - Templates
 - Tools
 - Search Results
- Updates (1)
- Roaming Extension Manager

Search results for "analyzers":

- Roslynator 2017**
A collection of 170+ analyzers and 170+ refactorings for C#, powered by Roslyn.
- Roslyn Security Guard**
Roslyn analyzers that aim to help security audit on .NET applications.
- Hot Commands for Visual Studio**
A collection of commands, analyzers, and refactorings for enhanced productivity in Visual Studio IDE
- Var Replacer C#**
This C# analyzer plugin will help you replace "var" with the actual type name. Use the light bulb to change it. If you want to have "var"...
- Visual Studio Project System Extensibility Preview**
Provides templates for defining new project types (with unique project file extensions) or modifying existing project types in Visual...
- Roslynator Refactorings 2017**
A collection of 170+ refactorings for C#, powered by Roslyn.
- Refactoring Essentials for Visual Studio 2017**
Free refactorings for C# and Visual Basic (VB) - plus more!
- SonarLint for Visual Studio 2017**
SonarLint provides on-the-fly feedback to developers on new bugs

1

Change your Extensions and Updates settings

⚠ Your changes will be scheduled. The selected installs, updates, and uninstalls will begin when all Microsoft Visual Studio windows are closed. Close

Search: analyzers

Created by: IC#Code
Version: 5.0.0.0
Downloads: 151
Rating: ★★★★★ (0 Votes)
[More Information](#)
[Report Extension to Microsoft](#)

Scheduled For Install:
Refactoring Essentials for Visual St... ×

Scheduled For Update:
None

Scheduled For Uninstall:
None

VSIX Installer

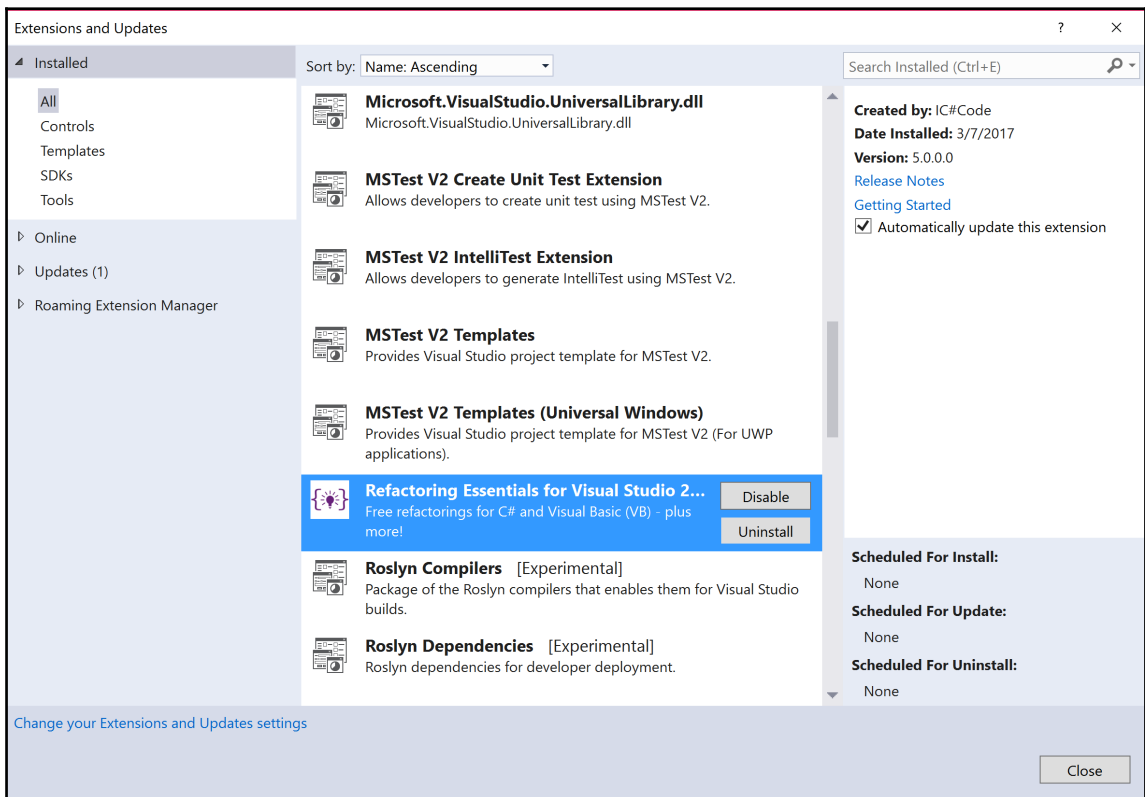
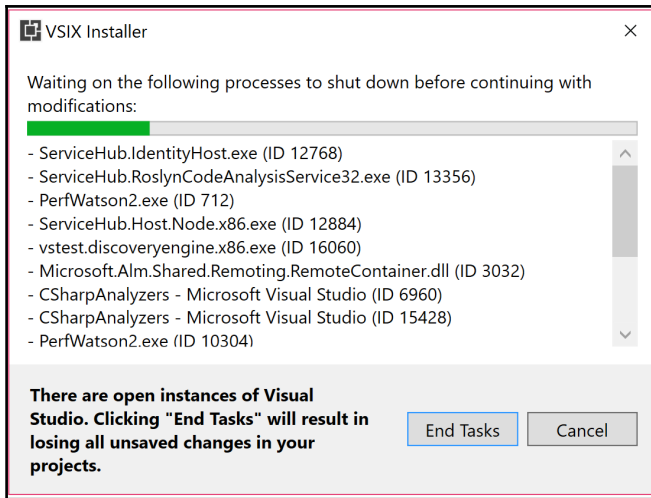
Scheduled tasks for Microsoft Visual Studio Enterprise 2017:

Install

Refactoring Essentials for Visual Studio 2017 [License](#)
 Digital Signature: None [Release Notes](#)

By clicking "Modify", you agree with the above license terms (if any) and the installation of any prerequisites.

Modify Cancel



Class1.cs

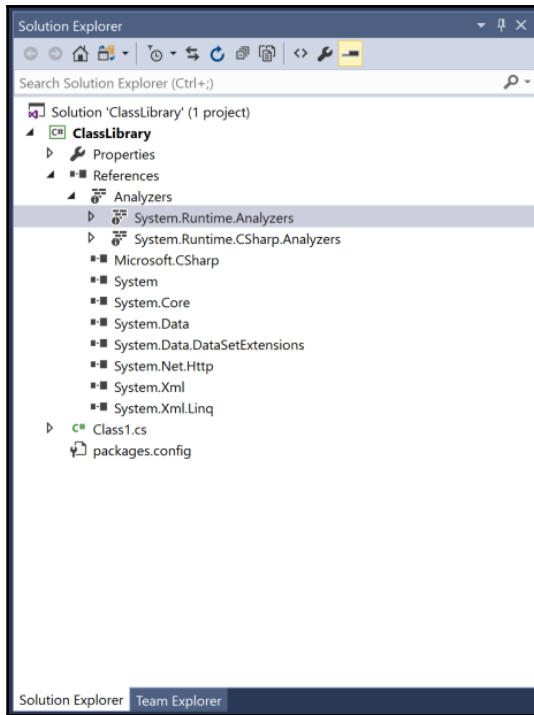
```
1 namespace ClassLibrary
2 {
3     0 references
4     public class Class1
5     {
6         0 references
7         void Method()
8         {
9             int[] values = new int[] { 1, 2, 3 };
10        }
11    }
12 }
```

100 %

Error List

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

Code	Description	Project	File	Line	Suppression S...
RECS0085	Redundant array creation expression	ClassLibrary	Class1.cs	7	Active



Solution Explorer

Search Solution Explorer (Ctrl+;)

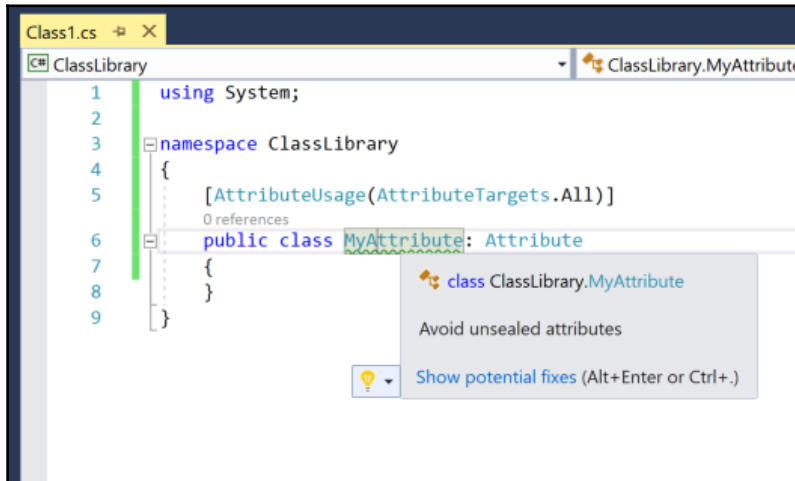
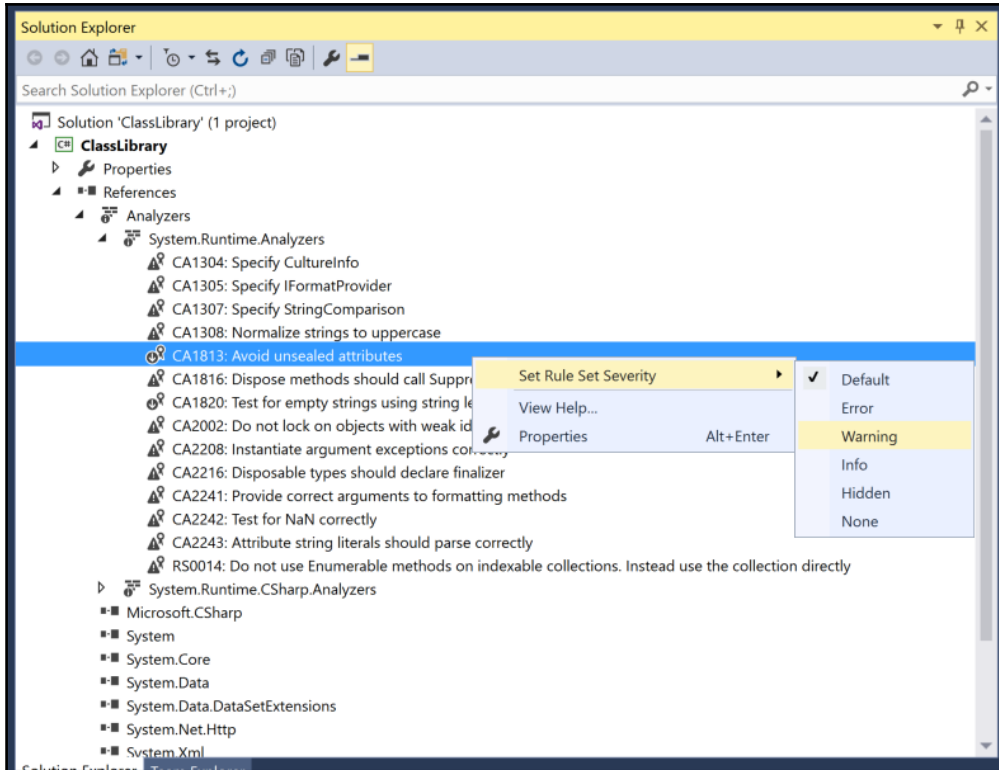
- ClassLibrary
 - Properties
 - References
 - Analyzers
 - System.Runtime.Analyzers
 - CA1304: Specify CultureInfo
 - CA1305: Specify IFormatProvider
 - CA1307: Specify StringComparison
 - CA1308: Normalize strings to uppercase
 - CA1813: Avoid unsealed attributes**
 - CA1816: Dispose methods should call SuppressFinalize
 - CA1820: Test for empty strings using string length
 - CA2002: Do not lock on objects with weak identity
 - CA2208: Instantiate argument exceptions correctly
 - CA2216: Disposable types should declare finalizer
 - CA2241: Provide correct arguments to formatting methods
 - CA2242: Test for NaN correctly
 - CA2243: Attribute string literals should parse correctly
 - RS0014: Do not use Enumerable methods on indexable collections. Instead use the collection directly
 - System.Runtime.CSharp.Analyzers
 - Microsoft.CSharp
 - System
 - System.Core
 - System.Data
 - System.Data.DataSetExtensions
 - System.Net.Http
 - System.Xml
 - System.Xml.Linq

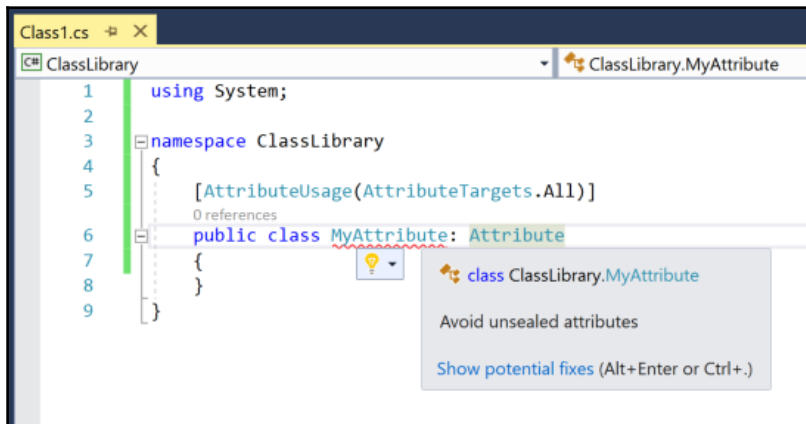
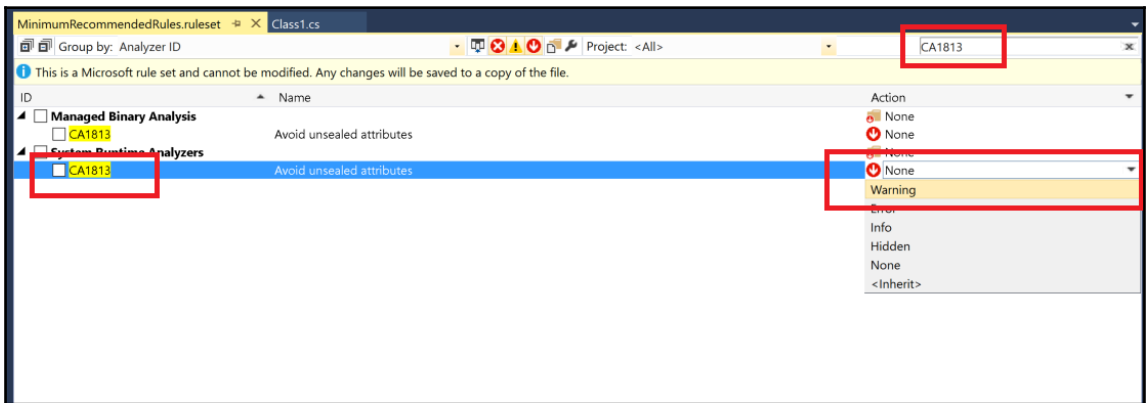
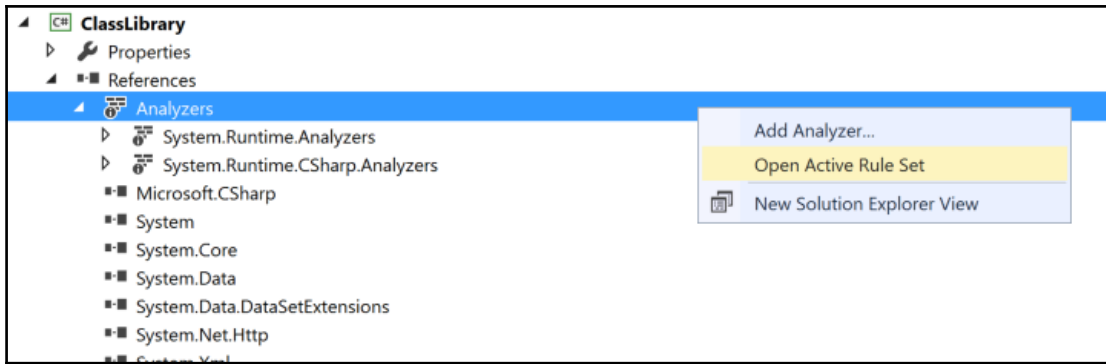
Solution Explorer | Team Explorer

Properties

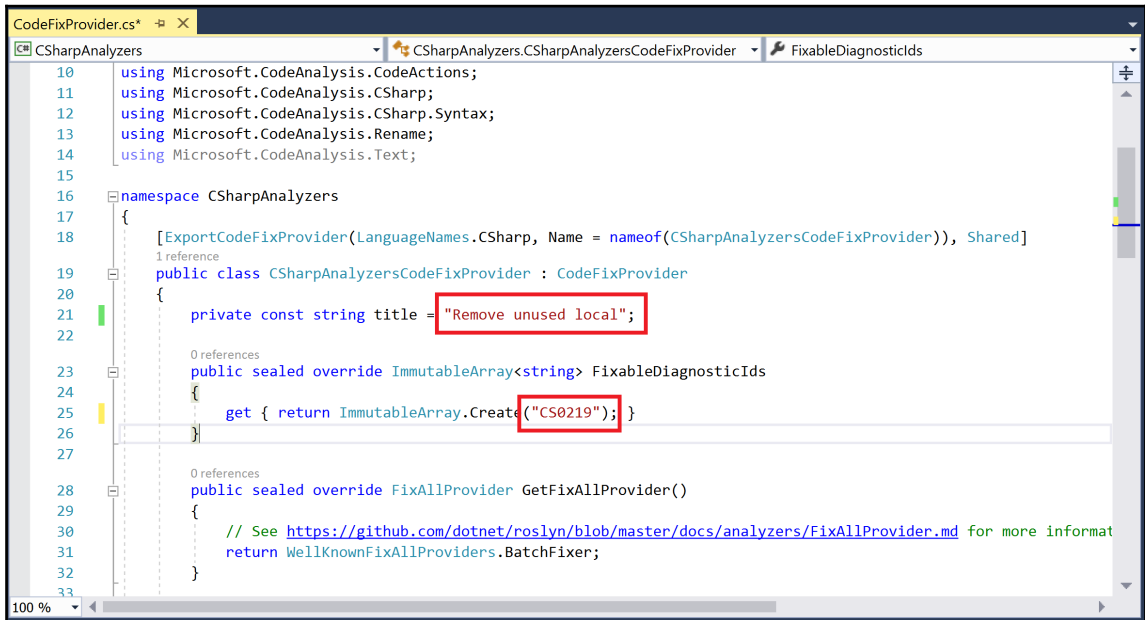
CA1813 Diagnostic Properties

Misc	
Category	Performance
Default severity	Warning
Description	The .NET Framework class library provides methods for retrieving custom attri
Effective severity	Suppressed
Enabled by default	False
Help link	http://msdn.microsoft.com/library/ms182267.aspx
ID	CA1813
Message	Avoid unsealed attributes
Tags	Telemetry
Title	Avoid unsealed attributes





Chapter 3: Writing IDE Code Fixes, Refactorings, and Intellisense Completion Providers



```
CodeFixProvider.cs* X
CSharpAnalyzers
CSharpAnalyzers.CSharpAnalyzersCodeFixProvider
FixableDiagnosticIds

10 using Microsoft.CodeAnalysis.CodeActions;
11 using Microsoft.CodeAnalysis.CSharp;
12 using Microsoft.CodeAnalysis.CSharp.Syntax;
13 using Microsoft.CodeAnalysis.Rename;
14 using Microsoft.CodeAnalysis.Text;
15
16 namespace CSharpAnalyzers
17 {
18     [ExportCodeFixProvider(LanguageNames.CSharp, Name = nameof(CSharpAnalyzersCodeFixProvider)), Shared]
19     public class CSharpAnalyzersCodeFixProvider : CodeFixProvider
20     {
21         private const string title = "Remove unused local";
22
23         public sealed override ImmutableArray<string> FixableDiagnosticIds
24         {
25             get { return ImmutableArray.Create("CS0219"); }
26         }
27
28         public sealed override FixAllProvider GetFixAllProvider()
29         {
30             // See https://github.com/dotnet/roslyn/blob/master/docs/analyzers/FixAllProvider.md for more informat
31             return WellKnownFixAllProviders.BatchFixer;
32         }
33     }

```

```
Class1.cs [X]
ClassLibrary11 Class1
1 public class Class1
2 {
3     References
4     public void Method1()
5     {
6         // Local declaration statement with unused local ('a')
7         int a = 0;
8         // Local declaration statement with a used ('c') and unused local ('b').
9         int b = 0;
10        System.Console.WriteLine("c: {0}, b: {0}", c, b);
11        // Local declaration statement with unused local ('d') and unused local ('e').
12        // Local declaration statement with unused local ('d') initializer is non-constant.
13        int d = c;
14
15        // Local declaration statement with errors ('e').
16        if (true)
17            var e = 1;
18    }
19 }
```

Lightbulb icon: (local variable) int a
The variable 'a' is assigned but its value is never used
Show potential fixes (Alt+Enter or Ctrl+.)

```
public void Method1()
{
    // Local declaration statement with unused local ('a')
    int a = 0;
    // Local declaration statement with a used ('c') and unused local ('b')
    System.Console.WriteLine("c: {0}, b: {0}", c, b);
    // Local declaration statement with unused local ('d') and unused local ('e').
    int d = c;
    // Local declaration statement with errors ('e').
    if (true)
        var e = 1;
}
```

Remove unused local
CS0219 The variable 'a' is assigned but its value is never used
Preview changes
Fix all occurrences in: Document | Project | Solution

```
3 public void Method1()
4 {
5
6 // Local declaration statement with a used ('c') and unused local ('b').
7 int b = 1, c = 2;
8
9 // Local declaration statement with a used ('c') and unused local ('b').
10 int c = 1, d = 2;
11 // Local declaration statement with a used ('c') and unused local ('b').
12 int b = 1, c = 2;
13 // Local declaration statement where unused local ('d') initializer is non-constant.
14 System.Console.WriteLine(c);
15 if (true)
16     var d = 1;
17 }
```

Remove unused local

CS0219 The variable 'b' is assigned but its value is never used.

Suppress CS0219

Preview changes

Fix all occurrences in: Document | Project | Solution

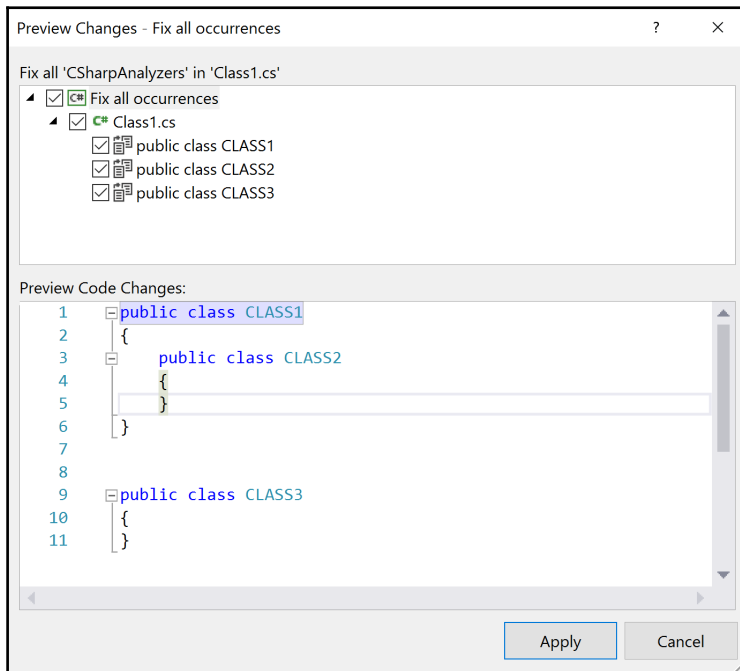
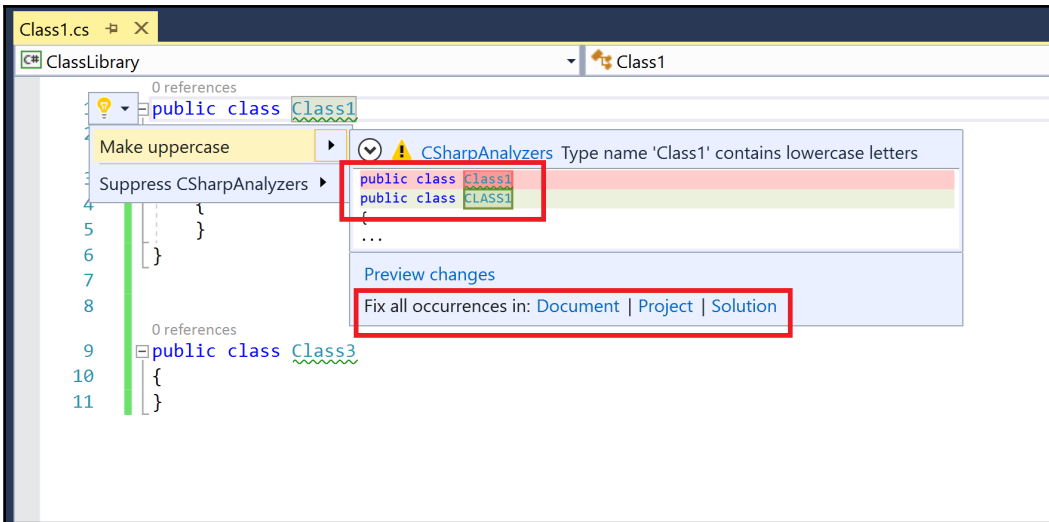
Syntax Visualizer

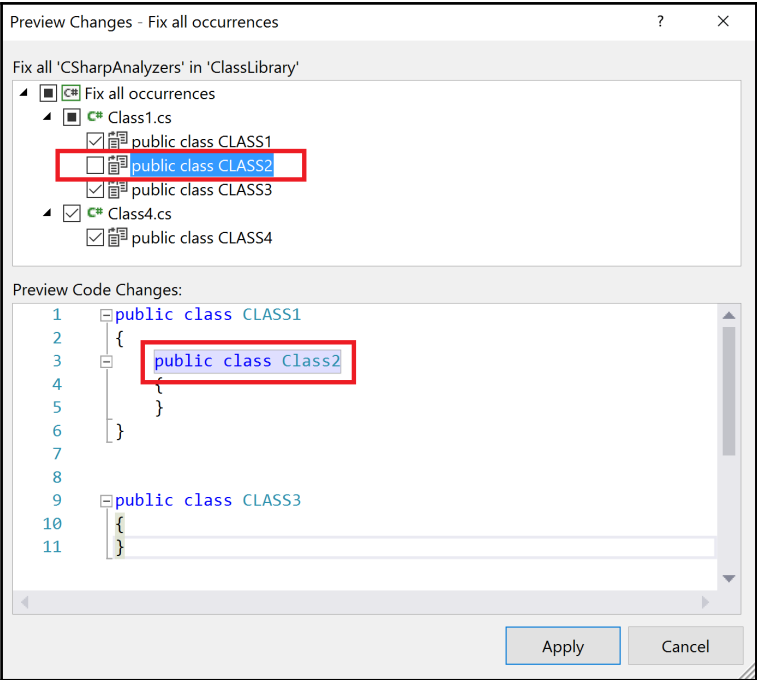
Syntax Tree

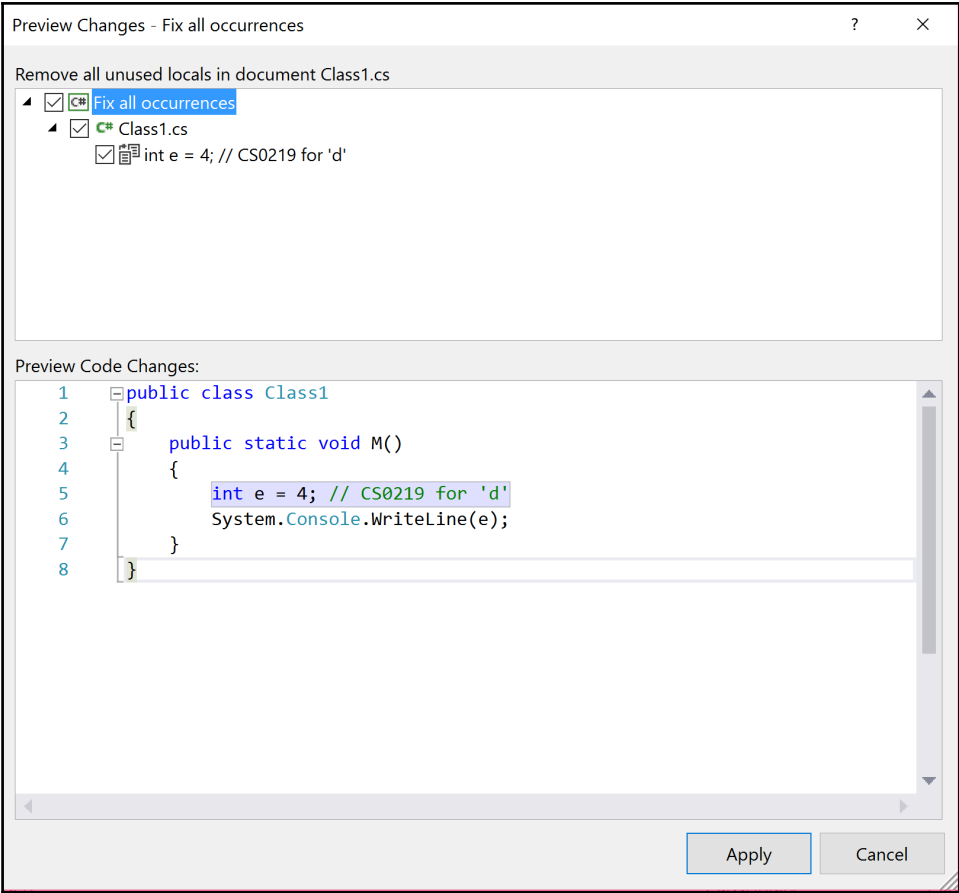
- MethodDeclaration [28..518]
 - PublicKeyword [28..34]
 - PredefinedType [35..39]
 - IdentifierToken [40..47]
 - ParameterList [47..49]
 - Block [55..518]
 - OpenBraceToken [55..56]
 - LocalDeclarationStatement [130..140]
 - LocalDeclarationStatement [236..253]
 - VariableDeclaration [236..252]
 - PredefinedType [236..239]
 - VariableDeclarator [240..245]
 - IdentifierToken [240..241]
 - EqualsValueClause [242..245]
 - CommaToken [245..246]
 - VariableDeclarator [247..252]
 - IdentifierToken [247..252]

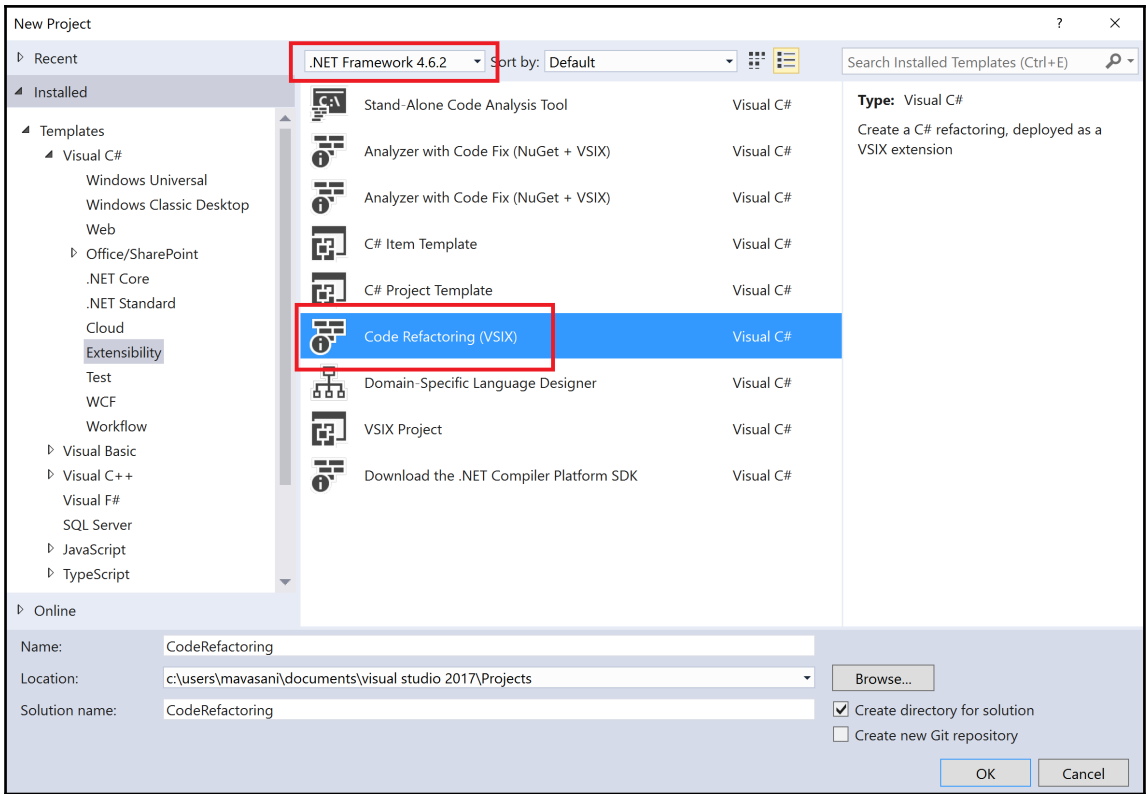
Class1.cs

```
1 public class Class1
2 {
3     0 references
4     public void Method1()
5     {
6         // Local declaration statement with unused local ('a')
7         int a = 0;
8
9         // Local declaration statement with a used ('c') and unused local ('b').
10        int b = 1, c = 2;
11        System.Console.WriteLine(c);
12
13        // Local declaration statement where unused local ('d') initializer is non-constant.
14        int d = c;
15
16        // Local declaration statement with errors ('e').
17        if (true)
18            var e = 1;
19    }
```









NuGet: CodeRefactoring

Browse **Installed** Updates 8

Search (Ctrl+E) Include prerelease

Package source:

Microsoft.CodeAnalysis.CSharp.Workspaces by Microsoft v1.0.1
v1.2.0-beta2

Microsoft.CodeAnalysis.Common by Microsoft v1.0.1
v2.0.0

Microsoft.CodeAnalysis.CSharp by Microsoft v1.0.1
v2.0.0

Microsoft.CodeAnalysis.CSharp.Workspaces by Microsoft v1.0.1
v2.0.0

Each package is licensed to you by its owner. NuGet is not responsible for, nor does it grant any licenses to, third-party packages.

Do not show this again

Microsoft.CodeAnalysis.CSharp.Workspaces

Installed: Uninstall

Version: Update

Options

Description

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

Version: 2.0.0

Author(s): Microsoft

License: <http://go.microsoft.com/fwlink/?LinkId=529443>

Date published: Thursday, March 9, 2017 (3/9/2017)

Project URL: <http://msdn.com/roslyn>

Report Abuse: <https://www.nuget.org/packages/>

C# ClassLibrary Class1 Method

```
1 public class Class1
2 {
3     0 references
4     private int MethodReturningTwoValues(out int x)
5     {
6         ...
7     }
8
9     0 references
10    private int Met
11    {
12        x = 0;
13        z = 1;
14        return y;
15    }
```

Use ValueTuple return type

Change signature...

```
private int MethodReturningTwoValues(out int x)
private (int, int) MethodReturningTwoValues()
{
    int x;
    x = 0;
    return 0;
    return (0, x);
}
```

Preview changes

```
1 public class Class1
2 {
3     private (int, int) MethodReturningTwoValues()
4     {
5         int x;
6         x = 0;
7         return (0, x);
8     }
9
10    private int MethodReturningThreeValues(out int x, int y, out int z)
11    {
12        ...
13    }
14    private (int, int, int) MethodReturningThreeValues(int y)
15    {
16        int x;
17        int z;
18        x = 0;
19        z = 1;
20        return y;
21        return (y, x, z);
22    }
23    ...
24 }
```

this.

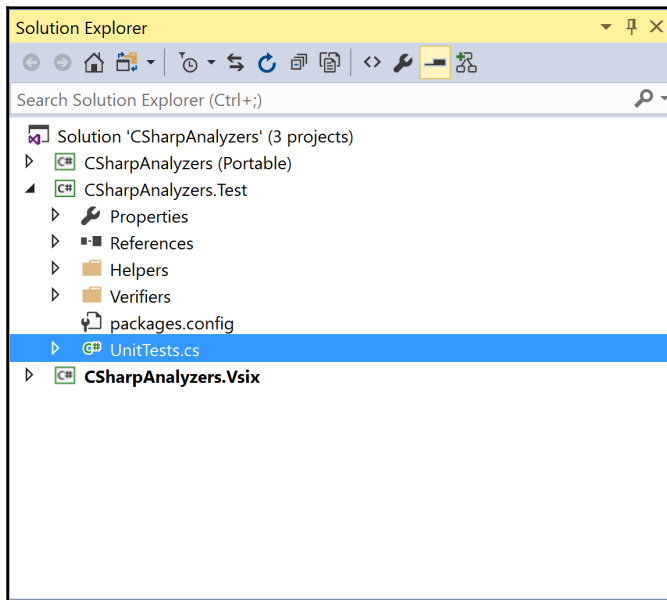
- Equals
- GetHashCode
- GetType
- InstanceMemberBase
- InstanceMemberDerived
- InstanceMethod
- MemberwiseClone
- ToString

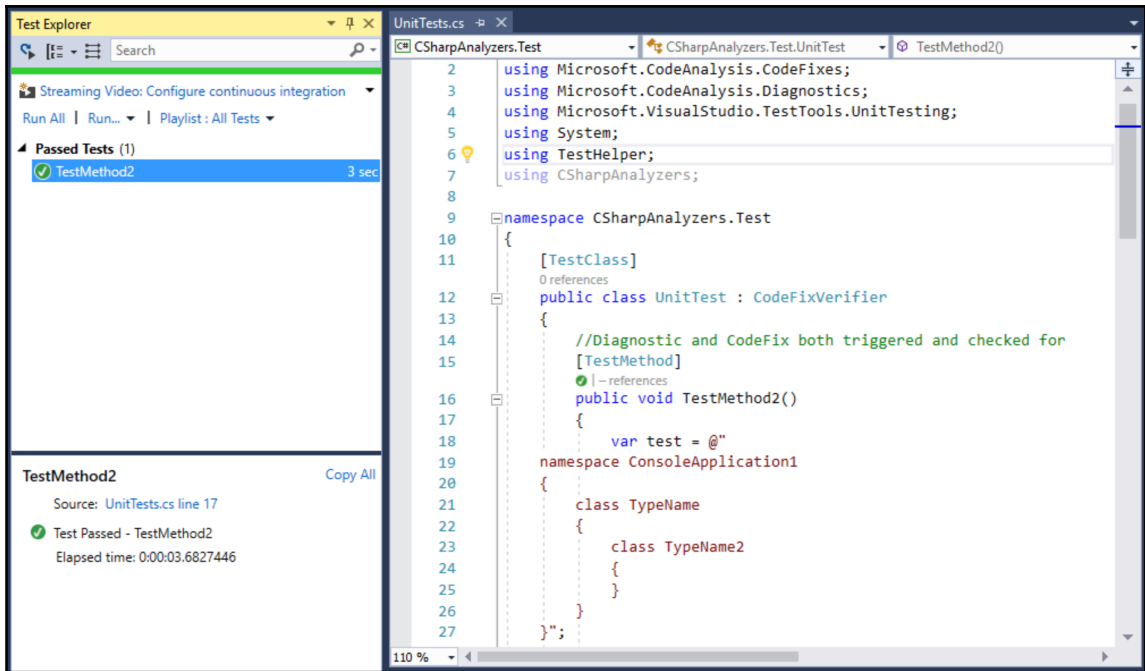
bool object.Equals(object obj)
Determines whether the specified object is equal to the current object.
Note: Tab twice to insert the 'Equals' snippet.

Completion List

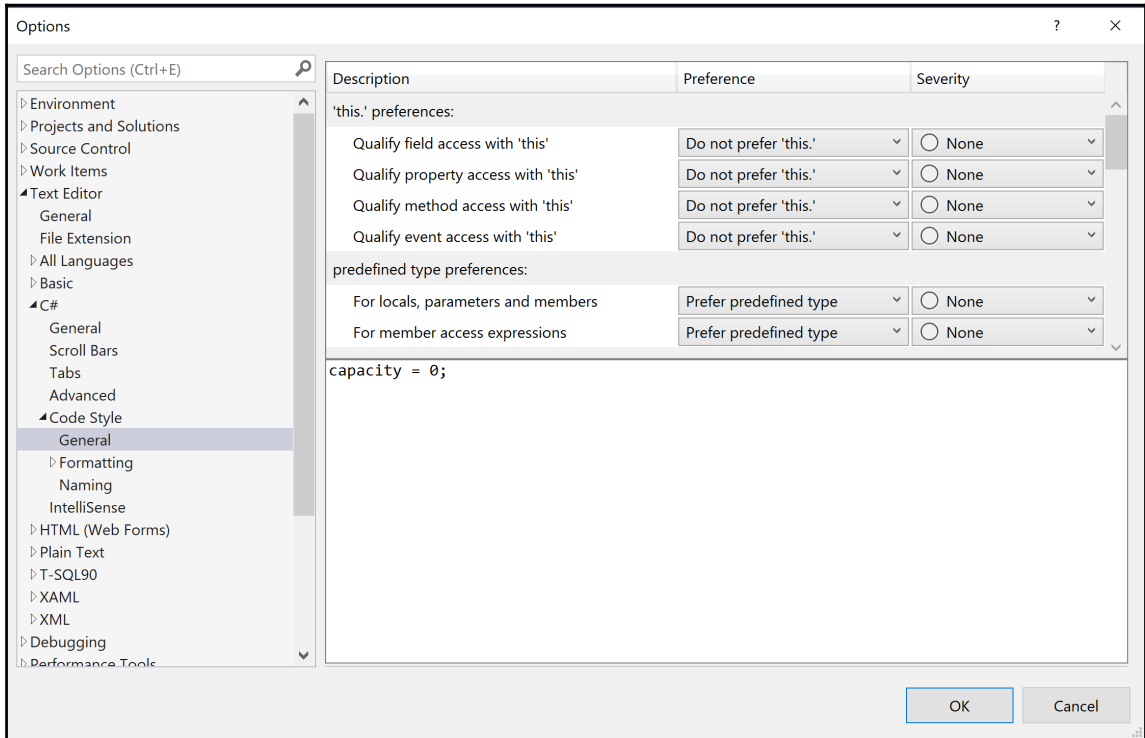
```
0 references
private void InstanceMethod()
{
    .
}
}

Equals bool object.Equals(object obj)
Finalize
GetHashCode
GetType
InstanceMemberBase
InstanceMemberDerived
InstanceMethod
MemberwiseClone
ToString
```





Chapter 4: Improving Code Maintenance of C# Code Base



Description	Preference	Severity
'this.' preferences:		
Qualify field access with 'this'	Do not prefer 'this.'	Suggestion
Qualify property access with 'this'	Do not prefer 'this.'	Suggestion
Qualify method access with 'this'	Do not prefer 'this.'	Suggestion
Qualify event access with 'this'	Do not prefer 'this.'	Suggestion
predefined type preferences:		
For locals, parameters and members	Prefer predefined type	Warning
For member access expressions	Prefer predefined type	Warning
'var' preferences:		
For built-in types	Prefer 'var'	Error
When variable type is apparent	Prefer 'var'	Error
Elsewhere	Prefer 'var'	Error

Description	Preference	Severity
Code block preferences:		
Prefer braces	Yes	Warning
For methods	Prefer expression body	Warning
For constructors	Prefer block body	Warning
For operators	Prefer block body	Warning
For properties	Prefer expression body	Warning
For indexes	Prefer expression body	Warning
For accessors	Prefer expression body	Warning

Error List						
Code	Description	Project	File	Line	Suppression S...	
IDE0007	use 'var' instead of explicit type	ClassLibrary	Class1.cs	18	Active	
IDE0007	use 'var' instead of explicit type	ClassLibrary	Class1.cs	19	Active	
IDE0007	use 'var' instead of explicit type	ClassLibrary	Class1.cs	20	Active	
IDE0012	Name can be simplified.	ClassLibrary	Class1.cs	6	Active	
IDE0012	Name can be simplified.	ClassLibrary	Class1.cs	7	Active	
IDE0013	Name can be simplified.	ClassLibrary	Class1.cs	7	Active	
IDE0011	Add braces to 'if' statement.	ClassLibrary	Class1.cs	26	Active	
IDE0022	Use expression body for methods	ClassLibrary	Class1.cs	32	Active	
IDE0003	Name can be simplified.	ClassLibrary	Class1.cs	11	Active	
IDE0003	Name can be simplified.	ClassLibrary	Class1.cs	12	Active	
IDE0003	Name can be simplified.	ClassLibrary	Class1.cs	13	Active	
IDE0020	Use pattern matching	ClassLibrary	Class1.cs	39	Active	
IDE0018	Variable declaration can be inlined	ClassLibrary	Class1.cs	48	Active	
IDE0031	Null check can be simplified	ClassLibrary	Class1.cs	59	Active	
IDE0029	Null check can be simplified	ClassLibrary	Class1.cs	60	Active	

The screenshot shows a code editor window with the following code snippet:

```

15
16     private int Method_PreferVar()
17     {
18         int i = 0;
19
20     }
21
22
23

```

A warning icon is visible next to line 18. A context menu is open over the code, showing the following options:

- use 'var' instead of explicit type
- Suppress IDE0007

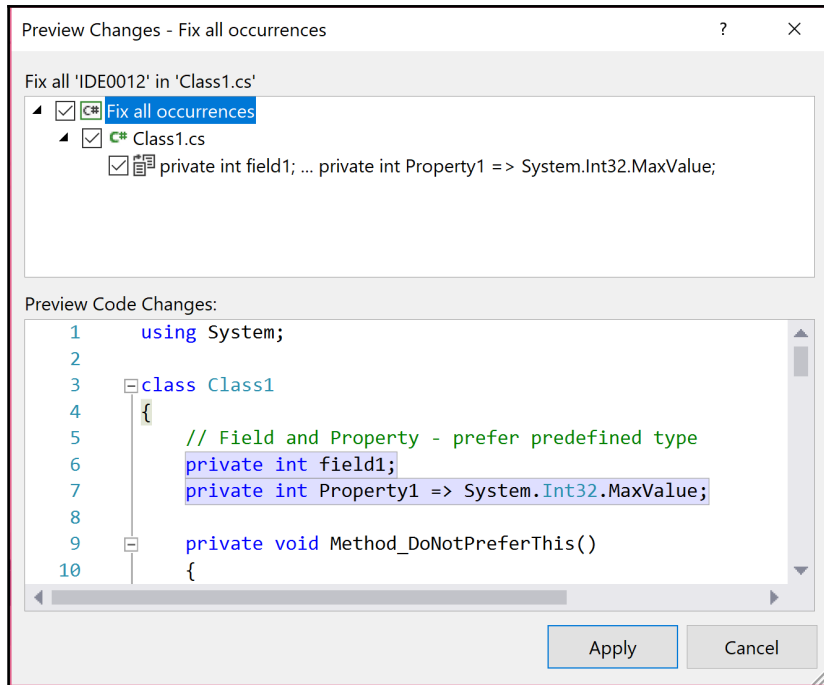
The 'Preview changes' dialog is also visible, showing the code with the following changes:

```

...
int i = 0;
var i = 0;
Class1 c = new Class1();
...

```

The 'Error List' window is also visible at the bottom, showing 3 errors.

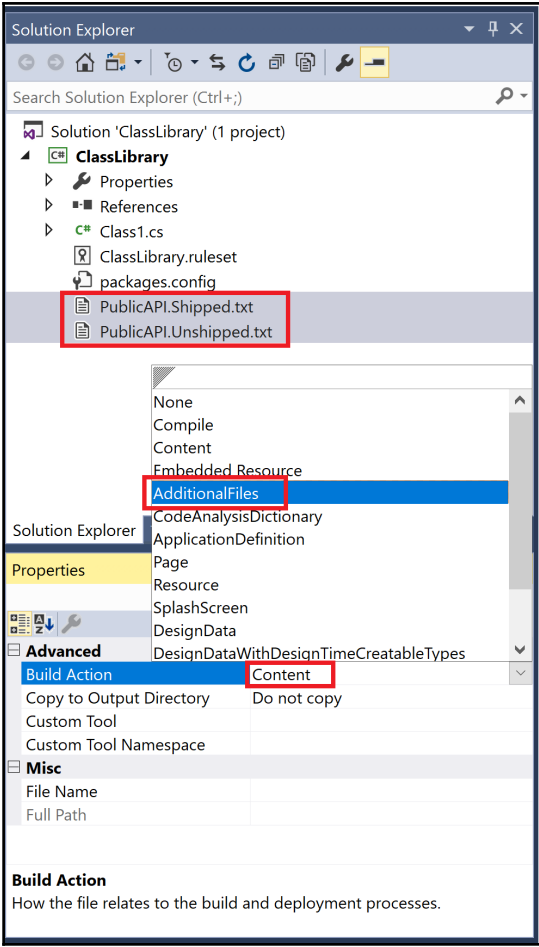


Error List

Entire Solution 3 Errors 5 Warnings 7 Messages Build + IntelliSense Search Error List

Code	Description	Project	File	Line	Suppression S...
IDE0007	use 'var' instead of explicit type	ClassLibrary	Class1.cs	18	Active
IDE0007	use 'var' instead of explicit type	ClassLibrary	Class1.cs	19	Active
IDE0007	use 'var' instead of explicit type	ClassLibrary	Class1.cs	20	Active
IDE0012	Name can be simplified.	ClassLibrary	Class1.cs	6	Active
IDE0012	Name can be simplified.	ClassLibrary	Class1.cs	7	Active
IDE0013	Name can be simplified.	ClassLibrary	Class1.cs	7	Active
IDE0011	Add braces to 'if' statement.	ClassLibrary	Class1.cs	26	Active
IDE0022	Use expression body for methods	ClassLibrary	Class1.cs	32	Active
IDE0003	Name can be simplified.	ClassLibrary	Class1.cs	11	Active
IDE0003	Name can be simplified.	ClassLibrary	Class1.cs	12	Active
IDE0003	Name can be simplified.	ClassLibrary	Class1.cs	13	Active
IDE0020	Use pattern matching	ClassLibrary	Class1.cs	39	Active
IDE0018	Variable declaration can be inlined	ClassLibrary	Class1.cs	48	Active
IDE0031	Null check can be simplified	ClassLibrary	Class1.cs	59	Active
IDE0029	Null check can be simplified	ClassLibrary	Class1.cs	60	Active

Error List Output



PublicAPI.Unshipped.txt PublicAPI.Shipped.txt Class1.cs

```

1 public class Class1
2 {
3     public int Field1;
4
5     public object Property1 => null;
6
7     public void Method1() { }
8
9     public void Method1(int x) { }
10
11     private void Method2() { }
12 }

```

100 %

Error List

Entire Solution 6 Errors 0 Warnings 0 Messages Build + IntelliSense Search Error List

Code	Description	Project	File	Line	Suppression S...
RS0016	Symbol 'Class1' is not part of the declared API.	ClassLibrary	Class1.cs	1	Active
RS0016	Symbol 'implicit constructor for Class1' is not part of the declared API.	ClassLibrary	Class1.cs	1	Active
RS0016	Symbol 'Field1' is not part of the declared API.	ClassLibrary	Class1.cs	3	Active
RS0016	Symbol 'Property1.get' is not part of the declared API.	ClassLibrary	Class1.cs	5	Active
RS0016	Symbol 'Method1' is not part of the declared API.	ClassLibrary	Class1.cs	7	Active
RS0016	Symbol 'Method1' is not part of the declared API.	ClassLibrary	Class1.cs	9	Active

```

1 public class Class1
2 {
3     public int Field1;
4
5     public void Method1() { }
6
7     public void Method1(int x) { }
8
9     private void Method2() { }
10
11
12 }

```

Lightbulb icon

- Add Field1 to public API
- Encapsulate field: 'Field1' (and use property)
- Encapsulate field: 'Field1' (but still use field)
- Suppress RS0016

RS0016 Symbol 'Field1' is not part of the declared API.

Class1.Field1 -> int

Preview changes

Fix all occurrences in: Document | Project | Solution

```

10
11 public void Method2() { }
12

```

Lightbulb icon

- Rename 'Method2' to 'Method3'
- Add Method3 to public API
- Suppress RS0016

RS0016 Symbol 'Method3' is not part of the declared API.

Class1.Method2() -> void

Class1.Method3() -> void

Preview changes

Fix all occurrences in: Document | Project | Solution

Error List						
Entire Solution		0 Errors	12 Warnings	0 of 1 Message	Build + IntelliSense	Search Error List
Code	Description	Project	File	Line	Suppression S...	
SA1025	Code must not contain multiple whitespace characters in a row.	ClassLibrary	Class1.cs	31	Active	
SA1101	Prefix local calls with this	ClassLibrary	Class1.cs	31	Active	
SA1200	Using directive must appear within a namespace declaration	ClassLibrary	Class1.cs	1	Active	
SA1300	Element 'method' must begin with an uppercase letter	ClassLibrary	Class1.cs	29	Active	
SA1303	Const field names must begin with upper-case letter.	ClassLibrary	Class1.cs	27	Active	
SA1401	Field must be private	ClassLibrary	Class1.cs	25	Active	
SA1402	File may only contain a single type	ClassLibrary	Class1.cs	20	Active	
SA1502	Element must not be on a single line	ClassLibrary	Class1.cs	7	Active	
SA1503	Braces must not be omitted	ClassLibrary	Class1.cs	13	Active	
SA1505	An opening brace must not be followed by a blank line.	ClassLibrary	Class1.cs	10	Active	
SA1600	Elements must be documented	ClassLibrary	Class1.cs	5	Active	
SA1633	The file header is missing or not located at the top of the file.	ClassLibrary	Class1.cs	1	Active	

```

29     private int method()
30     {
31         return PublicField;
32     }
33
34

```

Fix spacing

Prefix reference with 'this.'

Suppress SA1025

Suppress SA1101

SA1025 Code must not contain multiple whitespace characters in a row.

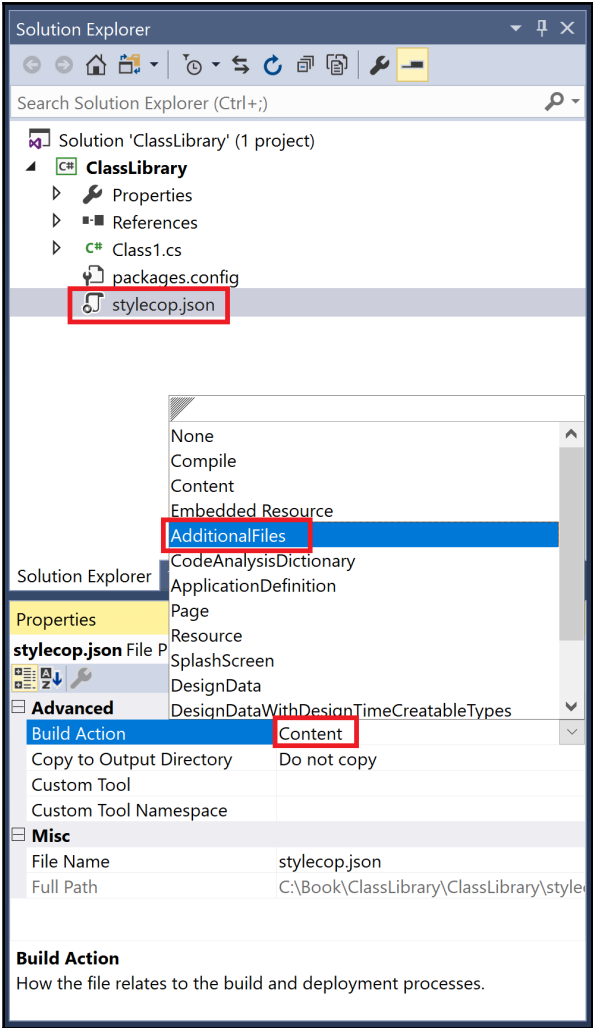
```

...
{
    return PublicField;
    return PublicField;
}
...

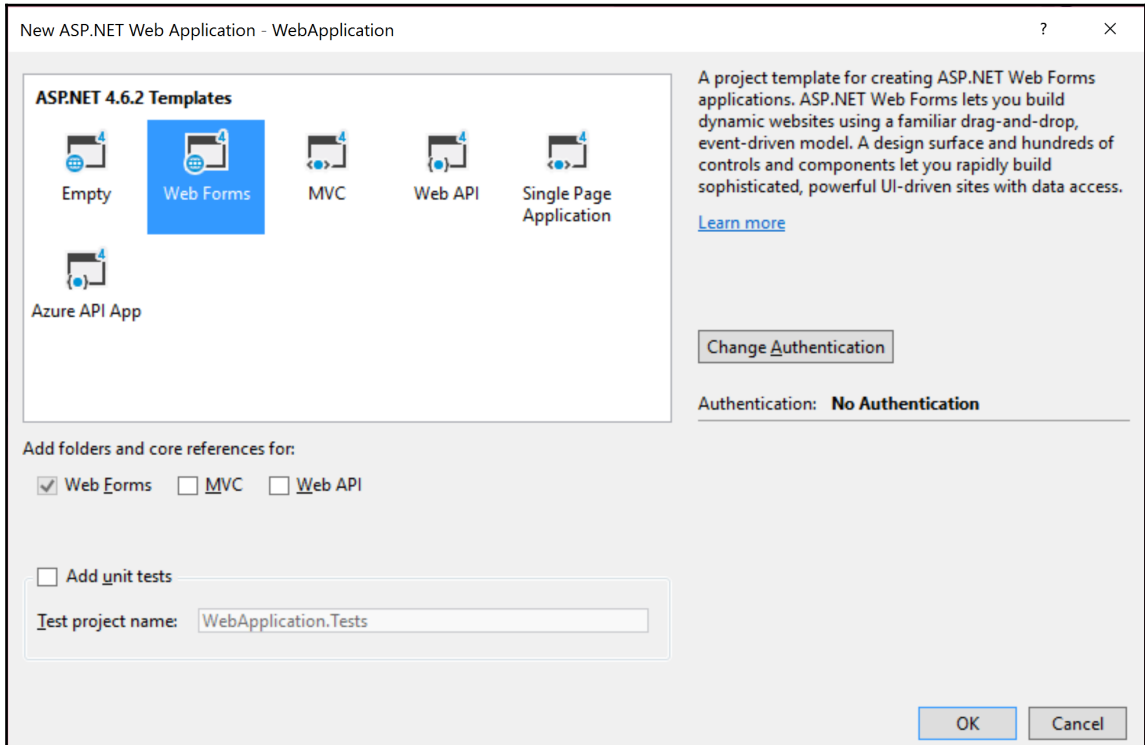
```

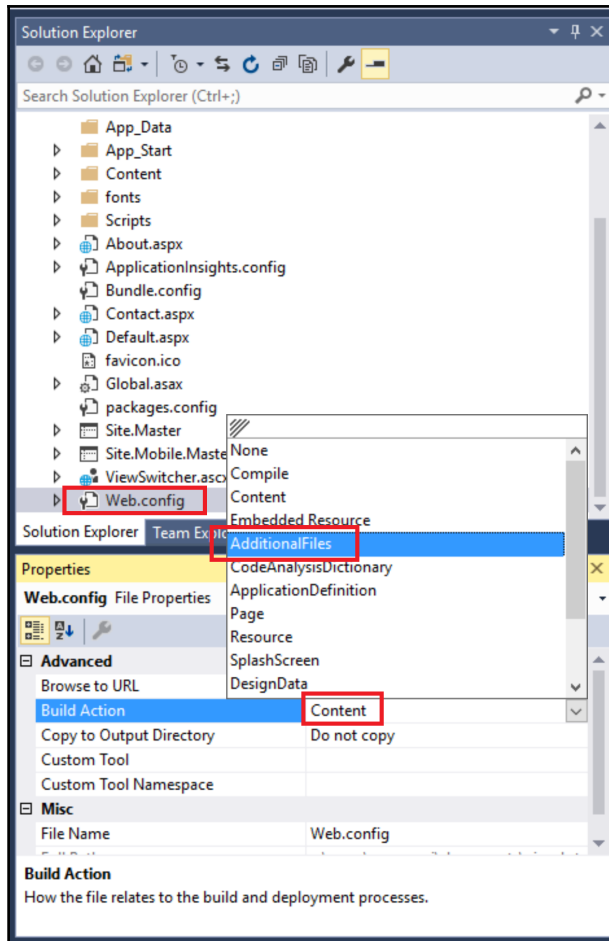
Preview changes

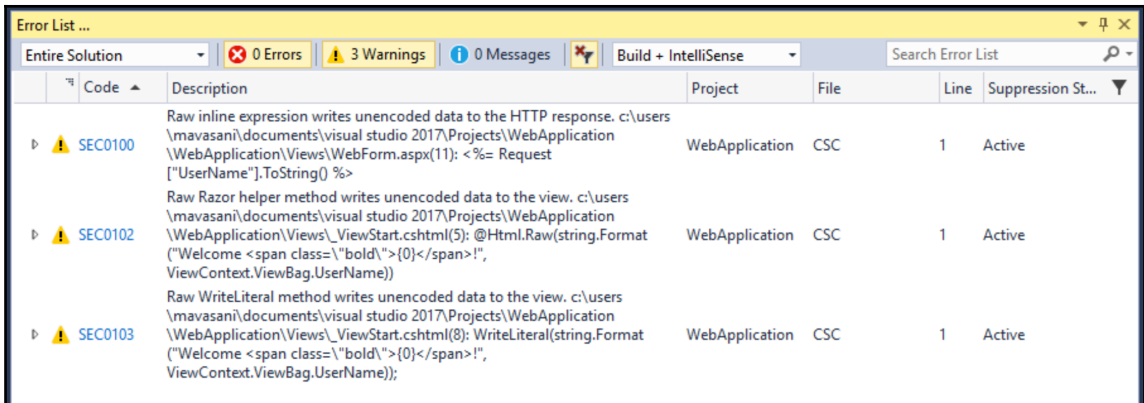
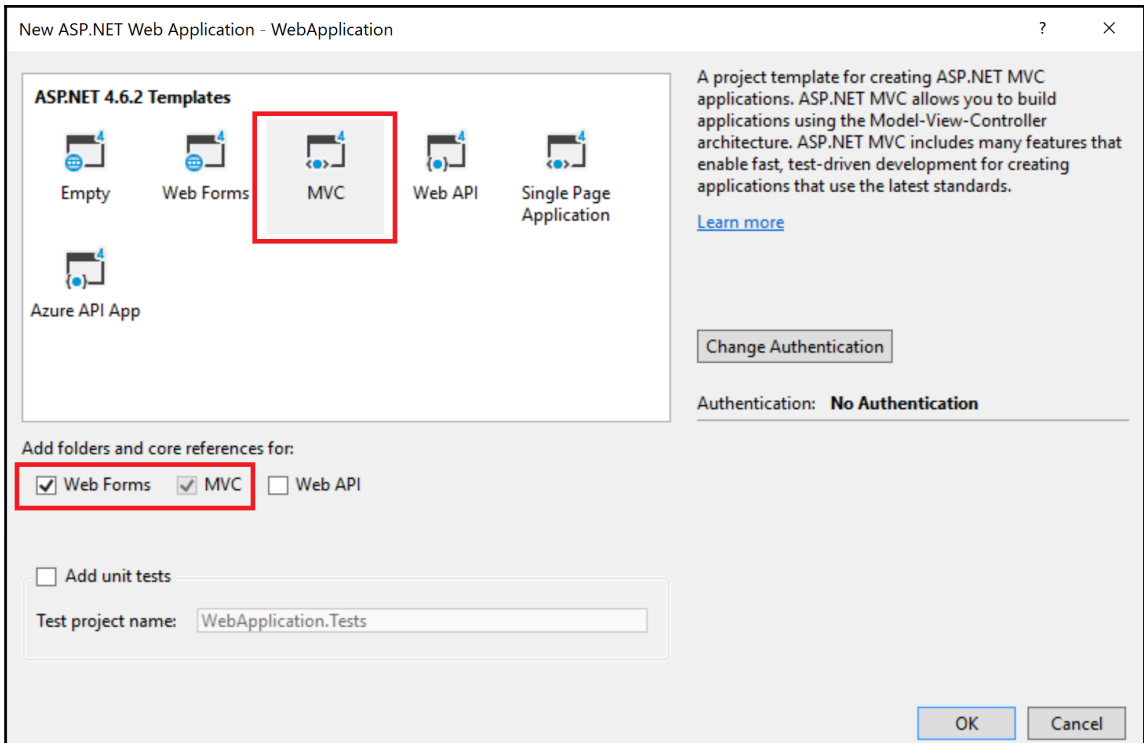
Fix all occurrences in: [Document](#) | [Project](#) | [Solution](#)



Chapter 5: Catch Security Vulnerabilities and Performance Issues in C# Code







Error List ...							
Entire Solution		0 Errors	4 Warnings	0 Messages	Build + IntelliSense	Search Error List	
	Code	Description	Project	File	Line	Suppression St...	
▶	SEC0106	SQL Injection - LINQ ExecuteQuery or ExecuteCommand method executes dynamic SQL without parameters.	ClassLibrary	Class1.cs	11	Active	
▶	SEC0107	SQL Injection - ADO.NET method is passed a dynamic SQL statement.	ClassLibrary	Class1.cs	15	Active	
▶	SEC0114	Untrusted data is passed to the LDAP Path or Filter expression.	ClassLibrary	Class1.cs	20	Active	
▶	SEC0114	Untrusted data is passed to the LDAP Path or Filter expression.	ClassLibrary	Class1.cs	26	Active	

New ASP.NET Web Application - WebApplication

ASP.NET 4.6.2 Templates

Empty

Web Forms

MVC

Web API

Single Page Application

Azure API App

A project template for creating ASP.NET Web Forms applications. ASP.NET Web Forms lets you build dynamic websites using a familiar drag-and-drop, event-driven model. A design surface and hundreds of controls and components let you rapidly build sophisticated, powerful UI-driven sites with data access.

[Learn more](#)

Change Authentication

Authentication: Individual User Accounts

Add folders and core references for:

Web Forms MVC Web API

Add unit tests

Test project name:

OK
Cancel

Error List ...							
Entire Solution		0 Errors	6 of 7 Warnings	0 Messages	Build + IntelliSense	Search Error List	
	C...	Description	Project	File	Line	Suppression St...	
▶	SEC0019	The Action is missing the AntiForgeryToken attribute. If this action modifies data on the backend, it could be vulnerable to Cross-Site Request Forgery attacks.	WebApplication	Class1.cs	14	Active	
▶	SEC0019	The Action is missing the AntiForgeryToken attribute. If this action modifies data on the backend, it could be vulnerable to Cross-Site Request Forgery attacks.	WebApplication	Class1.cs	20	Active	
▶	SEC0022	The AllowHtml attribute disables the validate request security feature.	WebApplication	Class1.cs	9	Active	
▶	SEC0023	Setting the ValidateInput method's first parameter to false disables the validate request security feature.	WebApplication	Class1.cs	13	Active	
▶	SEC0111	Unvalidated file paths are passed to the Controller FilePathResult action, which can allow arbitrary file downloads from the server.	WebApplication	Class1.cs	23	Active	
▶	SEC0113	The ServerCertificateValidationCallback method fails to properly validate the server's certificate.	WebApplication	Class1.cs	30	Active	

Error List ...							
Entire Solution		0 Errors	8 Warnings	0 Messages	Build + IntelliSense	Search Error List	
	Code	Description	Project	File	Line	Suppression St...	
▶	CA1063	Modify Class1.Dispose so that it calls Dispose(true), then calls GC.SuppressFinalize on the current object instance ('this' or 'Me' in Visual Basic), and then returns.	ClassLibrary	Class1.cs	20	Active	
▶	CA1801	Parameter unusedParam of method Method1 is never used. Remove the parameter or use it in the method body.	ClassLibrary	Class1.cs	15	Active	
▶	CA1810	Initialize all static fields in 'Class1' when those fields are declared and remove the explicit static constructor	ClassLibrary	Class1.cs	10	Active	
▶	CA1814	multiDimArray is a multidimensional array. Replace it with a jagged array if possible.	ClassLibrary	Class1.cs	8	Active	
▶	CA1815	ValueType should override Equals.	ClassLibrary	Class1.cs	39	Active	
▶	CA1815	ValueType should override the equality (==) and inequality (!=) operators.	ClassLibrary	Class1.cs	39	Active	
▶	CA1816	Class1.Dispose() calls GC.SuppressFinalize(object) on something other than itself. Change the call site to pass 'this' ('Me' in Visual Basic) instead.	ClassLibrary	Class1.cs	23	Active	
▶	CA1821	Remove empty Finalizers	ClassLibrary	Class1.cs	34	Active	

```

39  public struct ValueType
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

Override equals and operator equals on value types ▶

Generate constructor...

Generate Equals(object)...

Generate Equals and GetHashCode...

Generate overrides...

Extract Interface...

Move type to ValueType.cs

Suppress CA1815 ▶

CA1815 ValueType should override Equals.

```

...
public int Property { get; }

public override bool Equals(object obj)
{
    throw new NotImplementedException();
}

public override int GetHashCode()
{
    throw new NotImplementedException();
}

public static bool operator ==(ValueType left, ValueType right)
{
    throw new NotImplementedException();
}

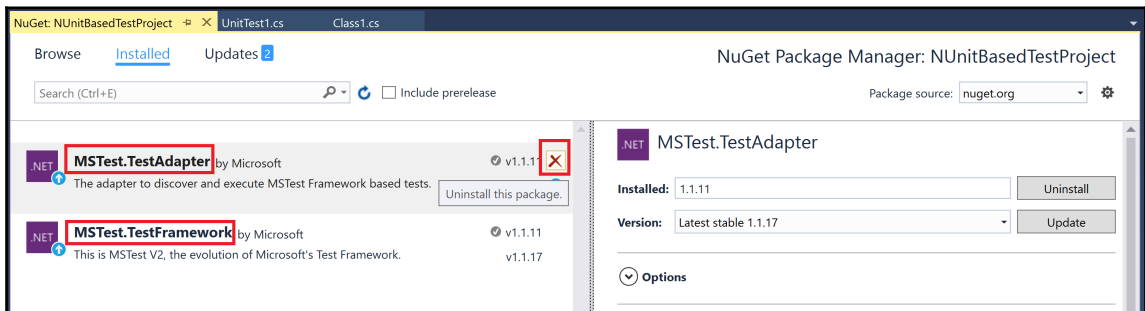
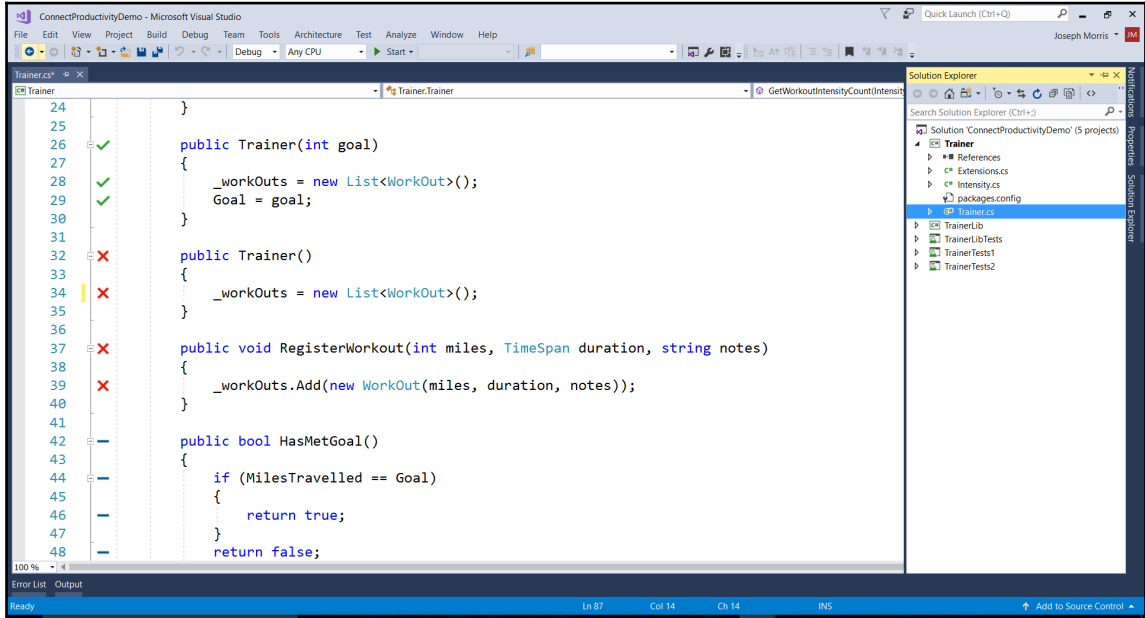
public static bool operator !=(ValueType left, ValueType right)
{
    throw new NotImplementedException();
}
}
...

```

Preview changes

Fix all occurrences in: [Document](#) | [Project](#) | [Solution](#)

Chapter 6: Live Unit Testing in Visual Studio Enterprise



The image shows a Visual Studio IDE window with a C# code file named `UnitTest1.cs`. The code defines a namespace `NUnitBasedTestProject` containing a class `UnitTest1`. Inside `UnitTest1`, there are two test methods:

- `TestMethod1()`: A test method that passes, indicated by a green checkmark in the left margin.
- `TestMethod2()`: A test method that fails, indicated by a red 'X' in the left margin.

The output window at the bottom shows the following log messages:

```
Output
Show output from: Live Unit Testing
[05:52:50.149 Info] Live Unit Testing started.
[05:52:51.337 Info] Build completed (succeeded).
[05:52:52.862 Info] Started executing 2 tests.
[05:52:54.573 Info] Finished executing tests. (Passed: 1, Failed: 1)
```

```
1 namespace ClassLibrary
2 {
3     10 references
4     public class Class1
5     {
6         5 references 3/3 passing
7         public bool Method1()
8         {
9             return true;
10        }
11        5 references 0/3 passing
12        public bool Method2()
13        {
14            return false;
15        }
16        0 references
17        public bool Method3(Class1 c)
18        {
19            return c.Method1();
20        }
21        0 references
22        public bool Method4(Class1 c)
23        {
24            return c.Method2();
25        }
26    }
27 }
```

The screenshot shows a Visual Studio window with a code editor. The code defines a namespace `ClassLibrary` containing a class `Class1` with four methods: `Method1()`, `Method2()`, `Method3(Class1 c)`, and `Method4(Class1 c)`. A test runner overlay is visible on the left side of the code editor, showing the results of unit tests for each method. The test runner shows 3/3 passing for `Method1()` and 0/3 passing for `Method2()`. A red box highlights the test runner icons. The status bar at the bottom shows 100% zoom.

Test Explorer

Class1.cs UnitTest1.cs

UnitTestProject

```

1  using Microsoft.VisualStudio.TestTools.UnitTesting;
2
3  namespace MSTestBasedTestProject
4  {
5      [TestClass]
6      0 references
7      public class UnitTest1
8      {
9          [TestMethod]
10         0 references
11         public void TestMethod1()
12         {
13             var c = new ClassLibrary.Class1();
14             Assert.IsTrue(c.Method1());
15         }
16
17         [TestMethod]
18         0 references
19         public void TestMethod2()
20         {
21             var c = new ClassLibrary.Class1();
22             Assert.IsTrue(c.Method2());
23         }
24     }

```

Failed Tests (1)
 X TestMethod2 18 ms
 Passed Tests (1)
 ✓ TestMethod1 8 ms

Summary

Last Test Run Failed (Total Run Time 0:00:00.0059975)
 X 1 Test Failed
 ✓ 1 Test Passed

Output

Show output from: Live Unit Testing

```

[14:18:55.627 Info] Live Unit Testing started.
[14:18:56.659 Info] Build completed (succeeded).
[14:18:58.434 Info] Started executing 2 tests.
[14:18:59.787 Info] Finished executing tests. (Passed: 1, Failed: 1)

```

Error List Output

Test Explorer

Class1.cs UnitTest1.cs

Test

✓ TestMethod1 8 ms

Run All | Run | Running tests...

Failed Tests (1)
 X TestMethod2 18 ms

Passed Tests (1)
 ✓ TestMethod1 8 ms

2 references 1/1 passing

```

4  {
5      public bool Method1()
6      {
7          return true;
8      }
9  }

```




```
5 | [ ] ✓
Covered by 1 test.
8 | [ ] ✓
2 references | 1/1 passing
public bool Method1()
{
    return true;
}
```



Output

Show output from: Live Unit Testing



Error List Output

Test Explorer

UnitTest2.cs Class2.cs UnitTest1.cs Class1.cs

using Microsoft.VisualStudio.TestTools.UnitTesting;

```

1
2
3 namespace MSTestBasedTestProject
4 {
5     [TestClass]
6     0 references
7     public class UnitTest2
8     {
9         [TestMethod]
10        0 references
11        public void TestMethod5()
12        {
13            var c = new ClassLibrary.Class2();
14            Assert.IsTrue(c.Method5());
15        }
16
17        [TestMethod]
18        0 references
19        public void TestMethod6()
20        {
21            var c = new ClassLibrary.Class2();
22            Assert.IsTrue(c.Method6());
23        }
24    }

```

Failed Tests (2)

- TestMethod5 27 ms
- TestMethod1 32 ms

Passed Tests (2)

- TestMethod6 < 1 ms
- TestMethod2 < 1 ms

TestMethod2 Copy All

Source: UnitTest1.cs line 17

- Test Passed - TestMethod2
- Elapsed time: 0:00:00.0003219

Output

Show output from: Live Unit Testing

```

[16:07:26.353 Info] Build completed (succeeded).
[16:07:27.518 Info] Started executing 2 tests.
[16:07:28.710 Info] Finished executing tests. (Passed: 1, Failed: 1)

```

Test Explorer

UnitTest2.cs Class2.cs UnitTest1.cs Class1.cs

ClassLibrary

```

2 {
3     4 references
4     public class Class1
5     {
6         2 references | 1/1 passing
7         public bool Method1()
8         {
9             return true;
10        }
11
12        2 references | 1/1 passing
13        public bool Method2()
14        {
15            return true;
16        }
17    }

```

Failed Tests (1)

- TestMethod5 25 ms

Passed Tests (3)

- TestMethod1 11 ms
- TestMethod2 < 1 ms
- TestMethod6 < 1 ms

Test Explorer

Streaming Video: Configure continuous...
Run All | Run... | Playlist: All Tests

Passed Tests (2)
✓ TestMethod1 8 ms
✓ TestMethod2 < 1 ms

UnitTest1.cs Class1.cs

ClassLibrary ClassLibrary.Class1

```
1 namespace ClassLibrary
2 {
3     4 references
4     public class Class1
5     {
6         2 references | 1/1 passing
7         public bool Method1()
8         {
9             return true;
10        }
11
12        2 references | 1/1 passing
13        public bool Method2()
14        {
15            return true;
16        }
17    }
18 }
```



Test Explorer

Streaming Video: Configure continuous...
Run All | Run... | Playlist: All Tests

Passed Tests (2)
✓ TestMethod1 8 ms
✓ TestMethod2 < 1 ms

UnitTest1.cs Class1.cs

ClassLibrary ClassLibrary.Class1

```
1 namespace ClassLibrary
2 {
3     4 references
4     public class Class1
5     {
6         2 references | 1/1 passing
7         public bool Method1()
8         {
9             return true;
10        }
11
12        2 references | 1/1 passing
13        public bool Method2()
14        {
15            return true;
16        }
17    }
18 }
```

Test Explorer

Streaming Video: Configure continuous...
Run All | Run... | Playlist: All Tests

Passed Tests (2)

- TestMethod1 8 ms
- TestMethod2 < 1 ms

UnitTest1.cs Class1.cs

ClassLibrary ClassLibrary.Class1

```
1 namespace ClassLibrary
2 {
3     4 references
4     public class Class1
5     {
6         2 references | 1/1 passing
7         public bool Method1()
8         {
9             return false;
10        }
11
12        2 references | 1/1 passing
13        public bool Method2()
14        {
15            return true;
16        }
17    }
18 }
```

Test Explorer

Streaming Video: Configure continuous...
Run All | Run... | Playlist: All Tests

Failed Tests (1)

- TestMethod1 26 ms

Passed Tests (1)

- TestMethod2 < 1 ms

UnitTest1.cs Class1.cs

ClassLibrary ClassLibrary.Class1

```
1 namespace ClassLibrary
2 {
3     4 references
4     public class Class1
5     {
6         2 references | 0/1 passing
7         public bool Method1()
8         {
9             return false;
10        }
11
12        2 references | 1/1 passing
13        public bool Method2()
14        {
15            return true;
16        }
17    }
18 }
```

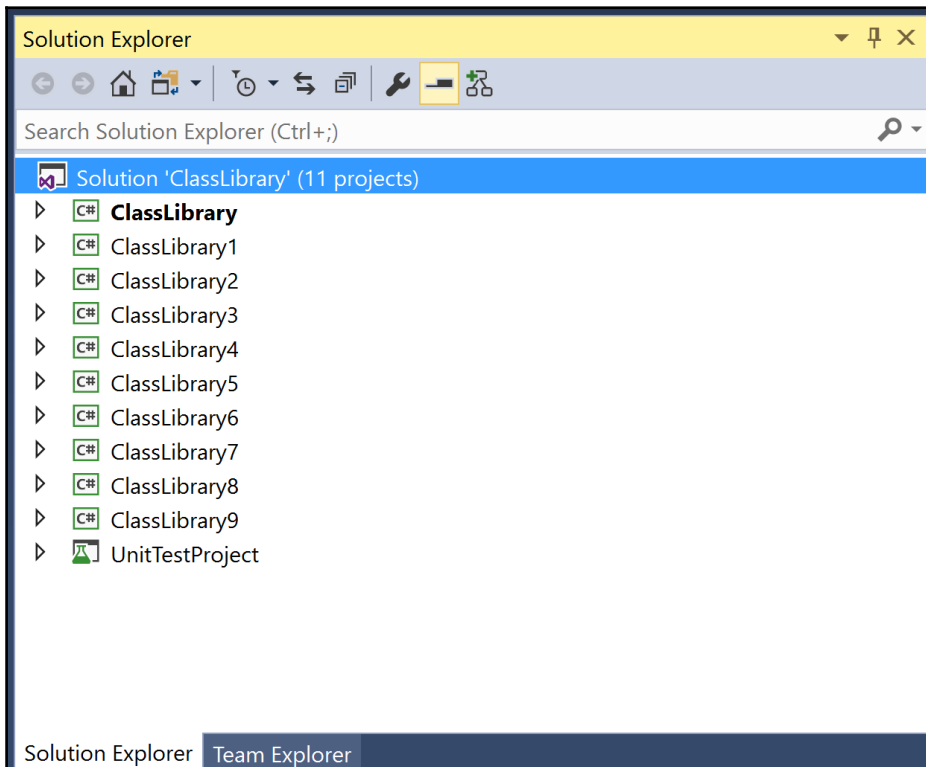
Test Explorer

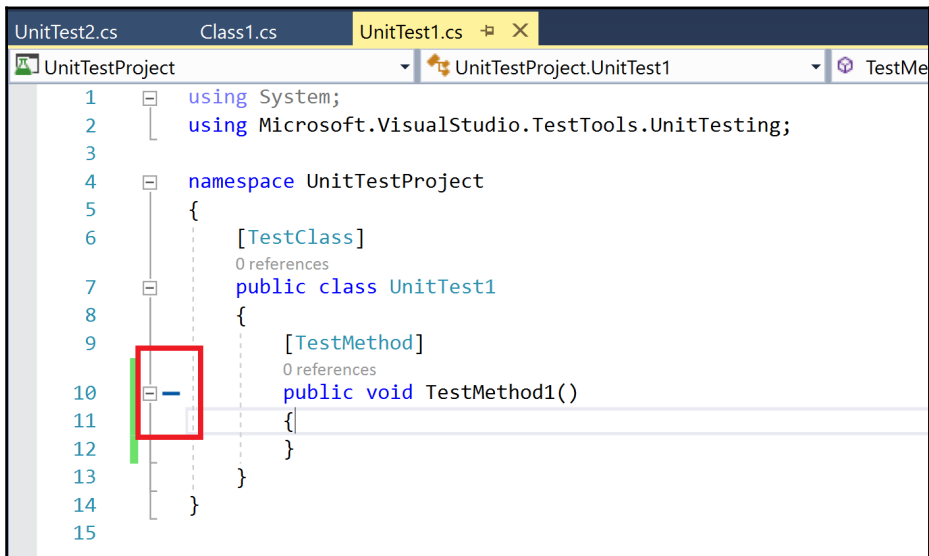
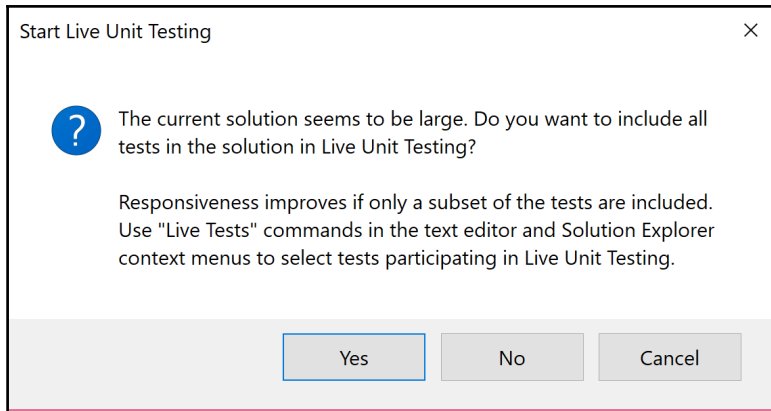
Streaming Video: Configure continuo...
Run All | Run... | Playlist: All Tests
Build your solution to discover all available tests. Click "Run All" to build, discover, and run all tests in your solution.

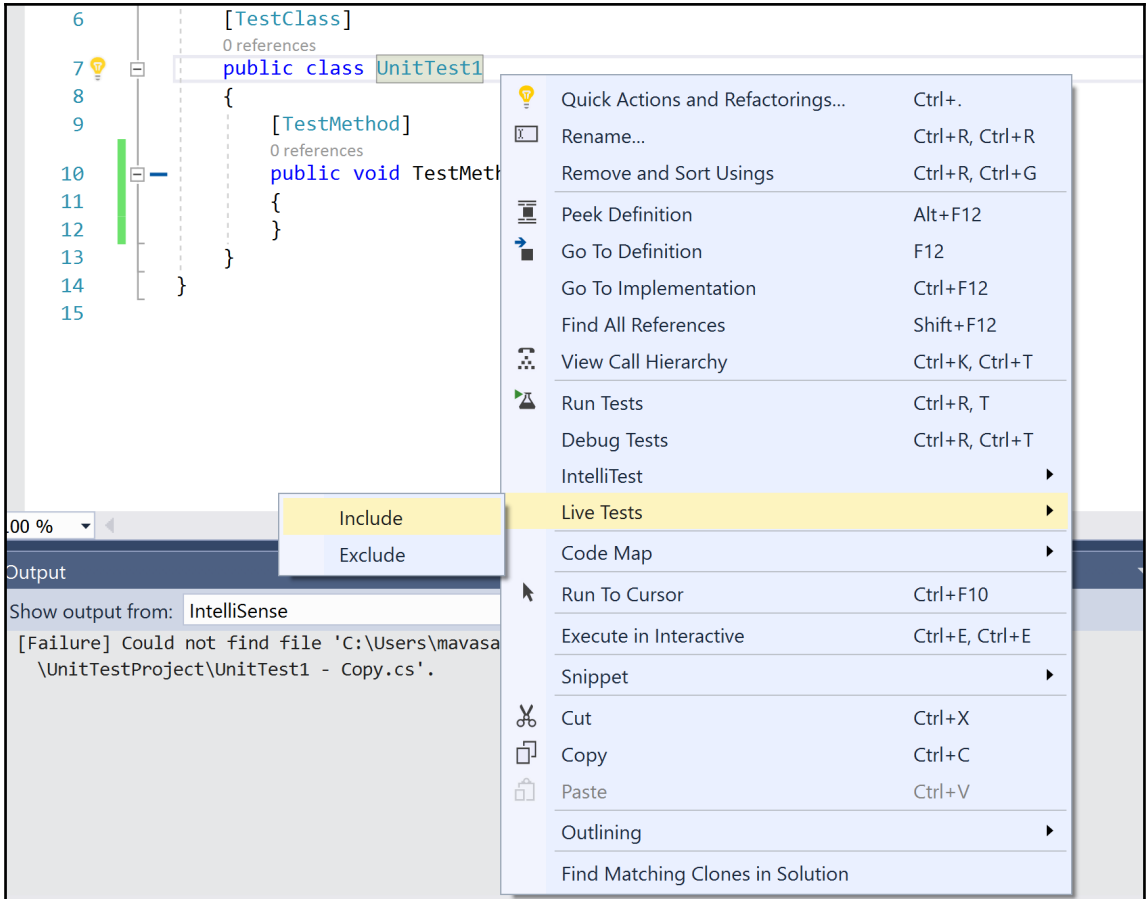
UnitTest1.cs | Class1.cs

C# ClassLibrary | ClassLibrary.Class1

```
1 namespace ClassLibrary
2 {
3     4 references
4     public class Class1
5     {
6         2 references
7         public bool Method1()
8         {
9             return false;
10        }
11        2 references
12        public bool Method2()
13        {
14            return true;
15        }
16    }
17 }
```







The image shows a Visual Studio code editor with a context menu open over the `UnitTest1` class. The code editor displays the following code:

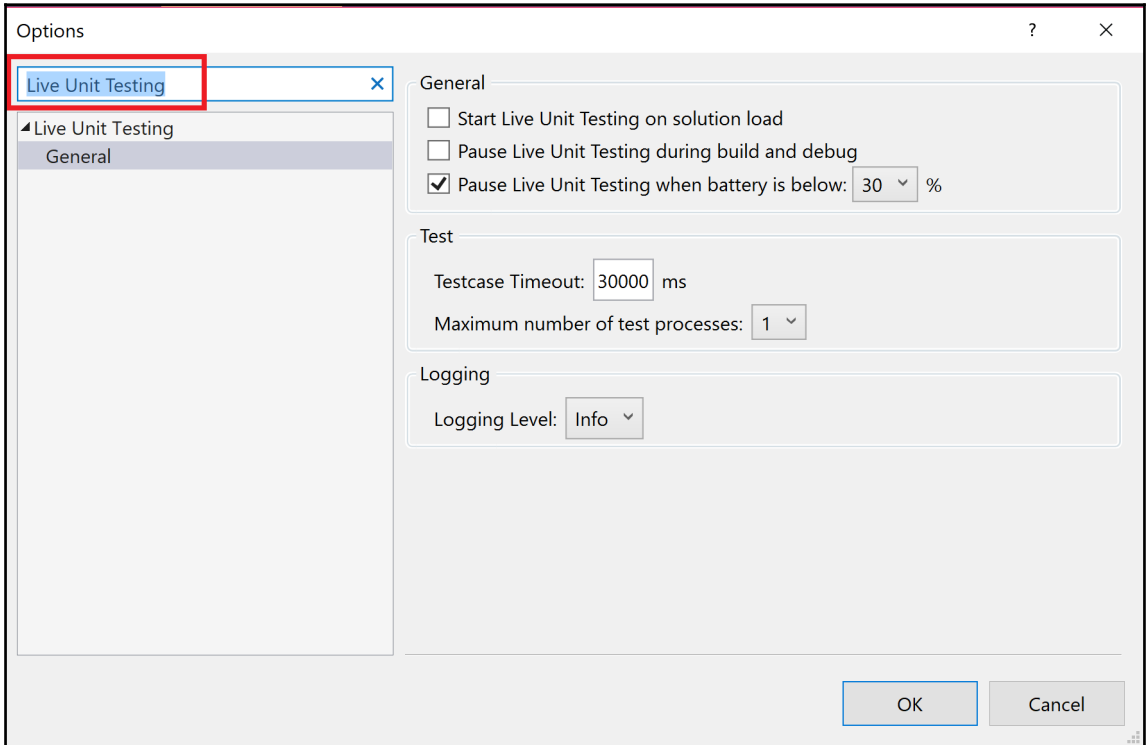
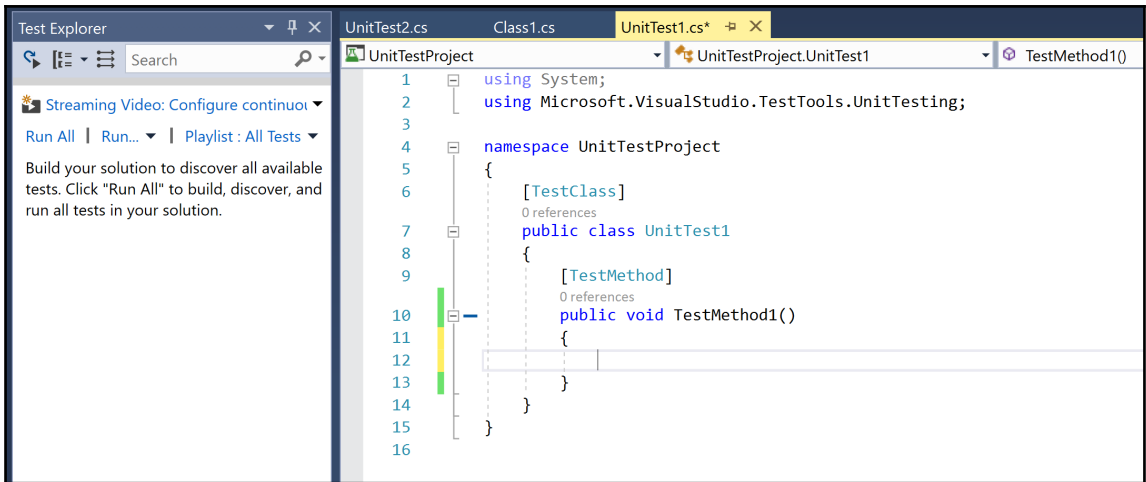
```
6 [TestClass]
7 0 references
8 public class UnitTest1
9 {
10 [TestMethod]
11 0 references
12 public void TestMet
13 {
14 }
15 }
```

The context menu is open, showing the following items:

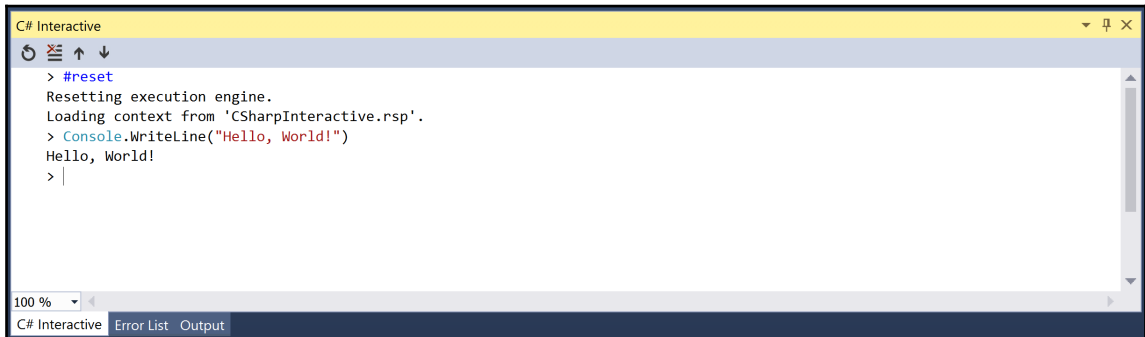
- Quick Actions and Refactorings... (Ctrl+.)
- Rename... (Ctrl+R, Ctrl+R)
- Remove and Sort Usings (Ctrl+R, Ctrl+G)
- Peek Definition (Alt+F12)
- Go To Definition (F12)
- Go To Implementation (Ctrl+F12)
- Find All References (Shift+F12)
- View Call Hierarchy (Ctrl+K, Ctrl+T)
- Run Tests (Ctrl+R, T)
- Debug Tests (Ctrl+R, Ctrl+T)
- IntelliTest
- Live Tests
- Code Map
- Run To Cursor (Ctrl+F10)
- Execute in Interactive (Ctrl+E, Ctrl+E)
- Snippet
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Outlining
- Find Matching Clones in Solution

The `Live Tests` option is highlighted. Below the code editor, the `Output` window is visible, showing the following error message:

```
Show output from: IntelliSense
[Failure] Could not find file 'C:\Users\mavasa\
  \UnitTestProject\UnitTest1 - Copy.cs'.
```



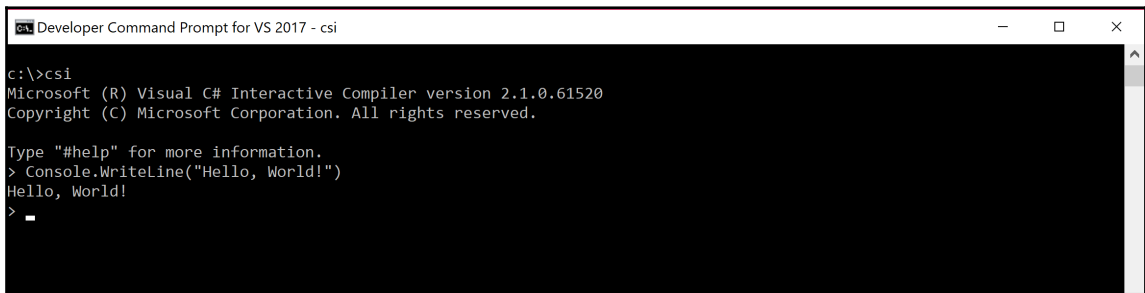
Chapter 7: C# Interactive and Scripting



```
C# Interactive
> #reset
Resetting execution engine.
Loading context from 'CSharpInteractive.rsp'.
> Console.WriteLine("Hello, World!")
Hello, World!
> |
```

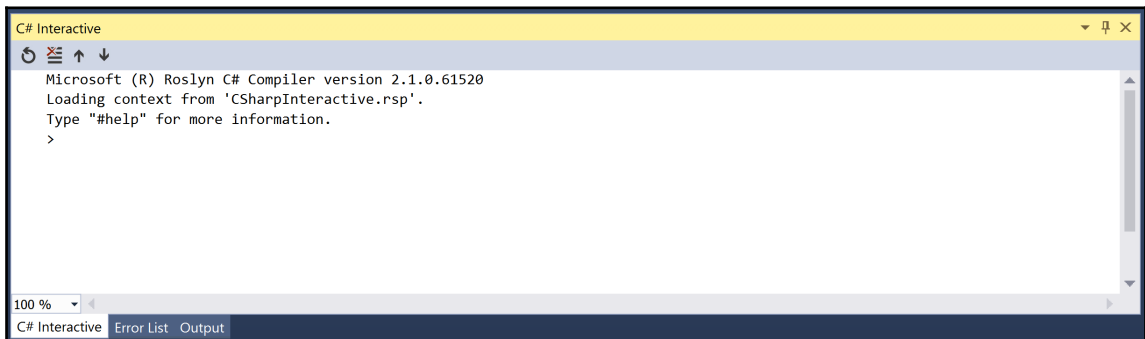
100 %

C# Interactive Error List Output



```
Developer Command Prompt for VS 2017 - csi
c:\>csi
Microsoft (R) Visual C# Interactive Compiler version 2.1.0.61520
Copyright (C) Microsoft Corporation. All rights reserved.

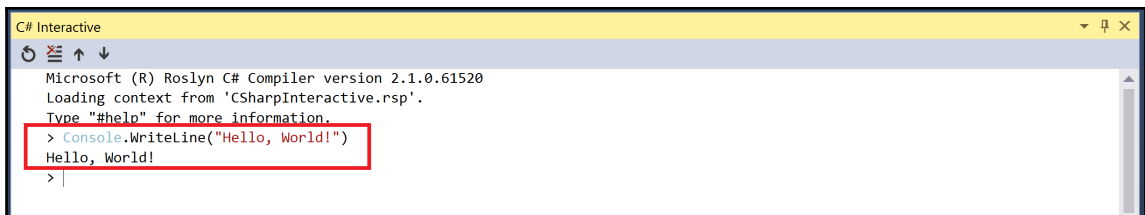
Type "#help" for more information.
> Console.WriteLine("Hello, World!")
Hello, World!
>
> -
```



```
C# Interactive
Microsoft (R) Roslyn C# Compiler version 2.1.0.61520
Loading context from 'CSharpInteractive.rsp'.
Type "#help" for more information.
>
>
```

100 %

C# Interactive Error List Output



```
C# Interactive
Microsoft (R) Roslyn C# Compiler version 2.1.0.61520
Loading context from 'CSharpInteractive.rsp'.
Type "#help" for more information.
> Console.WriteLine("Hello, World!")
Hello, World!
> |
```

```
C# Interactive
Microsoft (R) Roslyn C# Compiler version 2.1.0.61520
Loading context from 'CSharpInteractive.rsp'.
Type "#help" for more information.
> #r "C:\Users\mavasani\AppData\Local\Temp\Newtonsoft.Json.dll"
> using Newtonsoft.Json.Linq;
> JArray array = new JArray();
. array.Add("Manual text");
. array.Add(new DateTime(2000, 5, 23));
.
. JObject o = new JObject();
. o["MyArray"] = array;
.
. o.ToString()
"{\r\n  \"MyArray\": [\r\n    \"Manual text\",\r\n    \"2000-05-23T00:00:00\"\r\n  ]\r\n}"
> |
```

```
C# Interactive
> #load "C:\Users\mavasani\AppData\Local\Temp\MyScript.csx"
Executing MyScript.csx
>
```

```
C# Interactive
> #load "C:\Users\mavasani\AppData\Local\Temp\MyScript.csx"
Executing MyScript.csx
> #reset
Resetting execution engine.
Loading context from 'CSharpInteractive.rsp'.
> using Newtonsoft.Json;
(1,7): error CS0246: The type or namespace name 'Newtonsoft' could not be found (are you missing a using directive or an assembly reference?)
> |
```

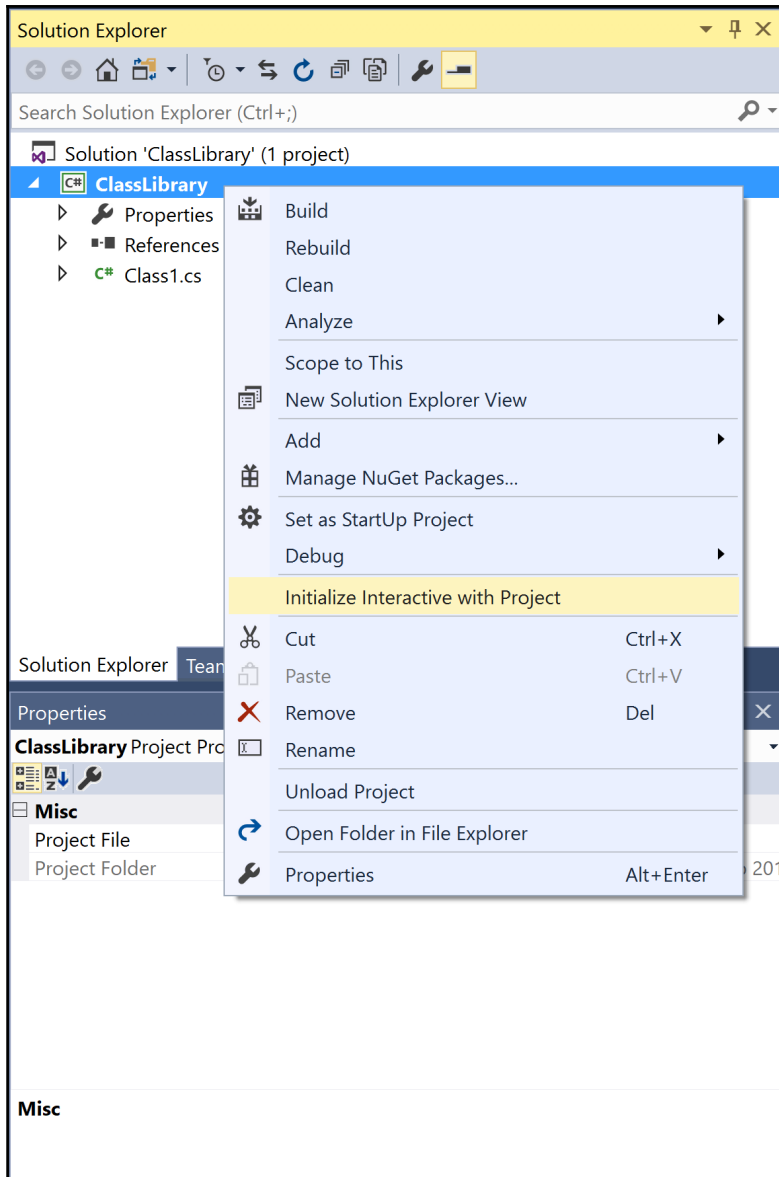


```
C# Interactive
> #help
Keyboard shortcuts:
  Enter           If the current submission appears to be complete, evaluate it. Otherwise, insert a new line.
  Ctrl-Enter      Within the current submission, evaluate the current submission.
                  Within a previous submission, append the previous submission to the current submission.
  Shift-Enter     Insert a new line.
  Escape          Clear the current submission.
  Alt-UpArrow     Replace the current submission with a previous submission.
  Alt-DownArrow   Replace the current submission with a subsequent submission (after having previously navigated backwards).
  Ctrl-Alt-UpArrow Replace the current submission with a previous submission beginning with the same text.
  Ctrl-Alt-DownArrow Replace the current submission with a subsequent submission beginning with the same text (after having previously navigated backwards).
  Ctrl-K, Ctrl-Enter Paste the selection at the end of interactive buffer, leave caret at the end of input.
  Ctrl-E, Ctrl-Enter Paste and execute the selection before any pending input in the interactive buffer.
  Ctrl-A          First press, select the submission containing the cursor. Second press, select all text in the window.

REPL commands:
  #cls, #clear    Clears the contents of the editor window, leaving history and execution context intact.
  #help           Display help on specified command, or all available commands and key bindings if none specified.
  #reset          Reset the execution environment to the initial state, keep history.

Script directives:
  #r              Add a metadata reference to specified assembly and all its dependencies, e.g. #r "myLib.dll".
  #load           Load specified script file and execute it, e.g. #load "myScript.csx".
>
```

```
C# Interactive
Microsoft (R) Roslyn C# Compiler version 2.1.0.61520
Loading context from 'CSharpInteractive.rsp'.
Type "#help" for more information.
> "World!"
"World!"
> "Hello, " + "World!"
"Hello, World!"
> @"Hello, World
  . with a new line!"
"Hello, World\r\nwith a new line!"
> |
```



```
C# Interactive
Microsoft (R) Roslyn C# Compiler version 2.1.0.61520
Loading context from 'CSharpInteractive.rsp'.
Type "#help" for more information.
> int x = 0;
> Console.WriteLine(x);
0
> #reset
Resetting execution engine.
Loading context from 'CSharpInteractive.rsp'.
> #r "C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.6\System.dll"
> #r "C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.6\System.Core.dll"
> #r "C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.6\System.Xml.Linq.dll"
> #r "C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.6\System.Data.DataSetExtensions.dll"
> #r "C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.6\Microsoft.CSharp.dll"
> #r "C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.6\System.Data.dll"
> #r "C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.6\System.Net.Http.dll"
> #r "C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.6\System.Xml.dll"
> #r "ClassLibrary.dll"
> using ClassLibrary;
>
```

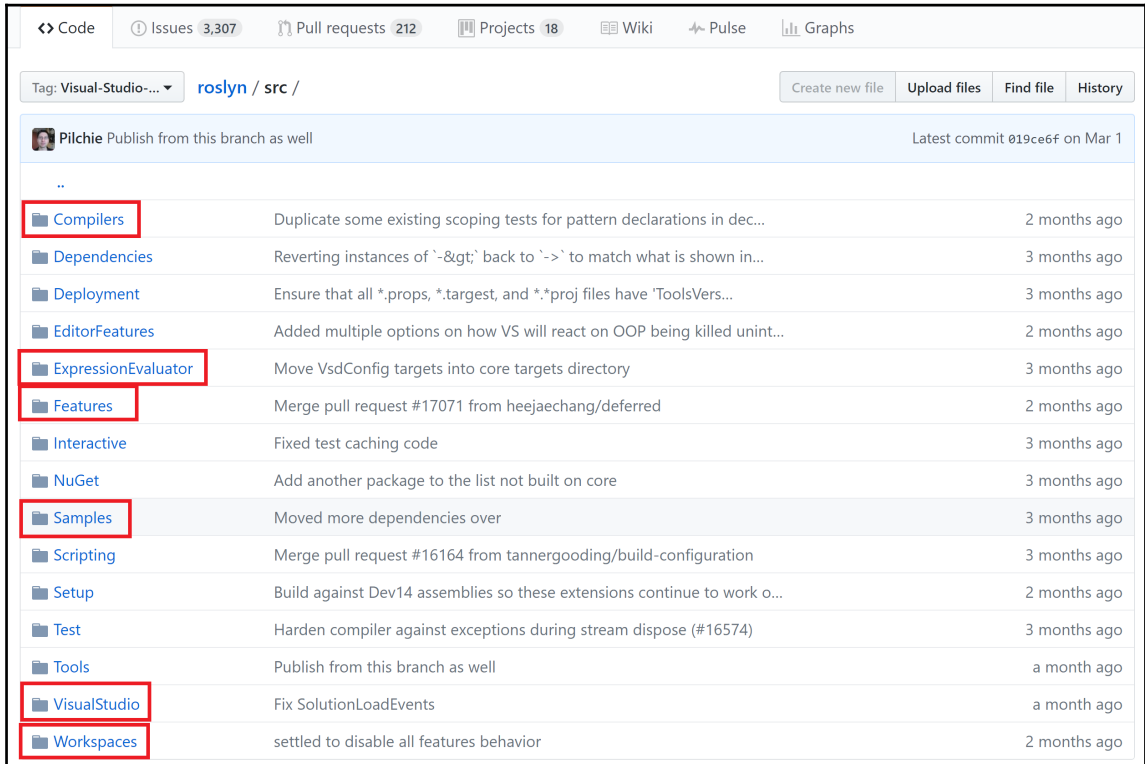
```
Administrator: Developer Command Prompt for VS 2017 - cs.exe
c:\>csi.exe
Microsoft (R) Visual C# Interactive Compiler version 2.1.0.61520
Copyright (C) Microsoft Corporation. All rights reserved.

Type "#help" for more information.
> Console.WriteLine("Hello, World!")
Hello, World!
>
```

```
Administrator: Developer Command Prompt for VS 2017 - csi -i
c:\>csi -i
Microsoft (R) Visual C# Interactive Compiler version 2.1.0.61520
Copyright (C) Microsoft Corporation. All rights reserved.

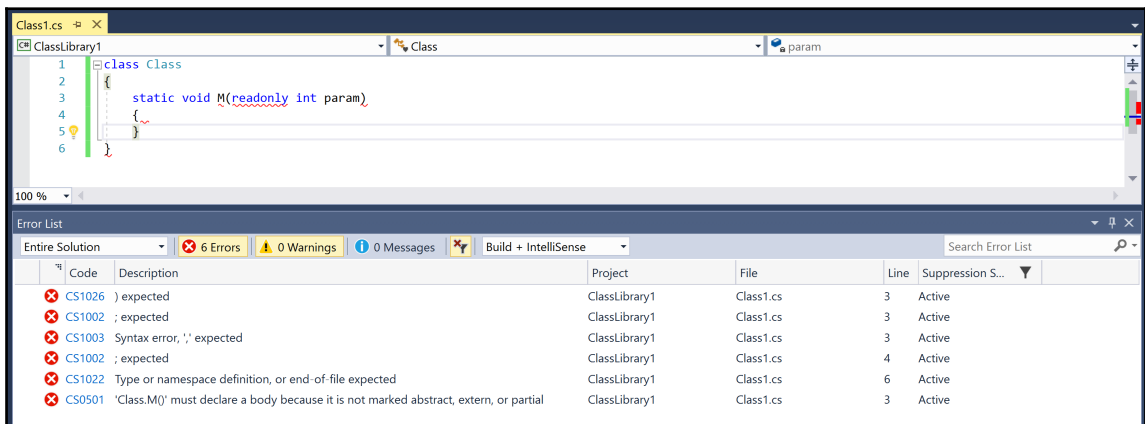
Type "#help" for more information.
> #help
Keyboard shortcuts:
  Enter      If the current submission appears to be complete, evaluate it.  Otherwise, insert a new line.
  Escape     Clear the current submission.
  UpArrow    Replace the current submission with a previous submission.
  DownArrow  Replace the current submission with a subsequent submission (after having previously navigated backwards
).
  Ctrl-C     Exit the REPL.
REPL commands:
  #help      Display help on available commands and key bindings.
Script directives:
  #r         Add a metadata reference to specified assembly and all its dependencies, e.g. #r "myLib.dll".
  #load      Load specified script file and execute it, e.g. #load "myScript.csx".
>
```

Chapter 8: Contribute Simple Functionality to Roslyn C# Compiler Open Source Code



The screenshot shows the GitHub repository for Roslyn C# Compiler. The repository path is `roslyn / src /`. The commit history is displayed as a table with folders highlighted by red boxes:

Folder	Description	Time
Compilers	Duplicate some existing scoping tests for pattern declarations in dec...	2 months ago
Dependencies	Reverting instances of <code>'-&gt;'</code> back to <code>'->'</code> to match what is shown in...	3 months ago
Deployment	Ensure that all *.props, *.target, and *.proj files have ToolsVers...	3 months ago
EditorFeatures	Added multiple options on how VS will react on OOP being killed uninit...	2 months ago
ExpressionEvaluator	Move VsdConfig targets into core targets directory	3 months ago
Features	Merge pull request #17071 from heejaechang/deferred	2 months ago
Interactive	Fixed test caching code	3 months ago
NuGet	Add another package to the list not built on core	3 months ago
Samples	Moved more dependencies over	3 months ago
Scripting	Merge pull request #16164 from tannergooding/build-configuration	3 months ago
Setup	Build against Dev14 assemblies so these extensions continue to work o...	2 months ago
Test	Harden compiler against exceptions during stream dispose (#16574)	3 months ago
Tools	Publish from this branch as well	a month ago
VisualStudio	Fix SolutionLoadEvents	a month ago
Workspaces	settled to disable all features behavior	2 months ago



The screenshot shows a Visual Studio editor window with a C# code file named `Class1.cs`. The code is as follows:

```
1 class Class
2
3     static void M(readonly int param),
4
5
6 }
```

The Error List at the bottom shows 6 errors:

Code	Description	Project	File	Line	Suppression S...
CS1026) expected	ClassLibrary1	Class1.cs	3	Active
CS1002	; expected	ClassLibrary1	Class1.cs	3	Active
CS1003	Syntax error, ',' expected	ClassLibrary1	Class1.cs	3	Active
CS1002	; expected	ClassLibrary1	Class1.cs	4	Active
CS1022	Type or namespace definition, or end-of-file expected	ClassLibrary1	Class1.cs	6	Active
CS0501	'Class.M()' must declare a body because it is not marked abstract, extern, or partial	ClassLibrary1	Class1.cs	3	Active

Class1.cs x

ClassLibrary2 Class M(int param)

```

1 class Class
2 {
3     static void M(readonly int param)
4     {
5     }
6 }

```

100 %

Error List

Entire Solution 1 Error 0 Warnings 0 Messages Build + IntelliSense Search Error List

Code	Description	Project	File	Line	Suppression S...
CS0106	The modifier 'readonly' is not valid for this item	ClassLibrary2	Class1.cs	3	Active

Class1.cs x

ClassLibrary2 Class

```

1 class Class
2 {
3     int X { get; set; }
4
5     void M(int x)
6     {
7         X = X;
8         var y = X + "" ;
9     }
10 }

```

100 %

Error List

Entire Solution 0 Errors 2 Warnings 0 Messages Build + IntelliSense Search Error List

Code	Description	Project	File	Line	Suppression S...
CS1717	Assignment made to same variable; did you mean to assign something else?	ClassLibrary2	Class1.cs	7	Active
CS0823	Use an explicit type for declaration as the initializer type 'string' is not apparent due to conversions	ClassLibrary2	Class1.cs	8	Active

ImplicitlyTypedLocalsTests.cs x

CSharpCompilerSemanticTest Microsoft.CodeAnalysis.CSharp.UnitTests.ImplicitlyTypedLocalTests VarInferredTypeNotApparent()

```

154
155 [Fact]
156 public void VarInferredTypeNotApparent()
157 {
158     var source = @"
159     class Class
160     {
161     void M(int x, string y)
162     {
163     var z = x + y;
164     }
165     }
166
167     CreateCompilationWithMscorlib(source).VerifyDiagnostics(
168         // (6,7): warning CS0823: Use an explicit type for declaration as the initializer type 'string' is not apparent due to conversions
169         // var z = x + y;
170         Diagnostic(ErrorCode.WRN_ImplicitlyTypedVariableNotRecommended, "z = x + y").WithArguments("string").WithLocation(6, 7));
171 }
172
173

```

100 %

The screenshot shows the Visual Studio Syntax Visualizer interface. On the left, the 'Syntax Tree' pane displays a tree structure for the code snippet. The 'InvocationExpression' node is highlighted with a red box. Below it, the 'Properties' pane shows details for the 'InvocationExpressionSyntax' node, with 'Type' and 'Kind' also highlighted in red. On the right, the 'Class1.cs' editor shows the source code with the same line highlighted in red. At the bottom, the 'Error List' pane is visible, showing '0 Errors' and '0 Warnings'.

Syntax Tree

- CompilationUnit [0..186]
 - UsingDirective [0..13]
 - NamespaceDeclaration [17..184]
 - NamespaceKeyword [17..26]
 - IdentifierName [27..39]
 - OpenBraceToken [41..42]
 - ClassDeclaration [48..181]
 - PublicKeyWord [48..54]
 - ClassKeyword [55..60]
 - IdentifierToken [61..67]
 - OpenBraceToken [73..74]
 - MethodDeclaration [84..174]
 - PublicKeyWord [84..90]
 - PredefinedType [91..95]
 - IdentifierToken [96..102]
 - ParameterList [102..104]
 - Block [114..174]
 - OpenBraceToken [114..115]
 - ExpressionStatement [129..163]
 - InvocationExpression [129..162]

Properties

Type	InvocationExpressionSyntax
Kind	InvocationExpression
ContainsDiagnostics	False
ContainsDirectives	False
ContainsSkippedText	False
Expression	Console.WriteLine
FullSpan	[117..162]
HasLeadingTrivia	True
HasStructuredTrivia	False
HasTrailingTrivia	False
IsMissing	False
IsStructuredTrivia	False
Language	C#
Parent	Console.WriteLine("Hello world!");
ParentTrivia	

Class1.cs

```

1 using System;
2
3 namespace ClassLibrary
4 {
5     public class Class1
6     {
7         public void Method()
8         {
9             Console.WriteLine("Hello world!");
10        }
11    }
12 }
13
  
```

Error List

0 Errors 0 Warnings 0 Messages

Code	Description	Project
------	-------------	---------

- ExpressionStatement [129..163]
 - InvocationExpression [129..162]
 - View Symbol (if any)
 - View TypeSymbol (if any)
 - View Converted TypeSymbol (if any)
 - View AliasSymbol (if any)
 - View Constant Value (if any)

Properties

Type	PEMethodSymbol
Kind	Method
CanBeReferencedByN	True
ConstructedFrom	System.Console.WriteLine(string)
ContainingAssembly	mscorlib, Version=4.0.0.0, Culture=neutral, Public
ContainingNamespace	System
ContainingSymbol	System.Console
ContainingType	System.Console
DeclaredAccessibility	Public
DeclaringSyntaxRefer	(Collection)
ExplicitInterfacemple	(Collection)
HasUnsupportedMet	False
HidesBaseMethodsBy	False
IsAbstract	False
IsAsync	False
IsCheckedBuiltin	False
IsDefinition	True
IsExtensionMethod	False
IsExtern	False
IsGenericMethod	False
IsImplicitlyDeclared	False
IsOverride	False
IsSealed	False
IsStatic	True
IsTupleMethod	False
IsVararg	False
IsVirtual	False
Kind	Method
Language	C#
Locations	(Collection)
MetadataName	WriteLine
MethodKind	Ordinary
Name	WriteLine
OriginalDefinition	System.Console.WriteLine(string)
OverriddenMethod	

The image shows the Visual Studio IDE with the Syntax Visualizer tool open. The Syntax Tree view on the left shows the following structure:

- CompilationUnit [0..109]
 - ClassBlock [0..107]
 - ClassStatement [0..19]
 - SubBlock [25..96]
 - SubStatement [25..44]
 - ExpressionStatement [54..83]
 - InvocationExpressionSyntax [54..83]**
 - SimpleMemberAccessExpression [54..67]
 - ArgumentList [67..83]

The Properties window for the selected **InvocationExpressionSyntax** node shows the following details:

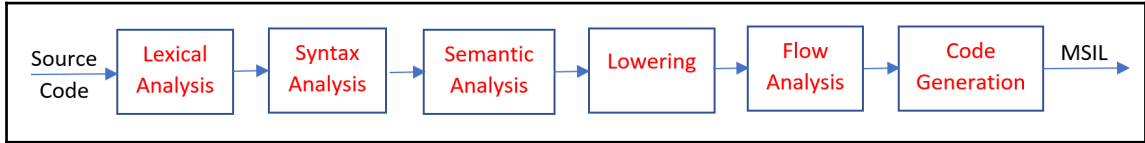
Type	InvocationExpressionSyntax
Kind	InvocationExpression
ArgumentList	("Hello World!")
ContainsAnnotations	False
ContainsDiagnostics	False
ContainsDirectives	False
ContainsSkippedText	False
Expression	Console.WriteLine
FullSpan	[46..85]
HasLeadingTrivia	True
HasStructuredTrivia	False
HasTrailingTrivia	True
IsDirective	False
IsMissing	False
IsStructuredTrivia	False
Language	Visual Basic
Parent	Console.WriteLine("Hello World!")
ParentTrivia	
RawKind	296
Span	[54..83]
SpanStart	54
SyntaxTree	Public Class Class1 Public Sub Method()

The main editor window shows the following code:

```
1 Public Class Class1
2     Public Sub Method()
3         Console.WriteLine("Hello World!")
4     End Sub
5 End Class
6
```

The Error List at the bottom is empty, showing 0 Errors, 0 Warnings, and 0 Messages.

Chapter 9: Design and Implement a New C# Language Feature



The screenshot shows the Syntax Visualizer tool with a syntax tree on the left and the source code on the right. The source code is as follows:

```
1 class C
2 {
3     void Method(int x)
4     {
5         x = x + 1;
6     }
7 }
```

The syntax tree on the left is structured as follows:

- CompilationUnit [0..71]
 - ClassDeclaration [0..71]
 - ClassKeyword [0..5]
 - IdentifierToken [6..7]
 - OpenBraceToken [9..10]
 - MethodDeclaration [16..68]
 - PredefinedType [16..20]
 - VoidKeyword [16..20]
 - IdentifierToken [21..27]
 - ParameterList [27..34]
 - OpenParenToken [27..28]
 - Parameter [28..33]
 - PredefinedType [28..31]
 - IdentifierToken [32..33]
 - CloseParenToken [33..34]
 - Block [40..68]
 - OpenBraceToken [40..41]
 - ExpressionStatement [51..61]
 - SimpleAssignmentExpression [51..60]
 - IdentifierName [51..52]
 - IdentifierToken [51..52]
 - EqualsToken [53..54]
 - AddExpression [55..60]
 - IdentifierName [55..56]
 - IdentifierToken [55..56]
 - PlusToken [57..58]
 - NumericLiteralExpression [59..60]
 - NumericLiteralToken [59..60]
 - SemicolonToken [60..61]
 - CloseBraceToken [67..68]
 - CloseBraceToken [70..71]
 - EndOfFileToken [71..71]

Red arrows indicate the mapping between the source code and the syntax tree nodes:

- Line 1: `class` maps to `ClassKeyword [0..5]`
- Line 1: `C` maps to `IdentifierToken [6..7]`
- Line 3: `void` maps to `VoidKeyword [16..20]`
- Line 3: `Method` maps to `IdentifierToken [21..27]`
- Line 3: `(` maps to `OpenParenToken [27..28]`
- Line 3: `int` maps to `IdentifierToken [32..33]`
- Line 3: `)` maps to `CloseParenToken [33..34]`
- Line 4: `{` maps to `OpenBraceToken [40..41]`
- Line 5: `x` maps to `IdentifierToken [51..52]`
- Line 5: `=` maps to `EqualsToken [53..54]`
- Line 5: `x` maps to `IdentifierToken [55..56]`
- Line 5: `+` maps to `PlusToken [57..58]`
- Line 5: `1` maps to `NumericLiteralToken [59..60]`
- Line 5: `;` maps to `SemicolonToken [60..61]`
- Line 6: `}` maps to `CloseBraceToken [67..68]`
- Line 7: `}` maps to `CloseBraceToken [70..71]`

Syntax.xml SyntaxKind.cs Microsoft.CodeAnalysis.CSharp.SyntaxKind SwitchExpression

```

325 ObjectCreationExpression = 8649,
326 AnonymousObjectCreationExpression = 8650,
327 ArrayCreationExpression = 8651,
328 ImplicitArrayCreationExpression = 8652,
329 StackAllocArrayCreationExpression = 8653,
330 OmittedArraySizeExpression = 8654,
331 InterpolatedStringExpression = 8655,
332 ImplicitElementAccess = 8656,
333 IsPatternExpression = 8657,
334 SwitchExpression = 8658,
335
336 Add SwitchExpression to public API
337 Suppress RS0016
338 SubtractExpression
339 MultiplyExpression

```

RS0016 Symbol 'SwitchExpression' is not part of the declared API.

Microsoft.CodeAnalysis.CSharp.SyntaxKind.SingleVariableDesignation = 8927 -> Microsoft.CodeAnalysis.CSharp.SyntaxKind
Microsoft.CodeAnalysis.CSharp.SyntaxKind.SwitchExpression = 8658 -> Microsoft.CodeAnalysis.CSharp.SyntaxKind
Microsoft.CodeAnalysis.CSharp.SyntaxKind.ThrowExpression = 9852 -> Microsoft.CodeAnalysis.CSharp.SyntaxKind

Preview changes
Fix all occurrences in: Document Project Solution

24 Errors

Code	Description	Project	File	Line	Suppression S...
RS0016	Symbol 'VisitSwitchExpression' is not part of the declared API.	CSharpCodeAnalysis	Syntax.xml.Main.Generate...	182	Active
RS0016	Symbol 'VisitSwitchExpression' is not part of the declared API.	CSharpCodeAnalysis	Syntax.xml.Main.Generate...	1415	Active
RS0016	Symbol 'VisitSwitchExpression' is not part of the declared API.	CSharpCodeAnalysis	Syntax.xml.Main.Generate...	2683	Active
RS0016	Symbol 'SwitchExpression' is not part of the declared API.	CSharpCodeAnalysis	Syntax.xml.Main.Generate...	5126	Active
RS0016	Symbol 'SwitchExpression' is not part of the declared API.	CSharpCodeAnalysis	Syntax.xml.Main.Generate...	5153	Active
RS0016	Symbol 'SwitchExpression' is not part of the declared API.	CSharpCodeAnalysis	Syntax.xml.Main.Generate...	5159	Active
RS0016	Symbol 'SwitchExpressionSyntax' is not part of the declared API.	CSharpCodeAnalysis	Syntax.xml.Syntax.Generat...	2321	Active

Syntax Visualizer Class1.cs ClassLibrary2 Class Main(string[] args)

Syntax Tree

```

IdentifierToken [39..43]
  ParameterList [43..58]
    Block [64..175]
      OpenBraceToken [64..65]
      ExpressionStatement [75..168]
        InvocationExpression [75..167]
          SimpleMemberAccessExpression [75..99]
            ArgumentList [99..167]
              OpenParenToken [99..100]
              Argument [100..166]
                SimpleMemberAccessExpression [100..111]
                  QuestionColonToken [112..114]
                  BracketedArgumentList [115..124]
                    ColonToken [125..126]
                    BracketedArgumentList [127..166]
                  CloseParenToken [166..167]
              SemicolonToken [167..168]
            CloseBraceToken [174..175]

```

```

class Class
{
    public static void Main(string[] args)
    {
        System.Console.WriteLine(args.Length % 2 : [0, 1, 2] : ["Zero", "One", "Two", "More than two"]);
    }
}

```

Class1.cs ClassLibrary2 Class M(int expr)

```

1 class Class
2 {
3     void M(int expr)
4     {
5         var exprStr = expr % 2 : [1, 2, 3] : ["One", "Two", "Three", "More than three"];
6         System.Console.WriteLine(exprStr);
7     }
8 }

```

0 Errors

Code	Description	Project	File
------	-------------	---------	------

```
Administrator: Developer Command Prompt for VS 2017
c:\roslyn>type test.cs
class Class
{
    public static void Main(string[] args)
    {
        System.Console.WriteLine(args.Length ? : [0, 1, 2] : ["Zero", "One", "Two", "More than two"]);
    }
}
c:\roslyn>c:\roslyn\Binaries\Debug\Exes\csc\csc.exe test.cs
Microsoft (R) Visual C# Compiler version 42.42.42424
Copyright (C) Microsoft Corporation. All rights reserved.

c:\roslyn>test.exe
CodeGen not yet implemented for: 'args.Length ? : [0, 1, 2] : ["Zero", "One", "Two", "More than two"]'
c:\roslyn>
```

```
Administrator: Developer Command Prompt for VS 2017
c:\roslyn>type test.cs
class Class
{
    public static void Main(string[] args)
    {
        System.Console.WriteLine Class ? : [0, 1, 2] : ["Zero", "One", "Two", "More than two"]);
    }
}
c:\roslyn>c:\roslyn\Binaries\Debug\Exes\csc\csc.exe test.cs
Microsoft (R) Visual C# Compiler version 42.42.42424
Copyright (C) Microsoft Corporation. All rights reserved.

test.cs(5,34): error CS0119: 'Class' is a type, which is not valid in the given context
c:\roslyn>
```

```
Administrator: Developer Command Prompt for VS 2017
c:\roslyn>c:\roslyn\Binaries\Debug\Exes\csc\csc.exe test.cs
Microsoft (R) Visual C# Compiler version 42.42.42424
Copyright (C) Microsoft Corporation. All rights reserved.

c:\roslyn>test.exe
Zero

c:\roslyn>test.exe arg1
One

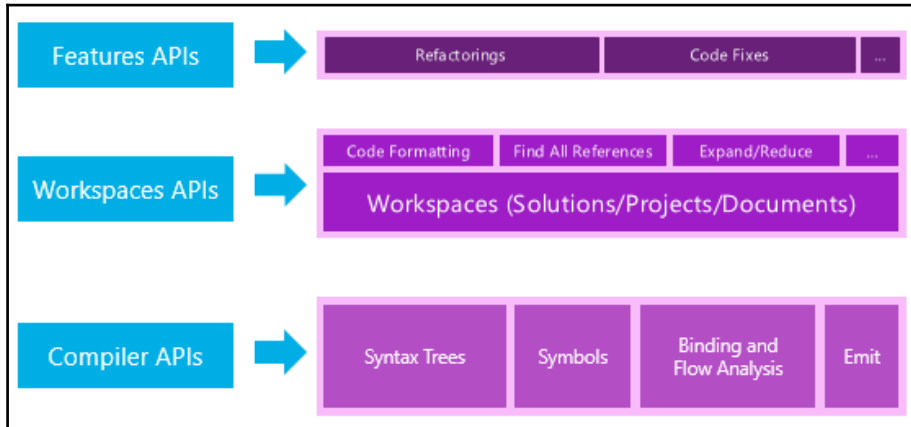
c:\roslyn>test.exe arg1 arg2
Two

c:\roslyn>test.exe arg1 arg2 arg3
More than two

c:\roslyn>
```

```
Administrator: Developer Command Prompt for VS 2017
c:\roslyn>C:\Users\mavasani\.nuget\packages\xunit.runner.console\2.2.0-beta4-build3444\tools\xunit.console.x86.exe "c:\r
oslyn\Binaries\Debug\Dlls\CSharpCompilerSyntaxTest\Roslyn.Compilers.CSharp.Syntax.UnitTests.dll" -html "c:\roslyn\Binar
ies\Debug\Dlls\CSharpCompilerSyntaxTest\XUnitResults\Roslyn.Compilers.CSharp.Syntax.UnitTests.html" -noshadow -method M
icrosoft.CodeAnalysis.CSharp.UnitTests.ExpressionParsingTexts.TestSwitchExpression
xUnit.net Console Runner (32-bit .NET 4.0.30319.42000)
  Discovering: Roslyn.Compilers.CSharp.Syntax.UnitTests
  Discovered:  Roslyn.Compilers.CSharp.Syntax.UnitTests
  Starting:    Roslyn.Compilers.CSharp.Syntax.UnitTests
  Finished:   Roslyn.Compilers.CSharp.Syntax.UnitTests
=== TEST EXECUTION SUMMARY ===
  Roslyn.Compilers.CSharp.Syntax.UnitTests Total: 1, Errors: 0, Failed: 0, Skipped: 0, Time: 0.289s
c:\roslyn>
```

Chapter 10: Command-Line Tools Based on Roslyn API



```
Administrator: Developer Command Prompt for VS 2017
C:\Users\mavasani\Documents\Visual Studio 2017\Projects\ConsoleApp\ConsoleApp>bin\Debug\ConsoleApp.exe
Usage: ConsoleApp.exe <%file_path%>

C:\Users\mavasani\Documents\Visual Studio 2017\Projects\ConsoleApp\ConsoleApp>bin\Debug\ConsoleApp.exe test.cs
Transformed source:

// Class with no accessibility modifier
internal class C1
{
    void M() {}
}

// Public class with no documentation comments
/// <summary>TODO: Add doc comments</summary>
public class C2
{
    void M() {}
}

C:\Users\mavasani\Documents\Visual Studio 2017\Projects\ConsoleApp\ConsoleApp>
```

```
Administrator: Developer Command Prompt for VS 2017
C:\Users\mavasani\Documents\Visual Studio 2017\Projects\ConsoleApp\ConsoleApp\bin\Debug\ConsoleApp.exe test.cs
Number of diagnostics: 1
(7,8): error CS1503: Argument 1: cannot convert from '<null>' to 'int'

Invocation: 'M1()'
  Overload resolution result: Succeeded
  Method Symbol: void C1.M1()

Invocation: 'M2(0)'
  Overload resolution result: Succeeded
  Method Symbol: void C1.M2(int x)

Invocation: 'M2(null)'
  Overload resolution result: OverloadResolutionFailure
  2 candidate symbols:
  Candidate Symbol: void C1.M2()
  Candidate Symbol: void C1.M2(int x)

C:\Users\mavasani\Documents\Visual Studio 2017\Projects\ConsoleApp\ConsoleApp>
```

```
Program.cs
TestSolution
TestSolution.Program
1 namespace TestSolution
2 {
3     class Program
4     {
5         static void Main(string[] args)
6         {
7             var implicitlyTypedLocal = 0;
8             int explicitlyTypedLocal = 1;
9         }
10    }
11 }
```

```
Program.cs
TestSolution
TestSolution.Program
1 namespace TestSolution
2 {
3     class Program
4     {
5         static void Main(string[] args)
6         {
7             var implicitlyTypedLocal = 0;
8             var explicitlyTypedLocal = 1;
9         }
10    }
11 }
```

```
Developer Command Prompt for VS 2017 - bin\Debug\ConsoleApp.exe "c:\users\mavasani\documents\visual studio 2017\Projects\TestSolutio...
C:\Users\mavasani\Documents\Visual Studio 2017\Projects\ConsoleApp\ConsoleApp>bin\Debug\ConsoleApp.exe "c:\users\mavasani\documents\visual studio 2017\Projects\TestSolution\TestSolution.sln"
Loading solution 'c:\users\mavasani\documents\visual studio 2017\Projects\TestSolution\TestSolution.sln'...
Project count: 2
Project: TestSolution
  Assembly name: TestSolution
  Language: C#
  Project file: c:\users\mavasani\documents\visual studio 2017\Projects\TestSolution\TestSolution\TestSolution.csproj
  Output file: c:\users\mavasani\documents\visual studio 2017\Projects\TestSolution\TestSolution\bin\Debug\TestSolution.dll
  Documents: 3
  Metadata references: 9
  Project references: 0

Project: ClassLibrary
  Assembly name: ClassLibrary
  Language: C#
  Project file: c:\users\mavasani\documents\visual studio 2017\Projects\TestSolution\ClassLibrary\ClassLibrary.csproj
  Output file: c:\users\mavasani\documents\visual studio 2017\Projects\TestSolution\ClassLibrary\bin\Debug\ClassLibrary.dll
  Documents: 3
  Metadata references: 9
  Project references: 0

Press any key to continue...

```

```
Press any key to continue...

Adding project 'AddedClassLibrary'...
Removing project 'ClassLibrary'...
Adding project reference from 'AddedClassLibrary' to 'TestSolution'...
Press any key to continue...

```

```
Project count: 2
Project: TestSolution
  Assembly name: TestSolution
  Language: C#
  Project file: c:\users\mavasani\documents\visual studio 2017\Projects\TestSolution\TestSolution\TestSolution.csproj
  Output file: c:\users\mavasani\documents\visual studio 2017\Projects\TestSolution\TestSolution\bin\Debug\TestSolution.dll
  Documents: 3
  Metadata references: 9
  Project references: 1

Project: AddedClassLibrary
  Assembly name: AddedProjectAssembly
  Language: C#
  Project file:
  Output file:
  Documents: 0
  Metadata references: 0
  Project references: 0

C:\Users\mavasani\Documents\Visual Studio 2017\Projects\ConsoleApp\ConsoleApp>

```