

An Introduction to the **.NET Compiler Platform**

“What Can It Do For Me?”

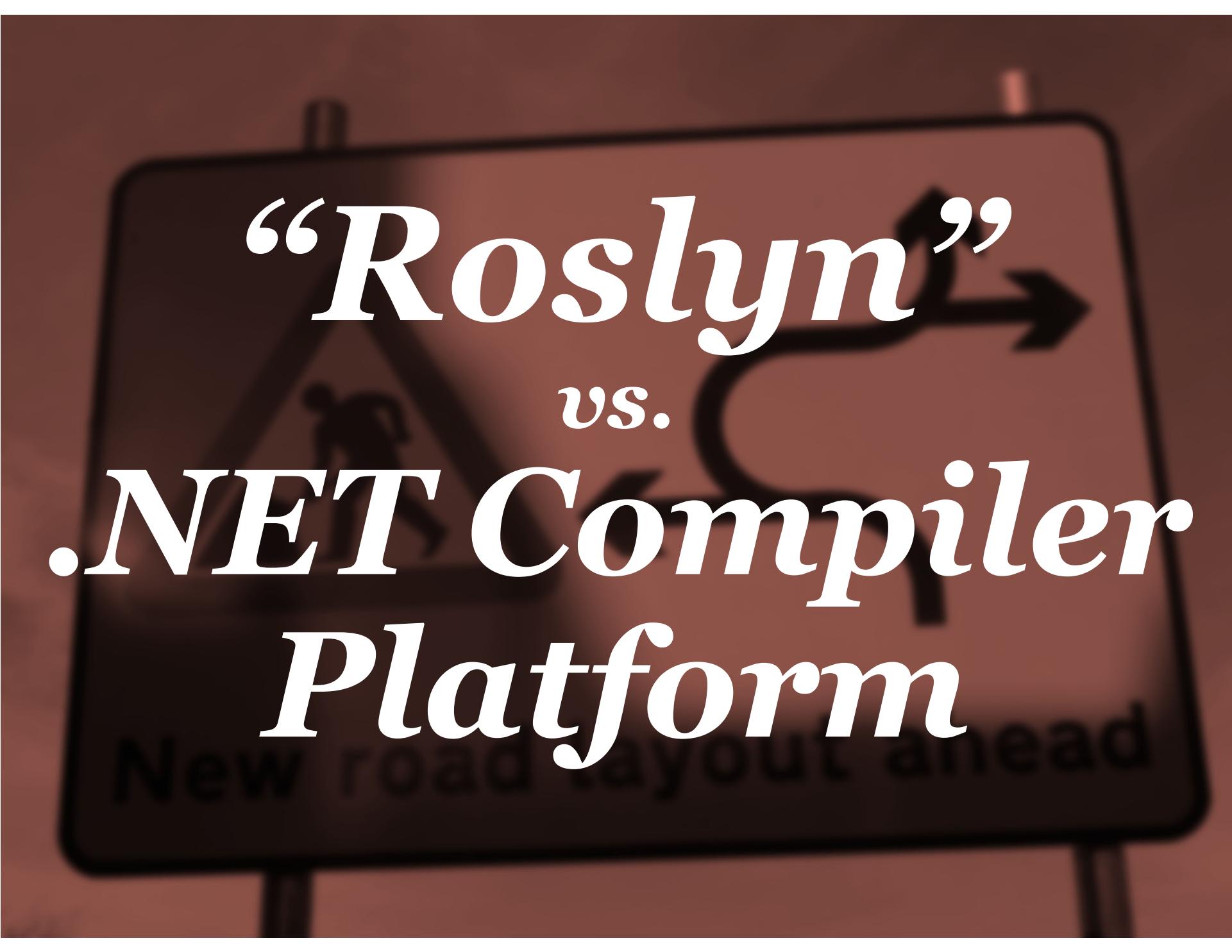
Presented By

Dave Glick

@daveaglick

<http://daveaglick.com>

Note: All images are Creative Commons licensed and attributed at the bottom of their first use as necessary.

A dark brown rectangular sign is mounted on a post. It features a white arrow pointing to the right at the top, followed by the text "New Road ahead" in a white sans-serif font.

“Roslyn” vs. .NET Compiler Platform

A Little History...



Image from <https://www.flickr.com/photos/lungstruck/3224745399>

A Little History...

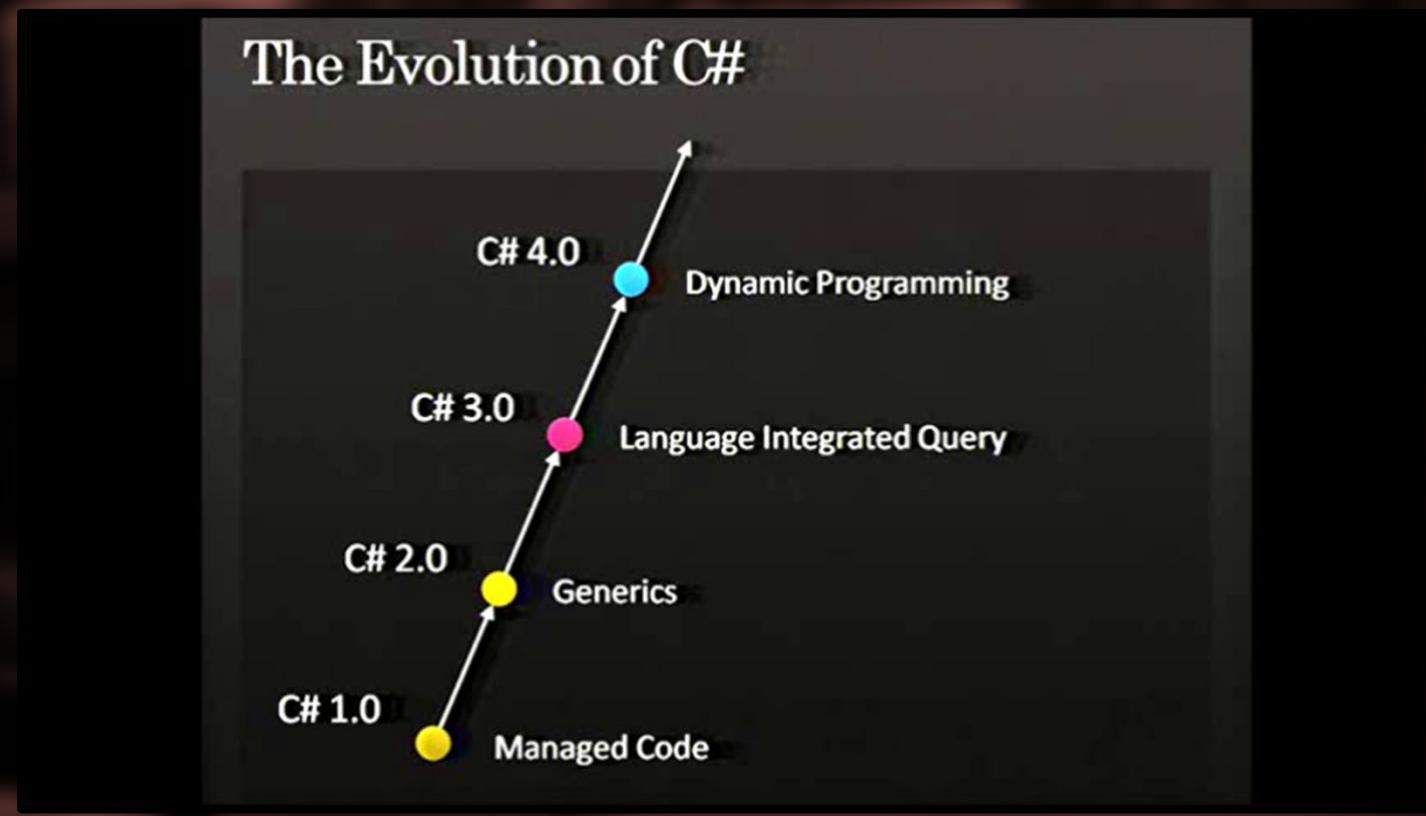
October 2008 (?)

“Compiler as a service” is announced

A Little History...

October 2008 (?)

“Compiler as a service” is announced



A Little History...

October 2008 (!)

“Compiler as a service” is announced

October 2011

First CTP

A Little History...

October 2008 (!)

“Compiler as a service” is announced

October 2011

First CTP

April 2014 (Build Conference)

Roslyn goes open source

A Little History...

October 2008 (!)

“Compiler as a service” is announced

October 2011

First CTP

April 2014 (Build Conference)

Roslyn goes open source

January 2015

Project moves to GitHub

A Little History...

October 2008 (!)

“Compiler as a service” is announced

October 2011

First CTP

April 2014 (Build Conference)

Roslyn goes open source

January 2015

Project moves to GitHub

Spring 2015

Renamed to .NET Compiler Platform

A Little History...

July 2015

.NET Compiler Platform 1.0 Released!

The background of the image features a sparkler against an orange gradient. The sparkler is positioned in the center, with its bright sparks radiating outwards towards the edges. The orange gradient is darker at the top and lighter at the bottom, creating a warm, celebratory feel.

*Why the
sudden
interest?*

Image from <https://www.flickr.com/photos/schtumple/10914109233>

DAT SUN

180B SSS 2-DOOR HARDTOP

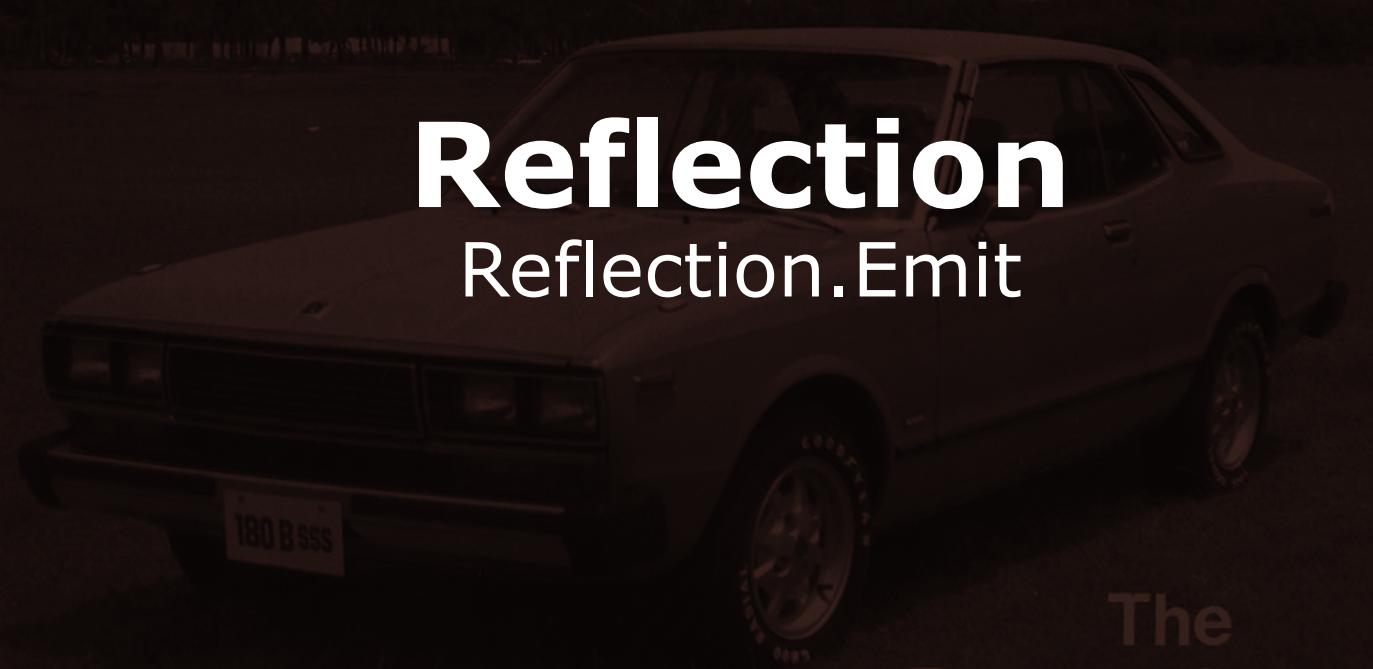
*What Used To
Be Used?*



The
Responsive car
for the man
of action.

DAT SUN

180B SSS 2-DOOR HARDTOP



Reflection
Reflection.Emit

The
Responsive car
for the man
of action.

DAT SUN

180B SSS 2-DOOR HARDTOP

CodeDOM



The
Responsive car
for the man
of action.

DAT SUN

180B SSS 2-DOOR HARDTOP

Expression Trees



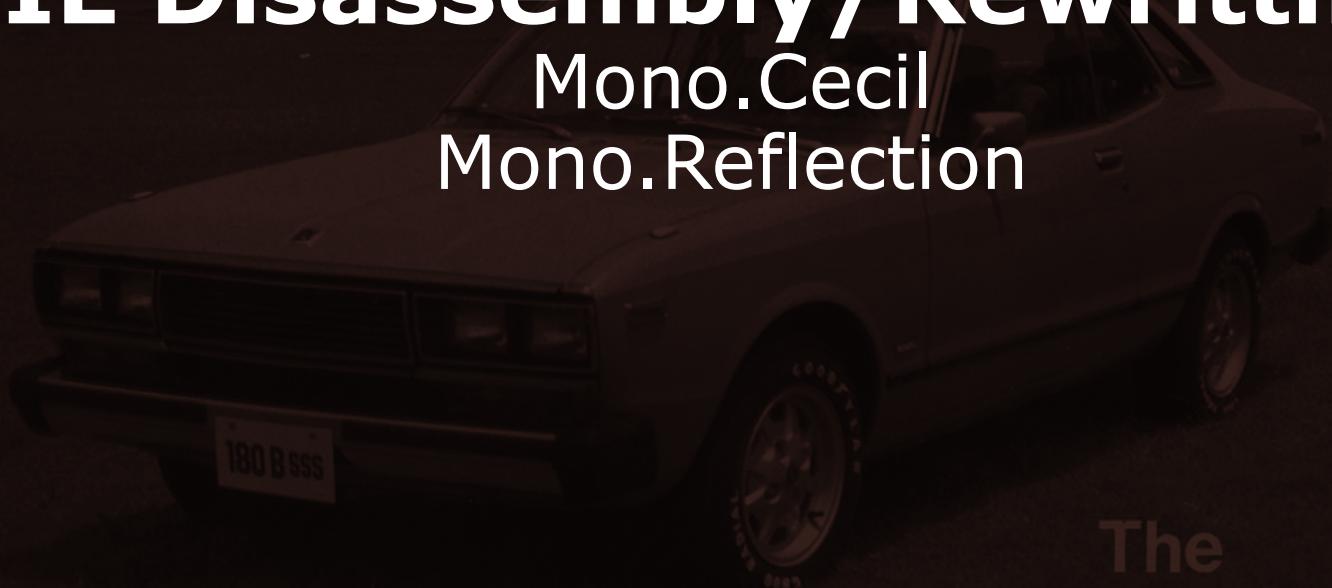
The
Responsive car
for the man
of action.

DAT SUN

180B SSS 2-DOOR HARDTOP

IL Disassembly/Rewriting

Mono.Cecil
Mono.Reflection



The
Responsive car
for the man
of action.

DAT SUN

180B SSS 2-DOOR HARDTOP

T4



The
Responsive car
for the man
of action.

What Is Roslyn?

It's a toolbox for...

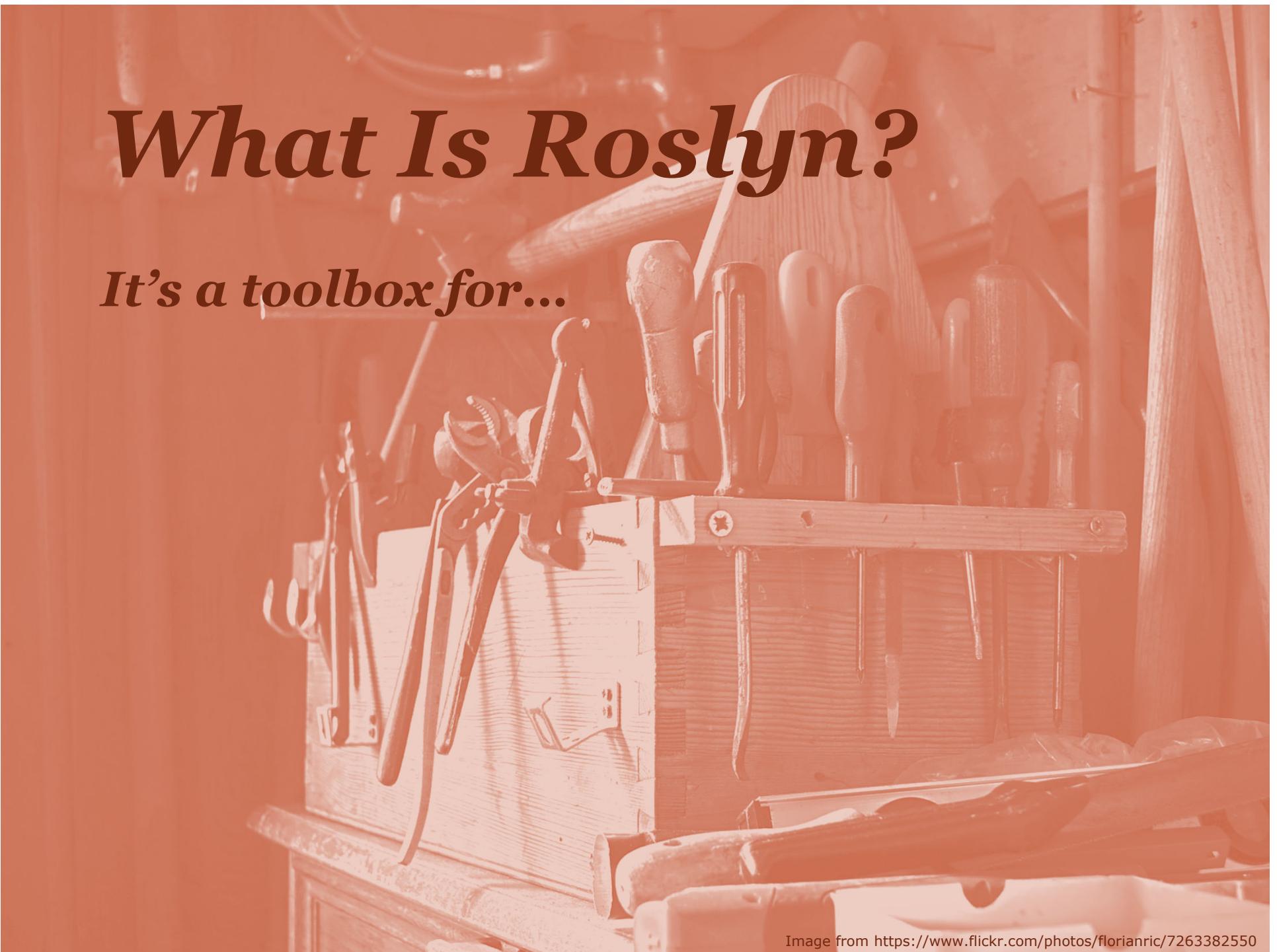


Image from <https://www.flickr.com/photos/florianric/7263382550>

What Is Roslyn?

It's a toolbox for...

Syntactic Analysis

What Is Roslyn?

It's a toolbox for...

Syntactic Analysis
Semantic Analysis

What Is Roslyn?

It's a toolbox for...

Syntactic Analysis

Semantic Analysis

Creating/Manipulating Code

What Is Roslyn?

It's a toolbox for...

Syntactic Analysis

Semantic Analysis

Creating/Manipulating Code

Compiling Code

What Is Roslyn?

It's a toolbox for...

Syntactic Analysis

Semantic Analysis

Creating/Manipulating Code

Compiling Code

**Working with
Projects/Solutions**

What Is Roslyn?

It's a toolbox for...

Syntactic Analysis

Semantic Analysis

Creating/Manipulating Code

Compiling Code

**Working with
Projects/Solutions**

Scripting

What Is Roslyn?

Features
APIs

Refactorings

Code Fixes

...

Workspace
APIs

Code Formatting

Find All References

Expand/Reduce

...

Workspaces
(Solutions/Projects/Documents)

Compiler
APIs

Syntax Trees

Symbols

Binding and
Flow Analysis

Emit

What Is Roslyn Not?

Image from <https://www.flickr.com/photos/dominicotine/7696681446>



What Is Roslyn Not?

Not Resharper, JustCode, etc.

What Is Roslyn Not?

Not Resharper, JustCode, etc.
Not for building DSLs

What Is Roslyn Not?

Not Resharper, JustCode, etc.
Not for building DSLs
Not an AOP framework (yet)

What Is Roslyn Not?

Not Resharper, JustCode, etc.

Not for building DSLs

Not an AOP framework (yet)

Not a disassembler

Getting Up and Running



Image from <https://www.flickr.com/photos/jwelcher/4638164698>

A photograph of a single, leafless tree standing in shallow, reflective water. The tree's branches are intricate and reach outwards. The background shows a vast, calm sea under a sky filled with soft, scattered clouds. The sun is low on the horizon, casting a warm, golden glow over the scene. The water reflects the light and the tree's silhouette.

Syntax Trees

Image from <https://www.flickr.com/photos/27870539@N07/7683557220>

Nodes, Tokens, and Trivia, OH MY!



Image from <https://www.flickr.com/photos/jamesjordan/2226419650>

How To Create A Syntax Tree?



How To Create A Syntax Tree?

CSharpSyntaxTree.ParseText(...)

How To Create A Syntax Tree?

CSharpSyntaxTree.Create(...)

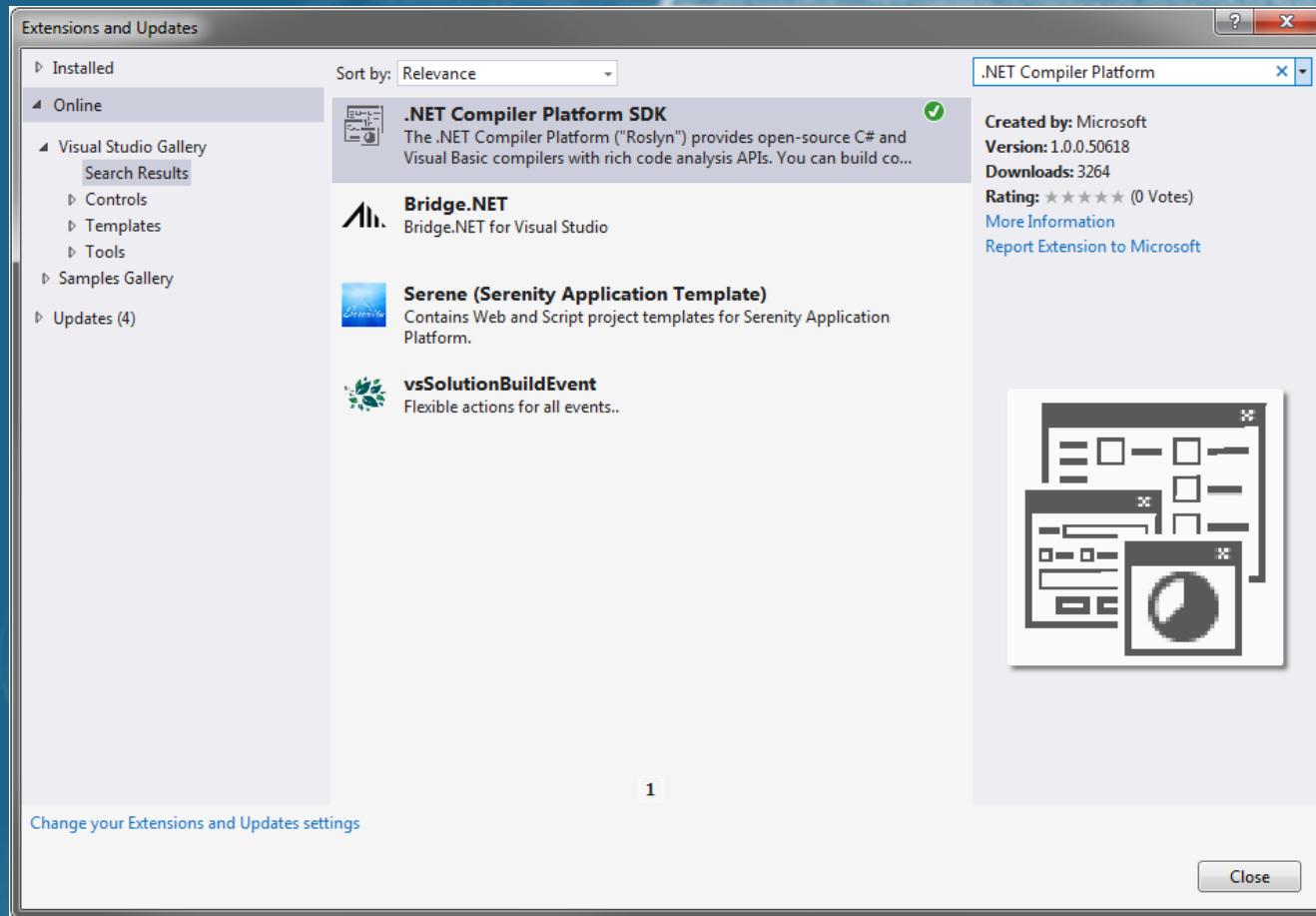
How To Create A Syntax Tree?

SyntaxFactory

Visual Studio Syntax Visualizer

Image from <https://www.flickr.com/photos/practicalowl/1185217331>

Visual Studio Syntax Visualizer



RoslynSamples - Microsoft Visual Studio (Administrator)

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

Debug Any CPU Start Submit

Program.cs* RoslynSamples.RoslynSamples.Program Main(string[] args)

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace RoslynSamples
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13         }
14     }
15 }
16
```

Server Explorer Toolbox

Solution Explorer Team Explorer Class View

100 %

Ready Ln 12 Col 10 Ch 10 INS

The screenshot shows the Microsoft Visual Studio interface with the title "RoslynSamples - Microsoft Visual Studio (Administrator)". The menu bar includes File, Edit, View, Project, Build, Debug, Team, Tools, Architecture, Test, Analyze, Window, and Help. The toolbar has icons for Back, Forward, Home, Stop, Refresh, and various project-related buttons. The status bar at the bottom shows "Ready" and "Ln 12 Col 10 Ch 10 INS". The main code editor window displays "Program.cs*" with the following C# code:

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace RoslynSamples
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13         }
14     }
15 }
16
```

The Solution Explorer, Team Explorer, and Class View panes are visible on the right side of the interface.

RoslynSamples - Microsoft Visual Studio (Administrator)

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

Debug Any CPU Start Roslyn Syntax Visualizer

Program.cs* RoslynSamples.RoslynSamples.Program

```
3     using System.Linq;
4     using System.Text;
5     using System.Threading.Tasks;
6
7     namespace RoslynSamples
8     {
9         class Program
10        {
11            static void Main(string[] args)
12            {
13            }
14
15            int X { get; set; }
16
17            int Y { get; set; }
18        }
19    }
20
```

Syntax Tree

- ClassDeclaration [155..307]
 - ClassKeyword [155..160]
 - IdentifierToken [161..168]
 - OpenBraceToken [174..175]
 - MethodDeclaration [185..238]
 - PropertyDeclaration [250..269]
 - PredefinedType [250..253]
 - IdentifierToken [254..255]
 - AccessorList [256..269]
 - OpenBraceToken [256..257]
 - GetAccessorDeclaration [258..262]
 - SetAccessorDeclaration [263..267]
 - CloseBraceToken [268..269]
 - Trail: EndOfLineTrivia [269..271]

Properties

Type	SyntaxTrivia
Kind	EndOfLineTrivia
ContainsDiagnostics	False
FullSpan	[269..271)
HasStructure	False
IsDirective	False
Language	C#
RawKind	8539
Span	[269..271)
SpanStart	269
SyntaxTree	using System;using System.Collections.Generic;
Token	}

100 %

Ready Ln 16 Col 9 Ch 9 INS

Exploring the Tree



Image from <https://www.flickr.com/photos/sjaustin/3008842794>

Exploring the Tree

LINQ-to-Objects

Exploring the Tree

CSharpSyntaxWalker

Manipulating the Tree

Image from <https://www.flickr.com/photos/fringedbenefit/5033077381>



Manipulating the Tree

CSharpSyntaxRewriter



Compiling

Image from <https://www.flickr.com/photos/jovanlaar/2137872208>

Creating a Compilation

Image from <https://www.flickr.com/photos/illuminaut/9405952796>

Creating a Compilation

CSharpCompilation.Create(...)

Creating a Compilation

```
CSharpCompilation.Create(...)  
    .AddSyntaxTrees(...)
```

Creating a Compilation

```
CSharpCompilation.Create(...)  
    .AddSyntaxTrees(...)  
    .AddReferences(...)
```

Creating a Compilation

```
CSharpCompilation.Create(...)
```

```
    .AddSyntaxTrees(...)
```

```
    .AddReferences(...)
```

```
    .WithOptions(...)
```

Now What?



Now What?

.Emit(...)

Now What?

`.GetDiagnostics(...)`

Now What?

`.ClassifyConversion(...)`

Now What?

`.GetSymbolsWithName(...)`

Now What?

`.GetSemanticModel(...)`

Semantics

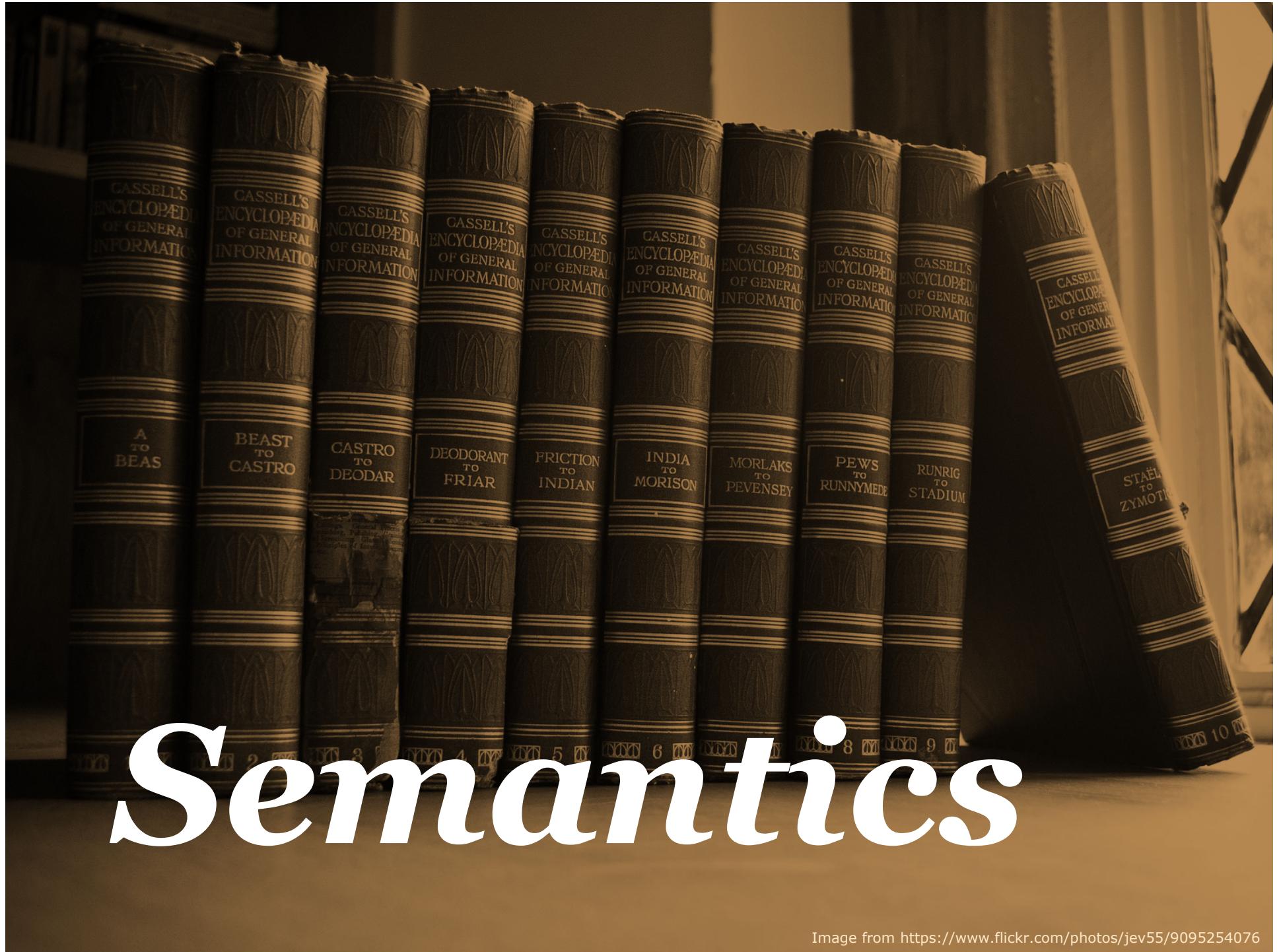


Image from <https://www.flickr.com/photos/jev55/9095254076>

Symbols

Image from <https://www.flickr.com/photos/irglover/6241307029>

Symbols

```
SemanticModel  
.GetDeclaredSymbol(...)
```

Symbols

```
SemanticModel  
.GetSymbolInfo(...)
```

RoslynSamples - Microsoft Visual Studio (Administrator)

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

Debug Any CPU Start

Program.cs* RoslynSamples.RoslynSamples.Program

```
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace RoslynSamples
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13         }
14
15         int X { get; set; }
16         struct System.Int32
17             Represents a 32-bit signed integer.
18         {
19             X = y;
20         }
21     }
22 }
23
```

Server Explorer Toolbox Solution Explorer Team Explorer Class View

Roslyn Syntax Visualizer

Syntax Tree

- ▲ ClassDeclaration [155..349]
 - ▷ ClassKeyword [155..160]
 - ▷ IdentifierToken [161..168]
 - ▷ OpenBraceToken [174..175]
 - ▷ MethodDeclaration [185..238]
- ▲ PropertyDeclaration [250..269]
 - ▲ PredefinedType [250..253]
 - ▲ IntKeyword [250..253]
 - Lead: EndOfLineTrivia [240..242]
 - Lead: WhitespaceTrivia [242..250]
 - Trail: WhitespaceTrivia [253..254]
 - ▷ IdentifierToken [254..255]
 - ▷ AccessorList [256..269]
 - ▷ MethodDeclaration [281..342]

Properties

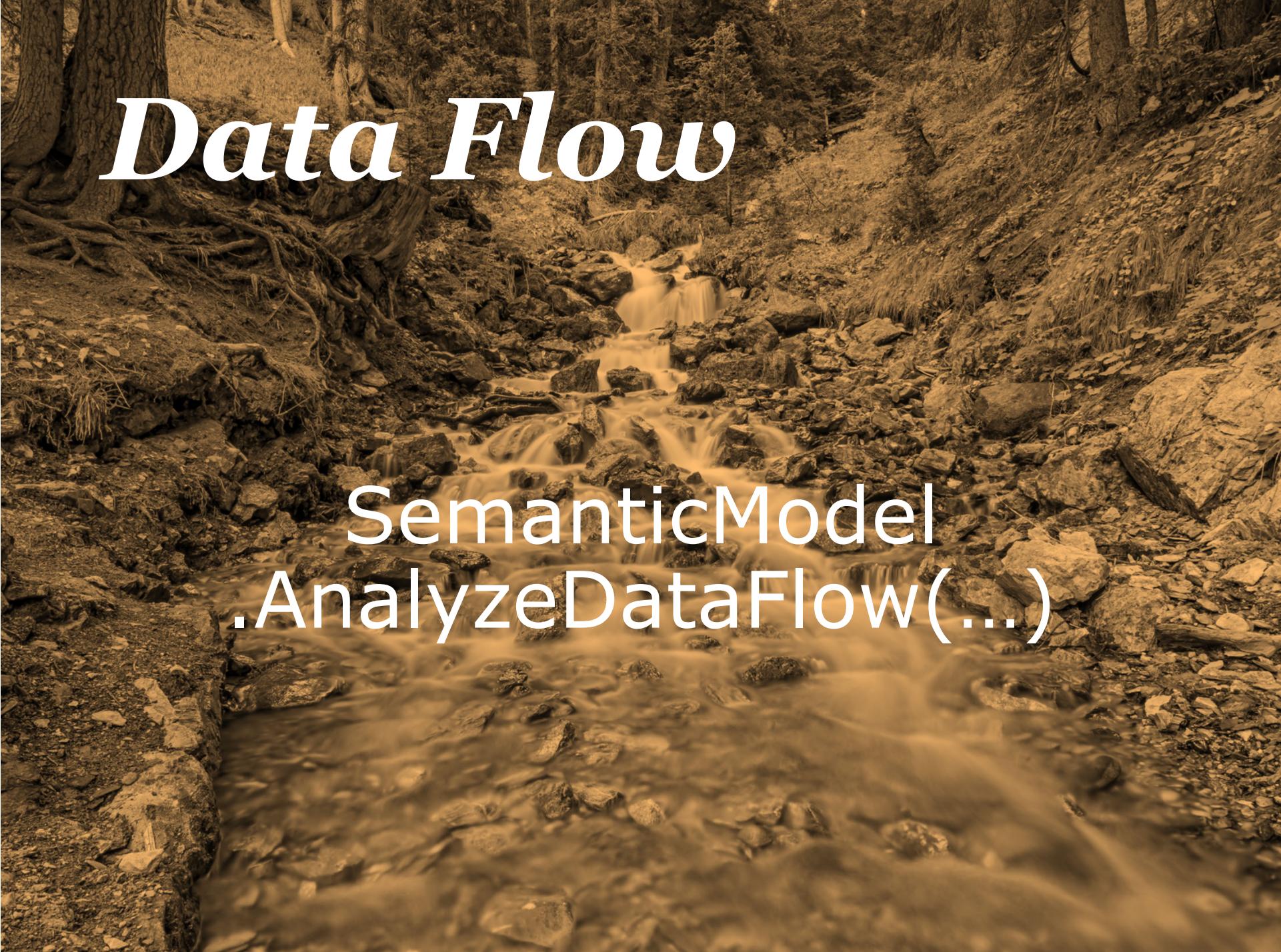
Type	SyntaxTrivia
Kind	WhitespaceTrivia
ContainsDiagnostics	False
FullSpan	[253..254)
HasStructure	False
IsDirective	False
Language	C#
RawKind	8540
Span	[253..254)
SpanStart	253
SyntaxTree	using System;using System.Collections.Generic;
Token	int

100 %

Ready Ln 15 Col 13 Ch 13 INS

Control Flow

SemanticModel
.AnalyzeControlFlow(...)

A sepia-toned photograph of a rocky stream flowing through a forest. The water is moving over and around large rocks, creating small cascades. The banks of the stream are made of dirt and more rocks, with some sparse vegetation. In the background, there are tall evergreen trees. The overall scene conveys a sense of natural flow and movement.

Data Flow

SemanticModel
.AnalyzeDataFlow(...)



Analyzers & Code Fixes

Image from <https://www.flickr.com/photos/tobanblack/3343068574>

Diagnostics

Image from <https://www.flickr.com/photos/kevincollins123/8001780511>

Diagnostics

EmitResult.Diagnostics

Diagnostics

Compilation
.GetDiagnostics(...)

Analyzers

Image from <https://www.flickr.com/photos/kaptainkobold/5473183284>

Analyzers

Derived from

`DiagnosticAnalyzer`

Analyzers

Must override

SupportedDiagnostics { get; }

Initialize(AnalysisContext)

Analyzers

RegisterSyntaxNodeAction	Triggered when a particular kind of syntax node has been parsed
RegisterSymbolAction	Triggered when a particular kind of symbol has been analyzed
RegisterSyntaxTreeAction	Triggered when the file's whole syntax tree has been parsed
RegisterSemanticModelAction	Triggered when a semantic model is available for the whole file
RegisterCodeBlockStartAction RegisterCodeBlockEndAction	Triggered before/after analysis of a method body or other code block
RegisterCompilationStartAction RegisterCompilationEndAction	Triggered before/after analysis of the entire project

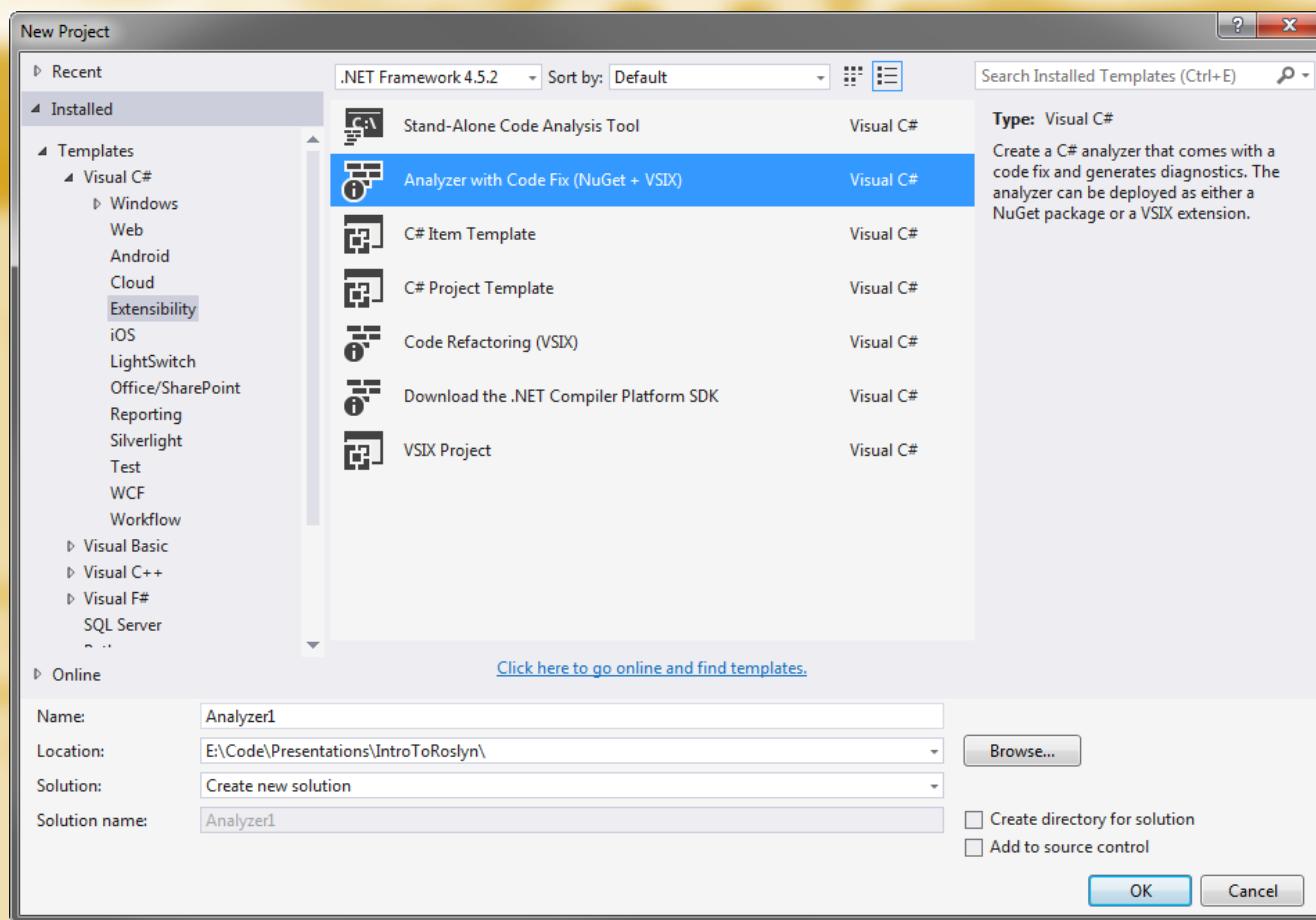
Analyzers

Compilation
.WithAnalyzers(...)

Analyzers

CompilationWithAnalyzers
.GetAnalyzerDiagnosticsAsync()

Analyzers



DontStartClassNamesWithMyAnalyzer - Microsoft Visual Studio (Administrator)

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

Diagnostic Tools DiagnosticAnalyzer.cs CodeFixProvider.cs packages.config

Server Explorer Solution Explorer

DiagnosticAnalyzer.cs

```
10  using Microsoft.CodeAnalysis.Diagnostics;
11  namespace DontStartClassNamesWithMyAnalyzer
12  {
13      [DiagnosticAnalyzer(LanguageNames.CSharp)]
14      public class DontStartClassNamesWithMyAnalyzerAnalyzer : DiagnosticAnalyzer
15      {
16          public const string DiagnosticId = "DG0001";
17          const string Title = "Don't start class names with 'My'";
18          const string MessageFormat = "Class name started with 'My': {0}";
19          const string Category = "Syntax";
20
21          static DiagnosticDescriptor Descriptor =
22              new DiagnosticDescriptor(DiagnosticId, Title, MessageFormat, Category,
23                                      DiagnosticSeverity.Warning, true);
24
25          public override ImmutableArray<DiagnosticDescriptor> SupportedDiagnostics
26          {
27              get { return ImmutableArray.Create(Descriptor); }
28          }
29      }
```

Output

Show output from: Debug

The program '[1932] devenv.exe: Program Trace' has exited with code 0 (0x0).
The program '[1932] devenv.exe' has exited with code 0 (0x0).

Solution Explorer

DontStartClassNamesWithMyAnalyzer

- .nuget
- packages.config
- DontStartClassNamesWithMyAnalyzer (Portable)
 - Properties
 - References
 - tools
 - CodeFixProvider.cs
 - Diagnostic.nuspec
 - DiagnosticAnalyzer.cs
 - packages.config
 - ReadMe.txt
- DontStartClassNamesWithMyAnalyzer.Test
 - Properties
 - References
 - Helpers
 - Verifiers
 - packages.config
 - UnitTests.cs
- DontStartClassNamesWithMyAnalyzer.Vsix
 - References
 - source.extension.vsixmanifest

Ready

Ln 21 Col 35 Ch 35 INS

11:30 PM
4/15/2015

Code Fixes

Image from <https://www.flickr.com/photos/emagic/29615044>

Code Fixes

Derived from

CodeFixProvider

Code Fixes

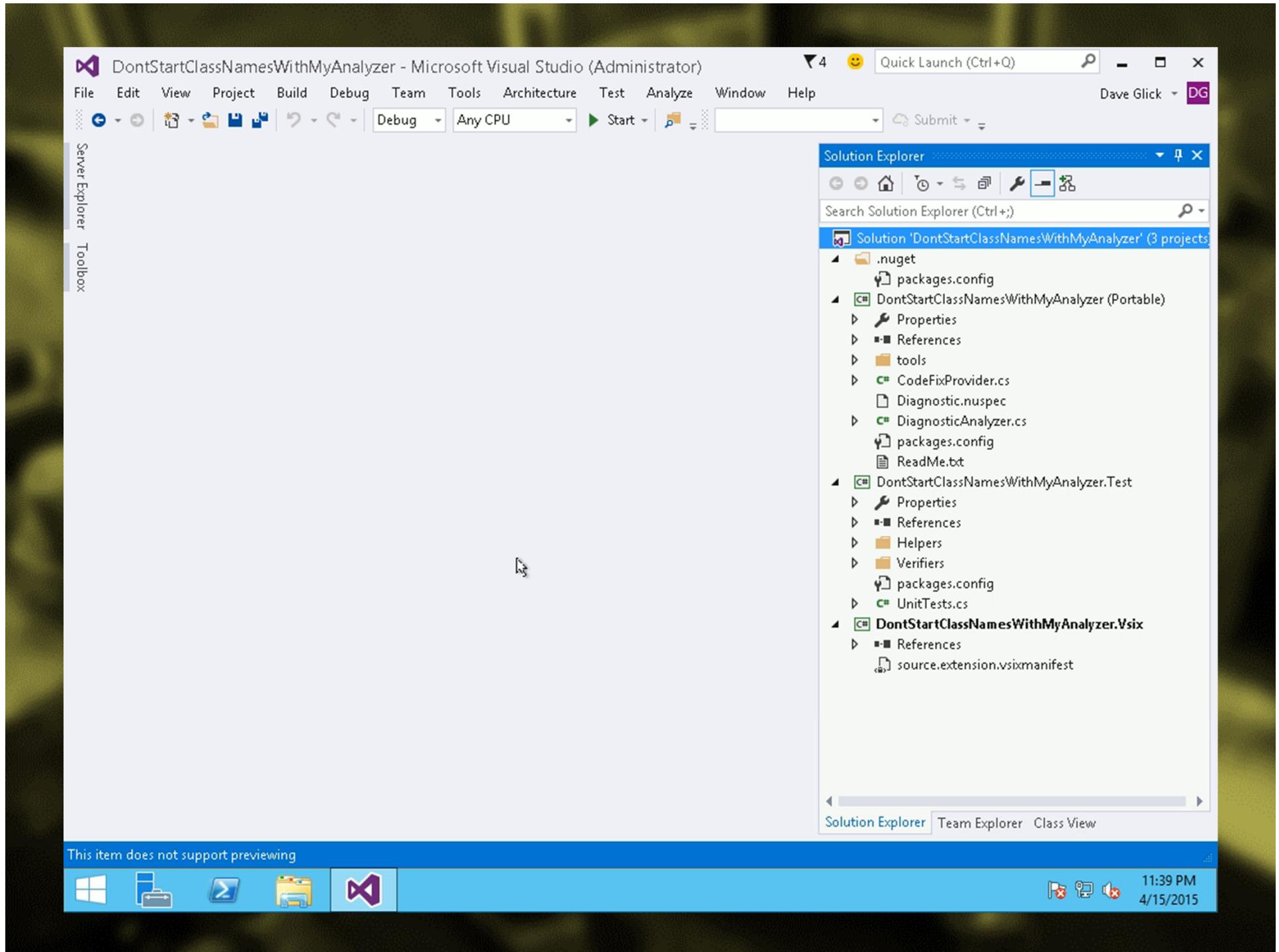
Must override

```
FixableDiagnosticIds { get; }
```

```
RegisterCodeFixesAsync(CodeFixContext)
```

Optionally

```
GetFixAllProvider()
```



A black and white photograph of a quill pen resting on a piece of paper next to an ink bottle.

Scripting

Image from <https://www.flickr.com/photos/buechertiger/8109415421>

*What Can It
Do For Me?*

Additional Resources

- *Introduction to the .NET Compiler Platform* on Pluralsight
- *Learn Roslyn Now* by Josh Varty:
 - <https://joshvarty.wordpress.com/learn-roslyn-now/>
- MSDN Articles by Alex Turner:
 - <https://msdn.microsoft.com/en-us/magazine/dn879356.aspx>
 - <https://msdn.microsoft.com/en-us/magazine/dn904670.aspx>