



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.

“Roslyn” vs. .NET Compiler Platform





“Roslyn”
vs.
*.NET Compiler
Platform*

New road layout ahead

A Little History...



A Little History...

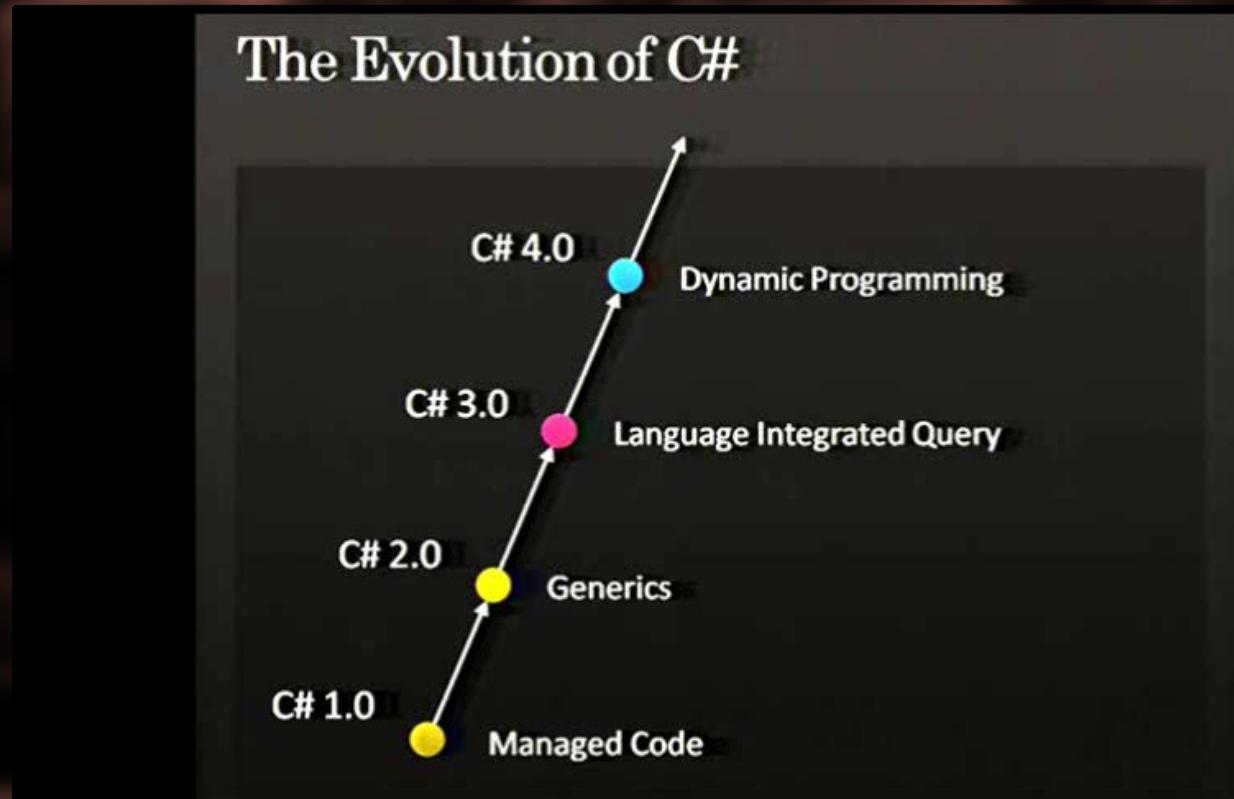
October 2008 (?)

“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 slide features a sparkler against a red gradient. The sparkler is positioned in the center-left, with its bright sparks radiating outwards towards the top and right. The background has a warm, reddish-orange hue that transitions into a darker shade towards the edges.

*Why the
sudden
interest?*



DAT SUN

180B SSS 2-DOOR HARDTOP



*What Gets
Used Today?*

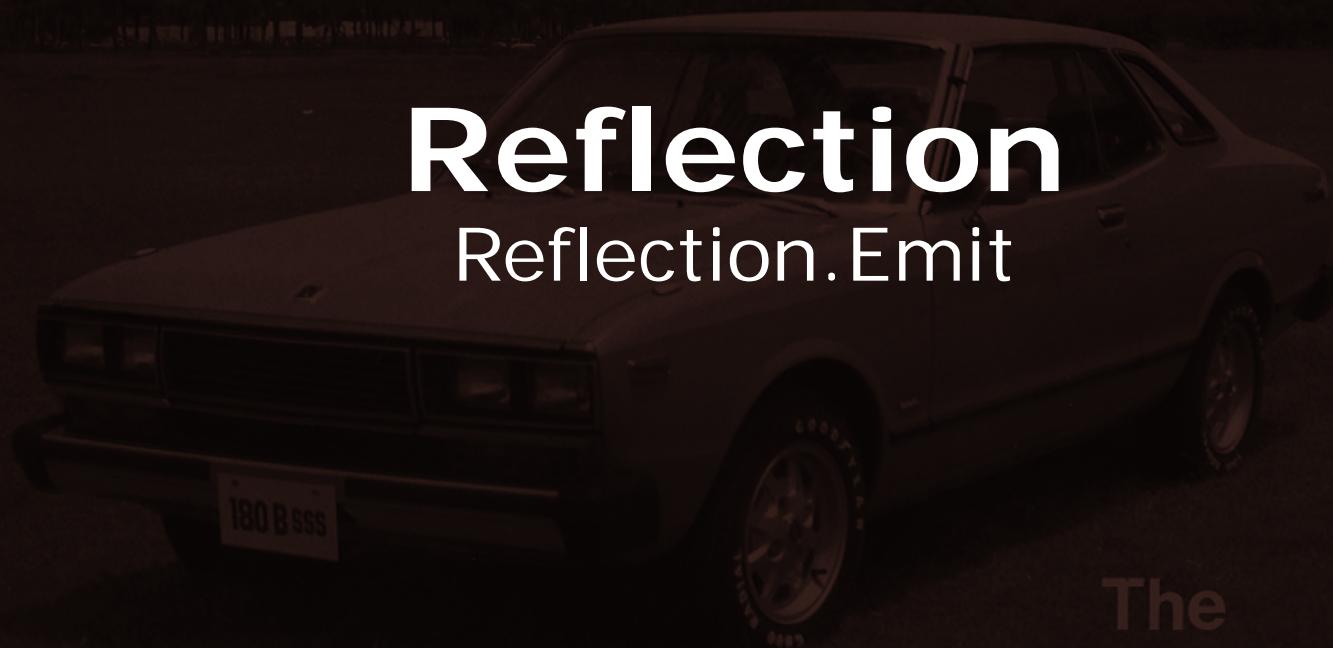
The
Responsive car
for the man
of action.

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...

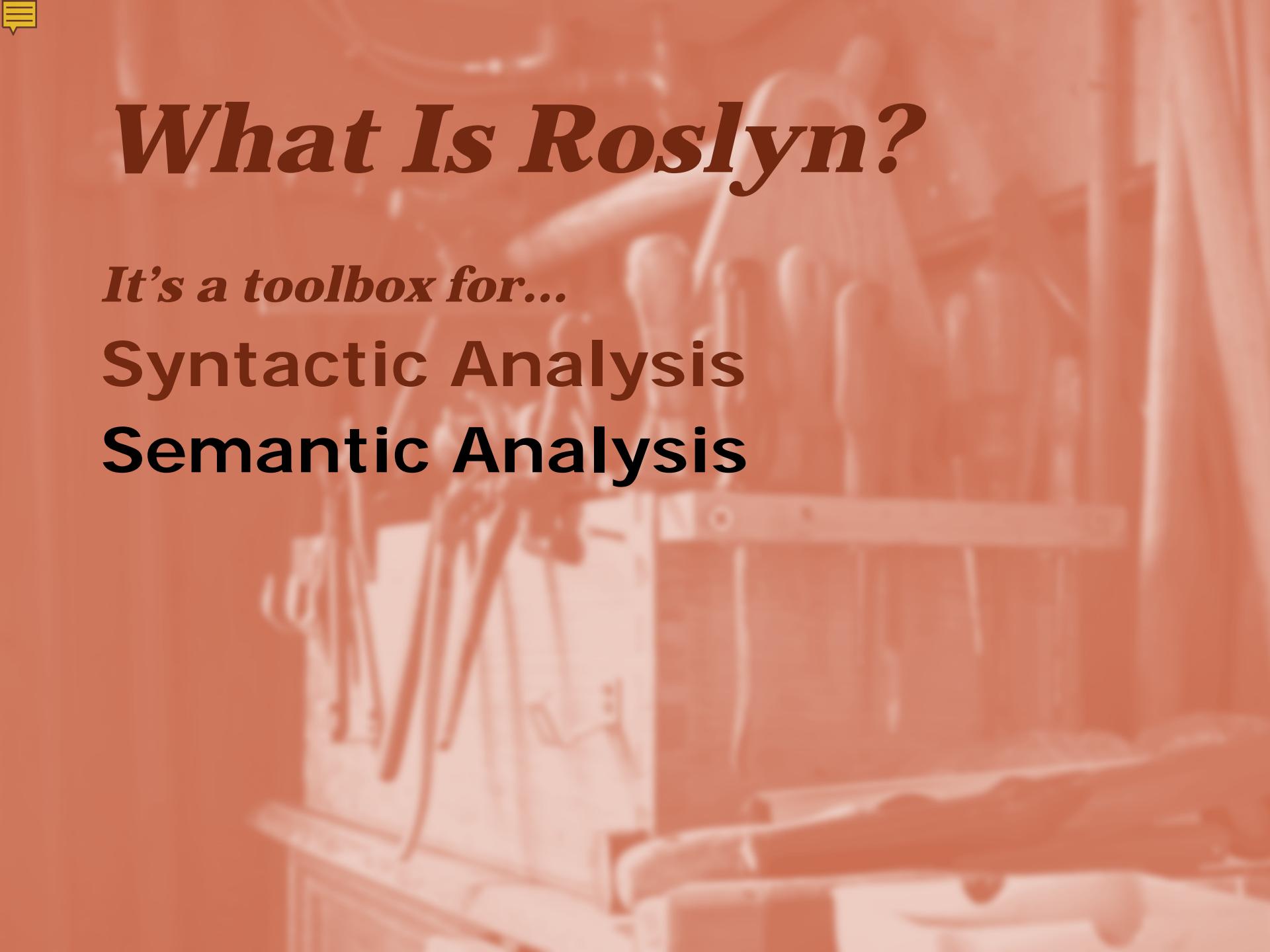


A close-up, slightly blurred photograph of a person's hands resting on a light-colored wooden keyboard. The hands are positioned as if ready to type. The background is dark and out of focus.

What Is Roslyn?

It's a toolbox for...

Syntactic Analysis

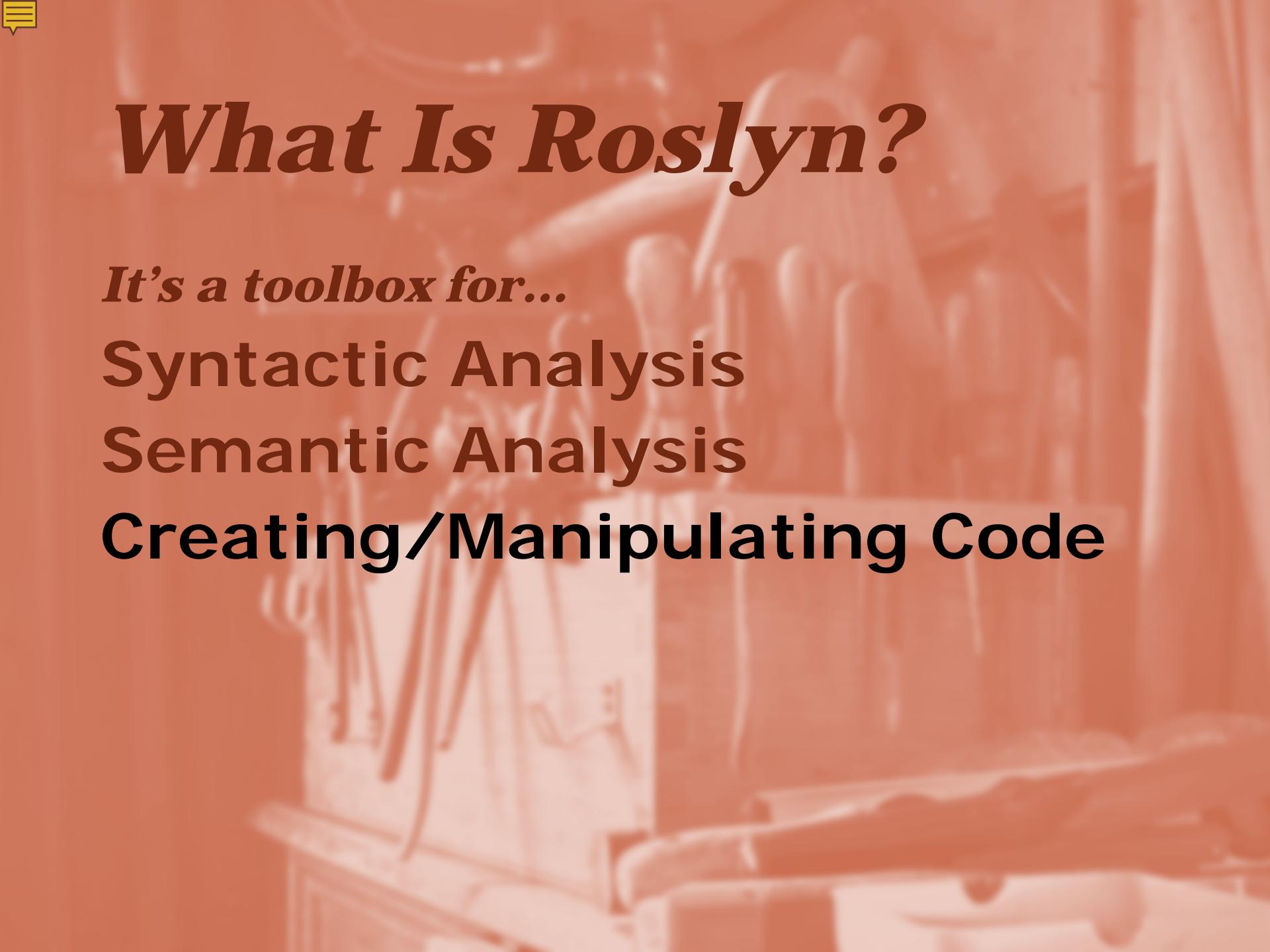
A close-up, slightly blurred photograph of a person's hands resting on a light-colored wooden keyboard. The hands are positioned as if ready to type. The background is dark and out of focus.

What Is Roslyn?

It's a toolbox for...

Syntactic Analysis

Semantic Analysis



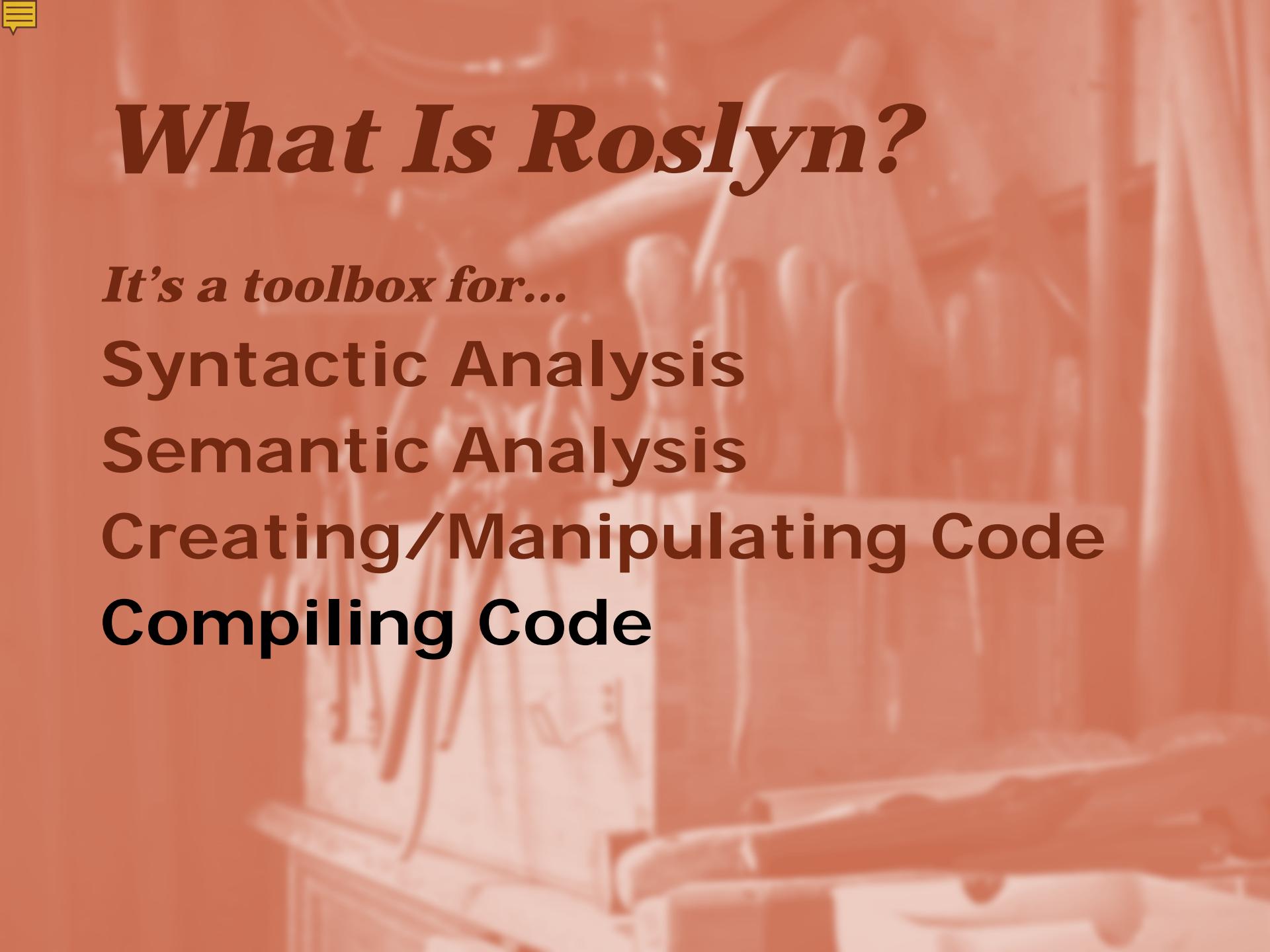
What Is Roslyn?

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...

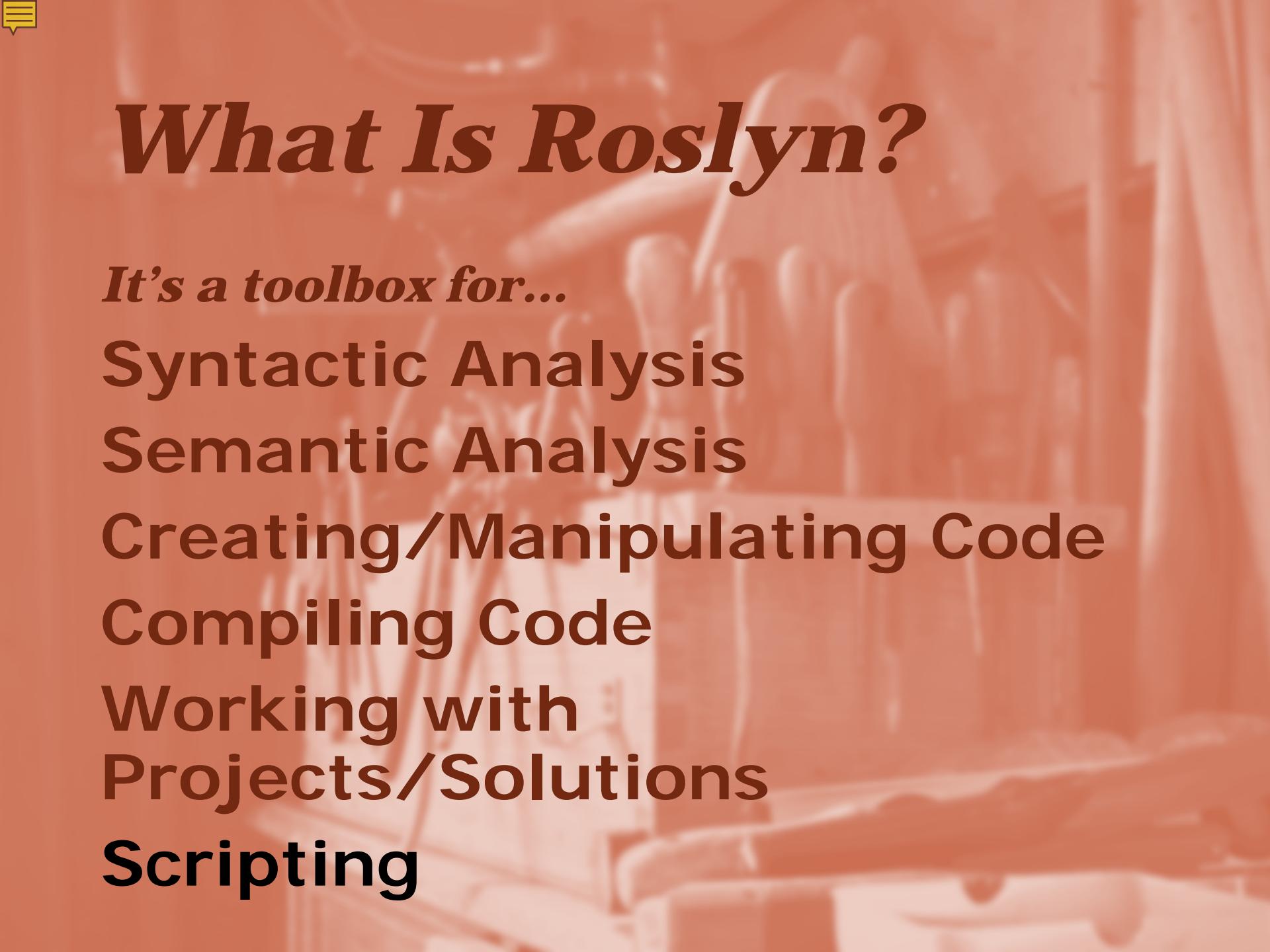
Syntactic Analysis

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

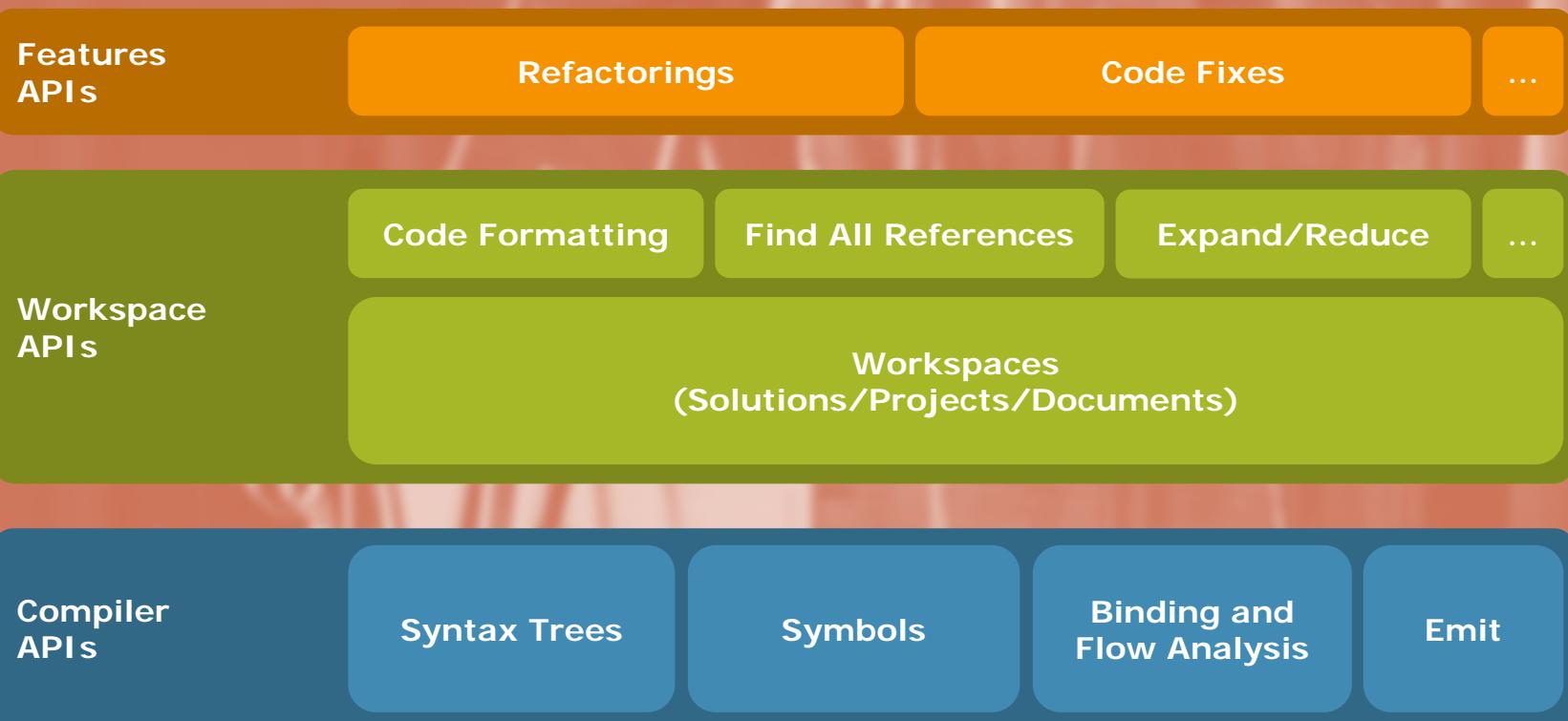
Compiling Code

**Working with
Projects/Solutions**

Scripting



What Is Roslyn?



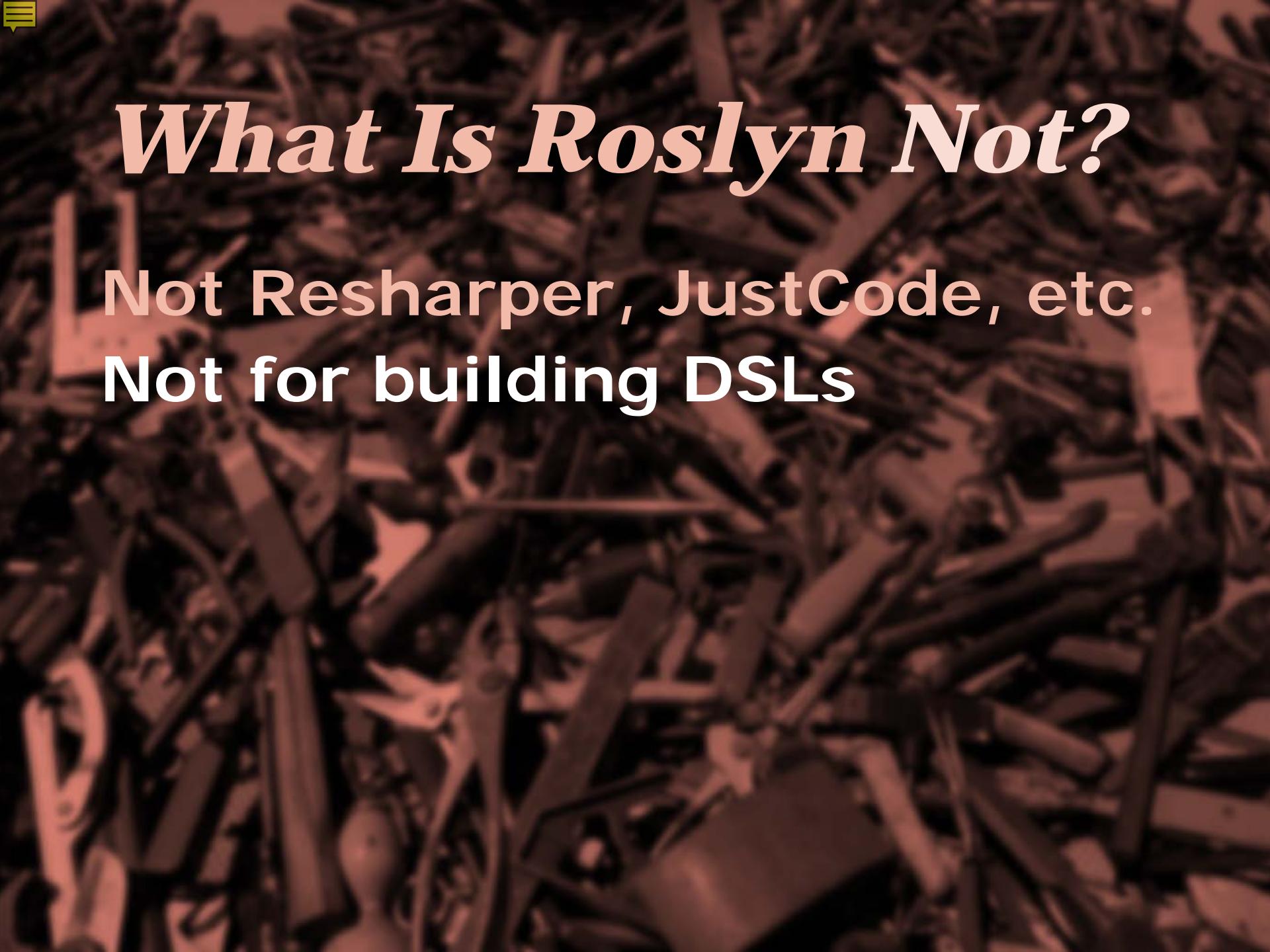
What Is Roslyn Not?





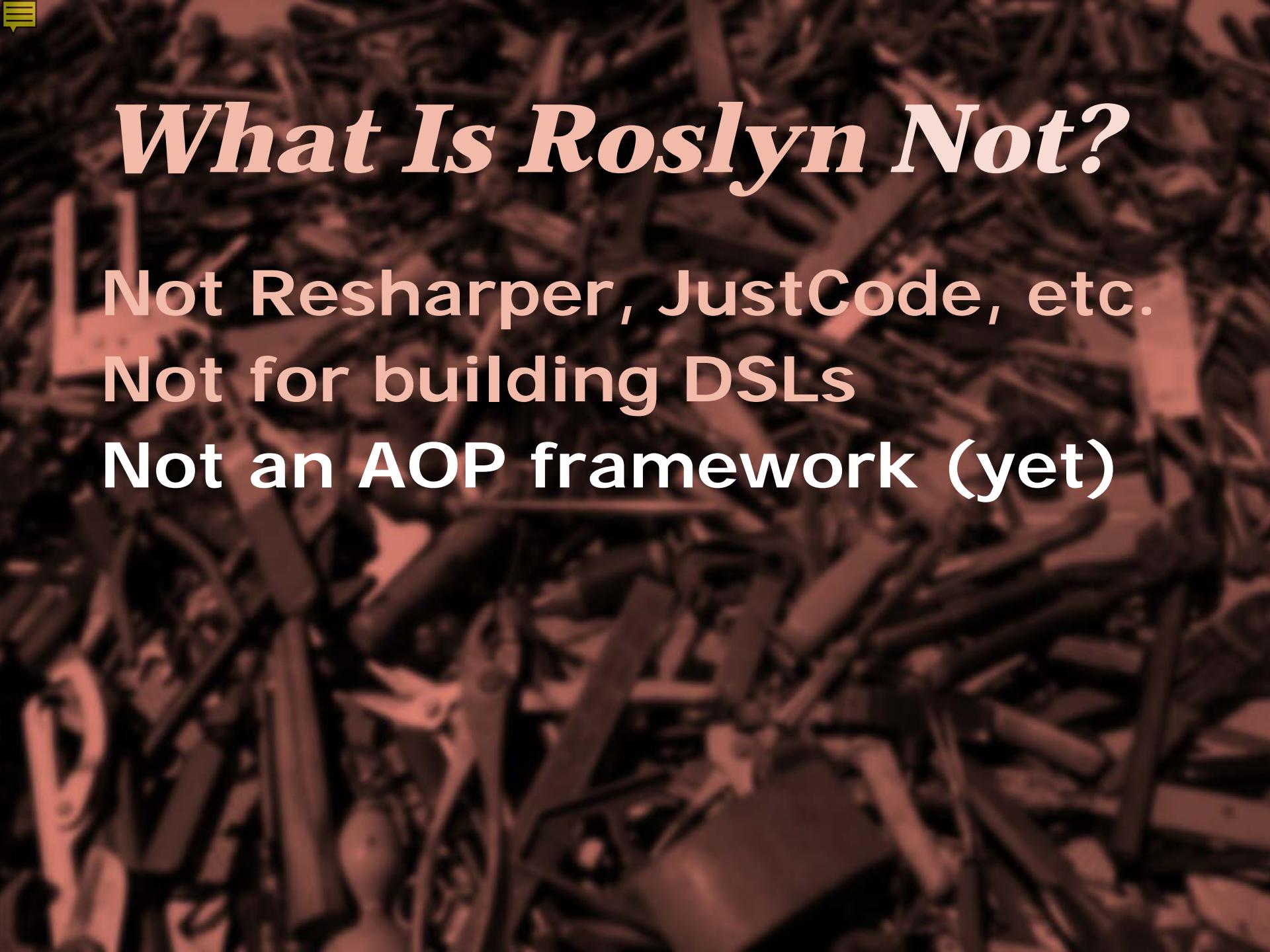
What Is Roslyn Not?

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





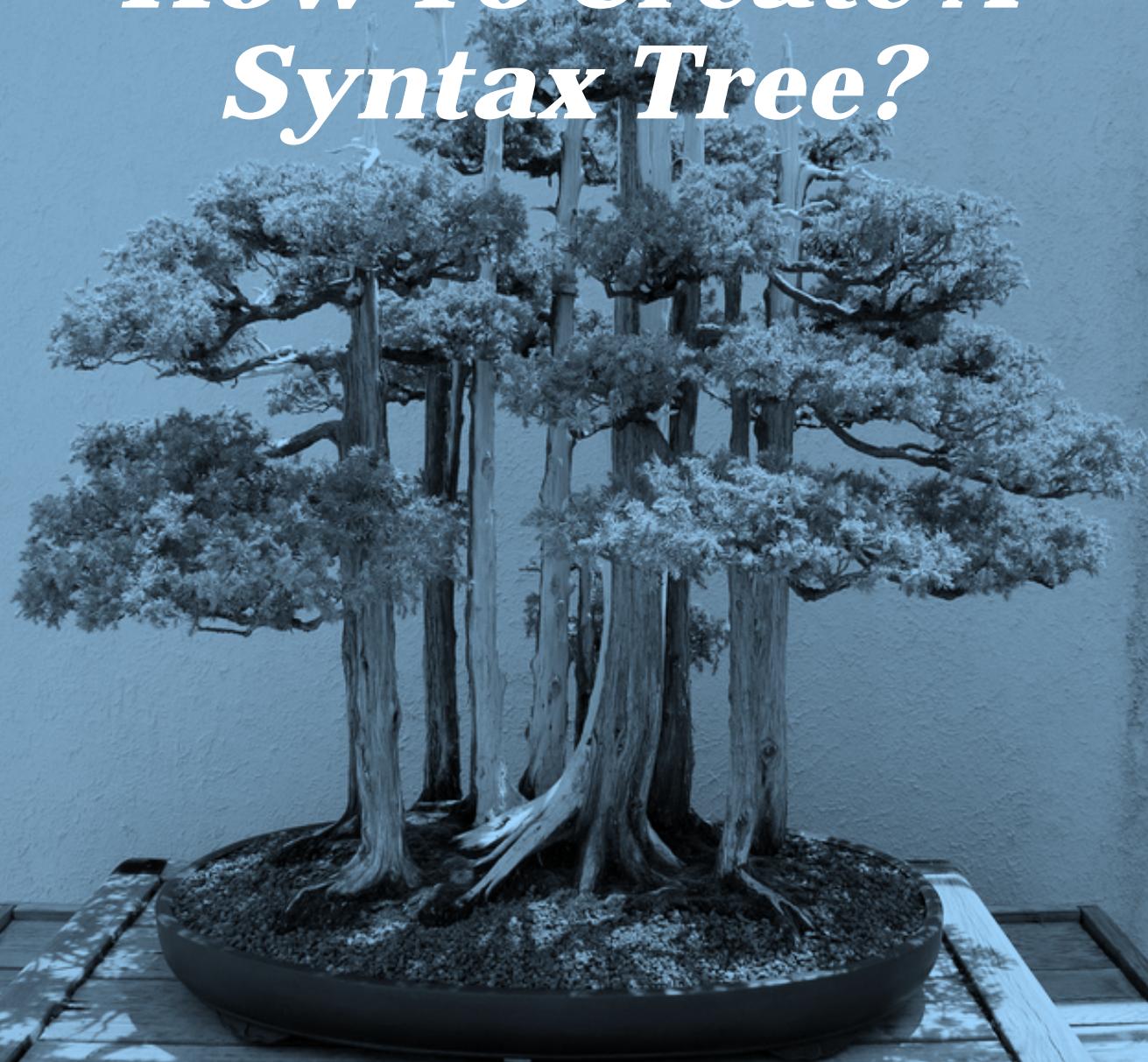
Syntax Trees



Nodes, Tokens, and Trivia, OH MY!



How To Create A Syntax Tree?





How To Create A Syntax Tree?

CSharpSyntaxTree.ParseText(...)



How To Create A Syntax Tree?

CSharpSyntaxTree.Create(...)

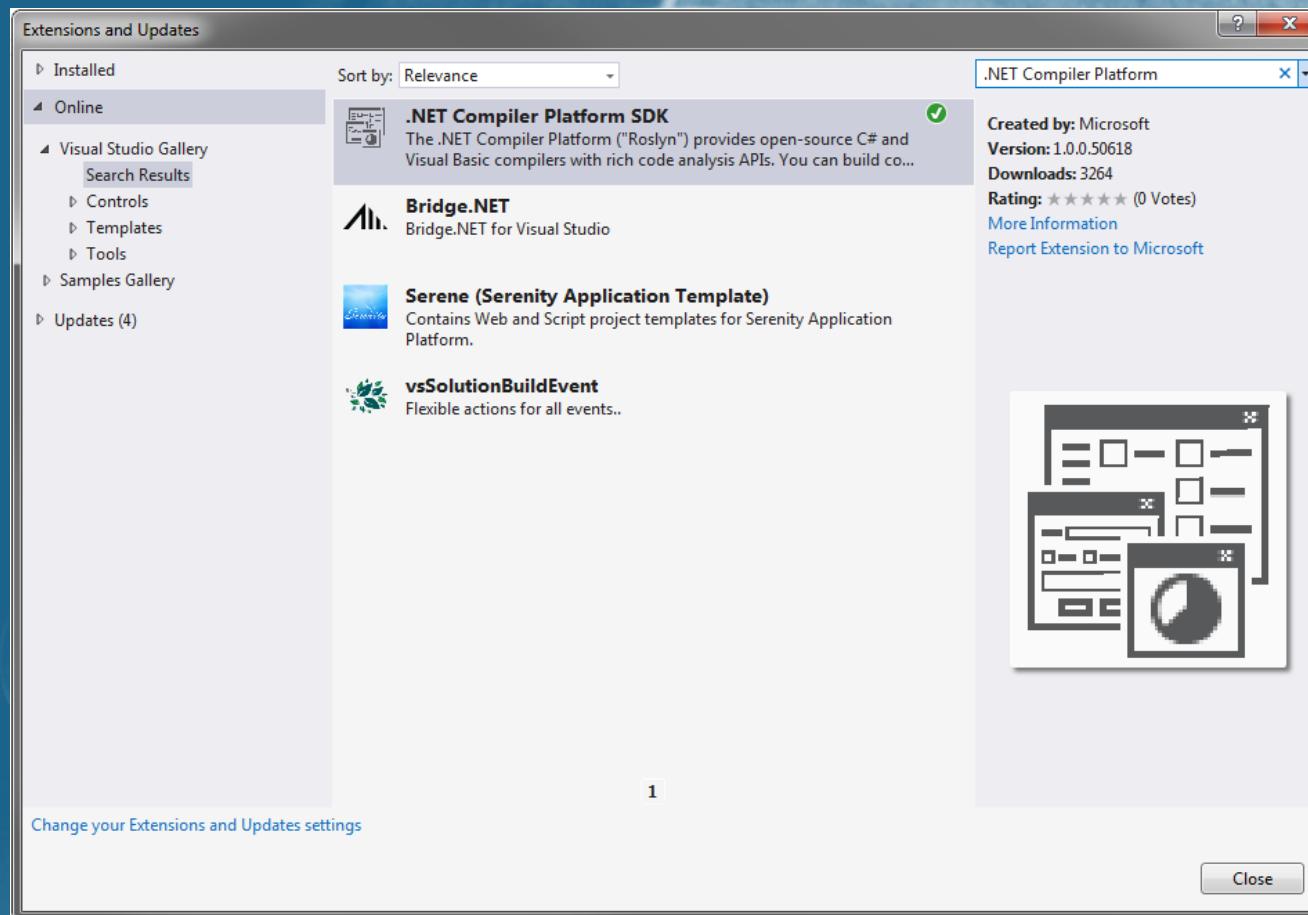


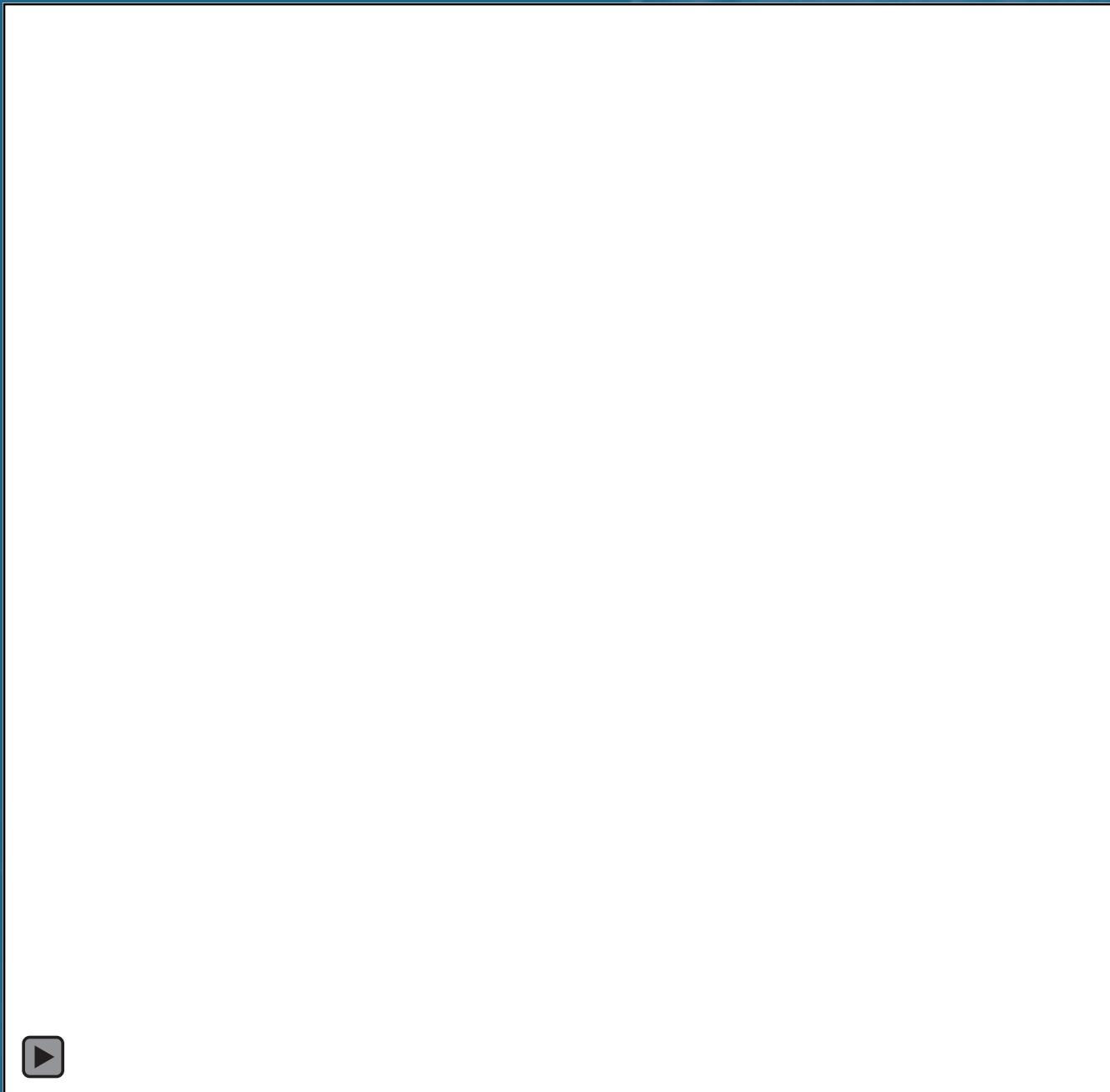
How To Create A Syntax Tree?

SyntaxFactory

Visual Studio Syntax Visualizer

Visual Studio Syntax Visualizer







Exploring the Tree





Exploring the Tree

LINQ-to-Objects

Exploring the Tree

CSharpSyntaxWalker

Manipulating the Tree



Manipulating the Tree

CSharpSyntaxRewriter



Compiling

Creating a Compilation

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?

.GetSymbolsWithNames(...)



Now What?

.GetSemanticModel(...)

Semantics





Symbols

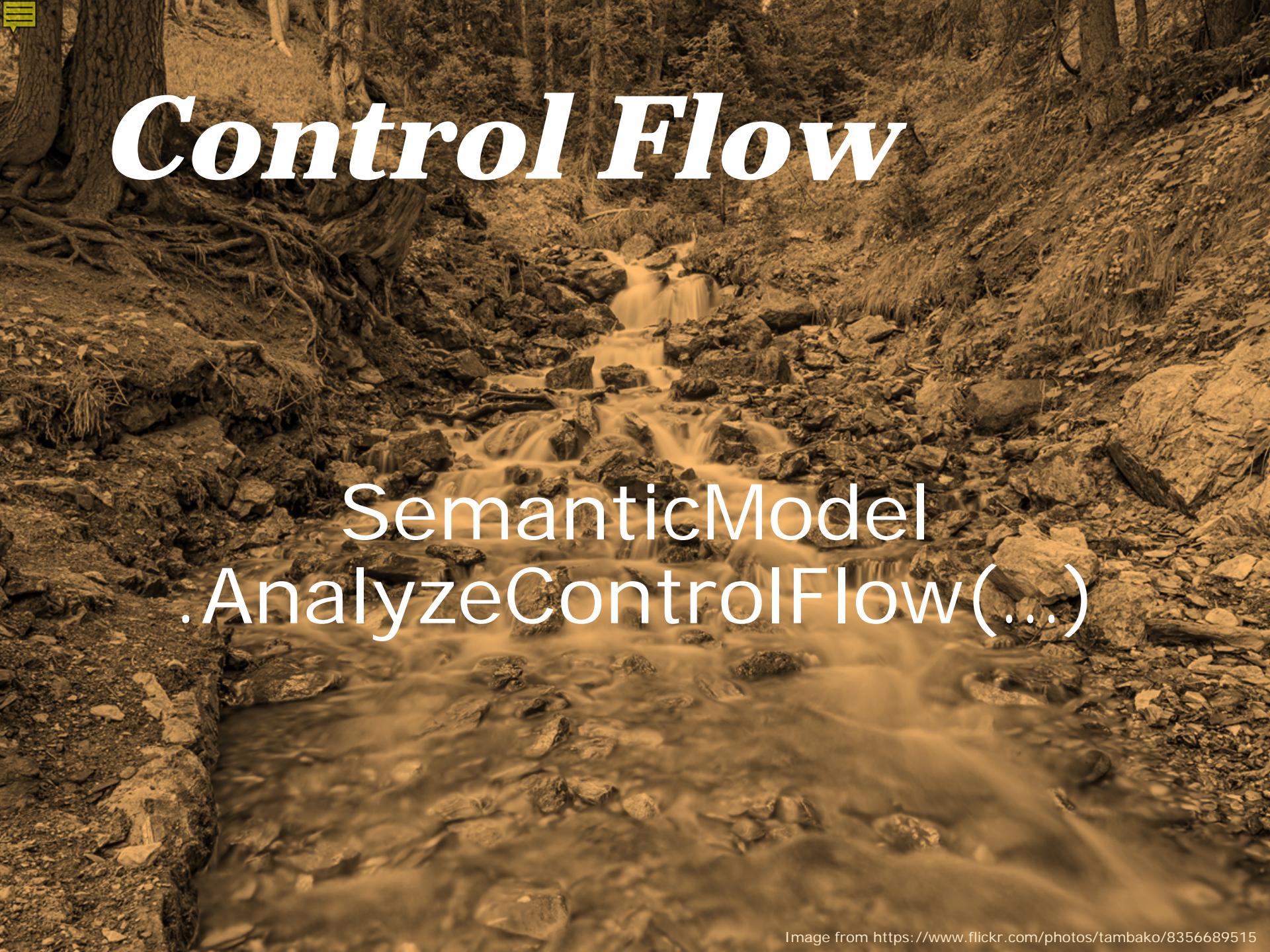
Symbols

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

Symbols

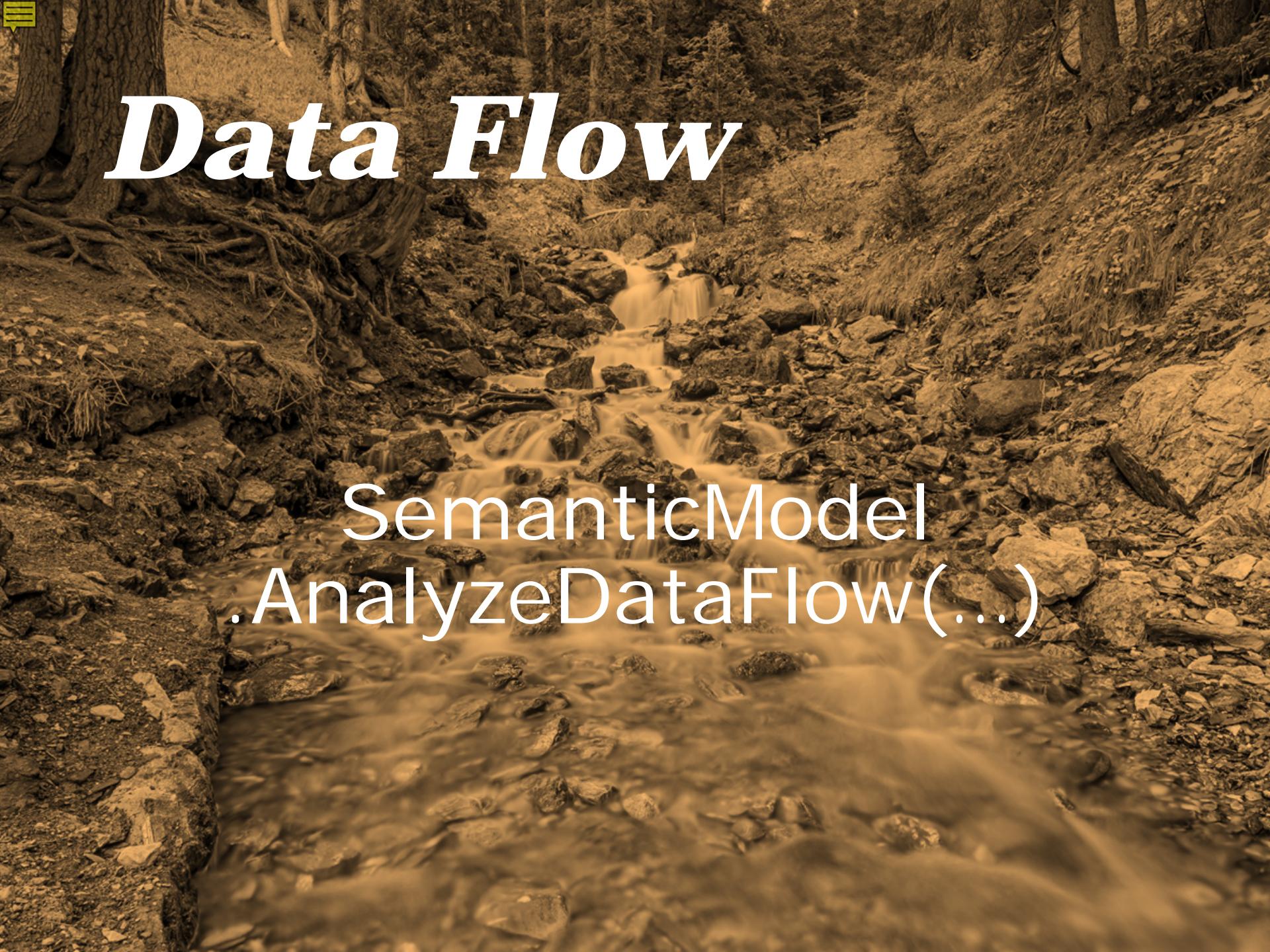
```
SemanticModel  
.GetSymbolInfo(...)
```





Control Flow

SemanticModel
.AnalyzeControlFlow(...)

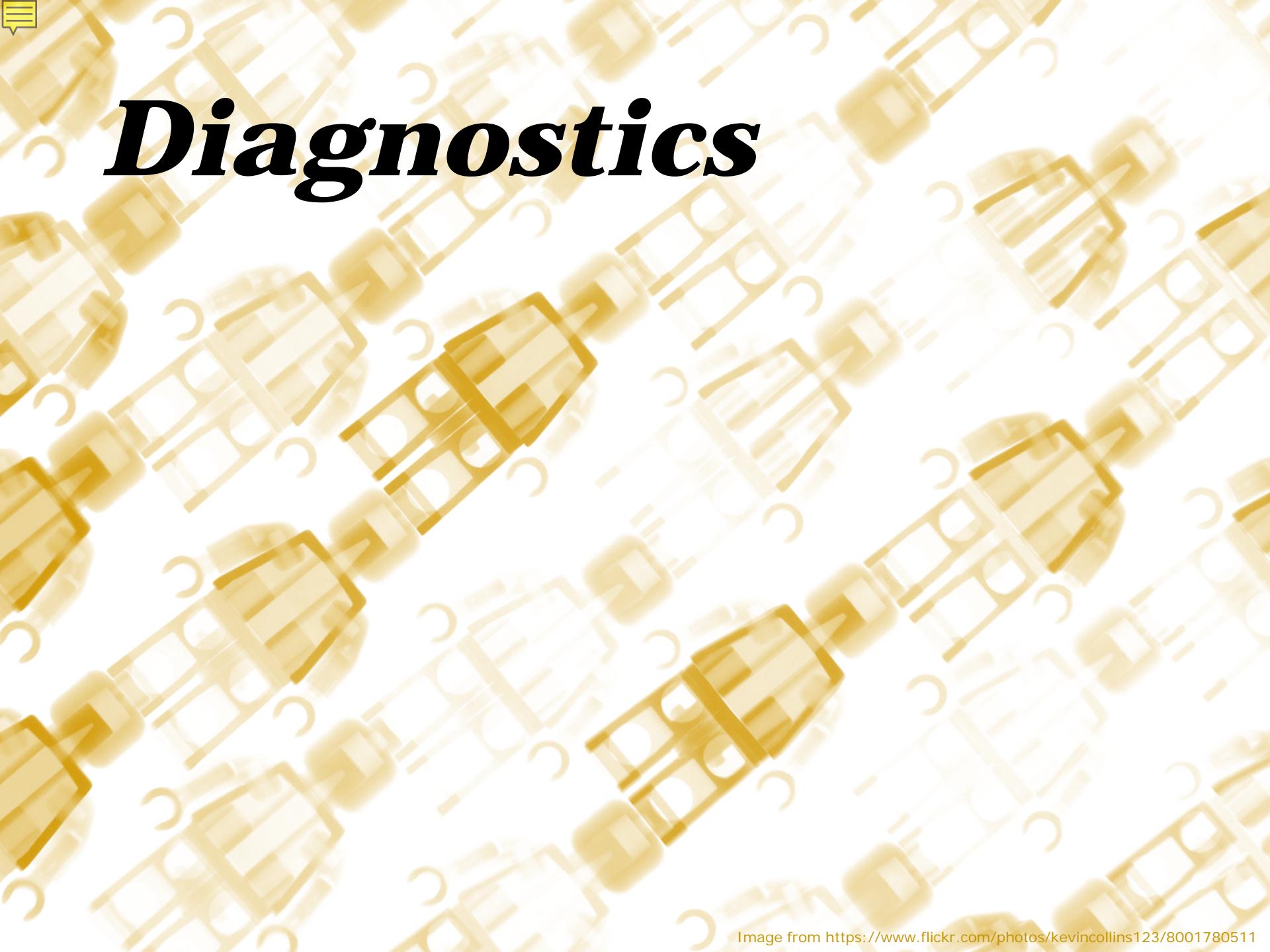


Data Flow

SemanticModel
.AnalyzeDataFlow(...)



Analyzers & Code Fixes



Diagnostics

Diagnostics

EmitResult.Diagnostics

Diagnostics

Compilation
.GetDiagnostics(...)

Analyzers

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

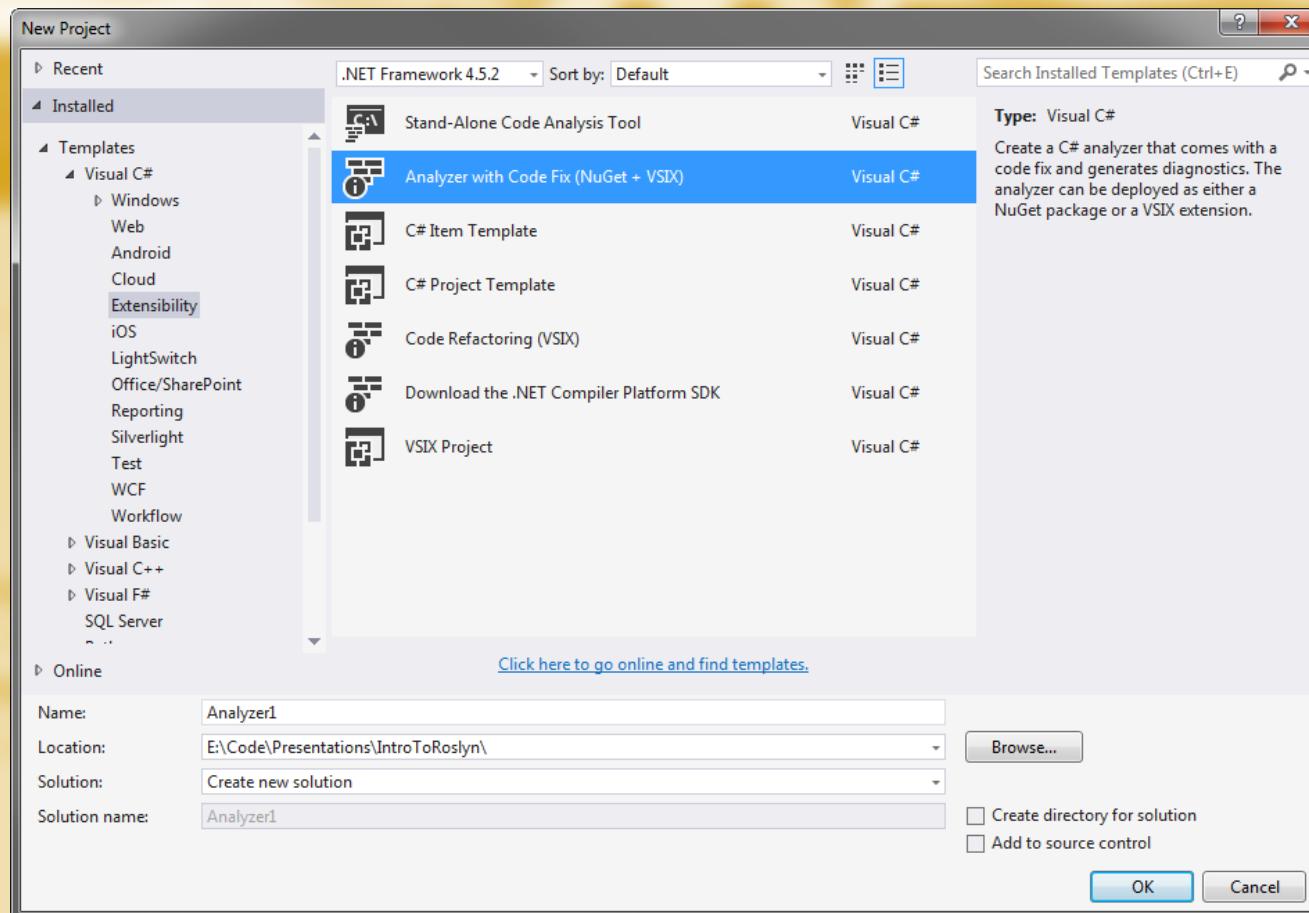
Analyzers

Compilation
.WithAnalyzers(...)

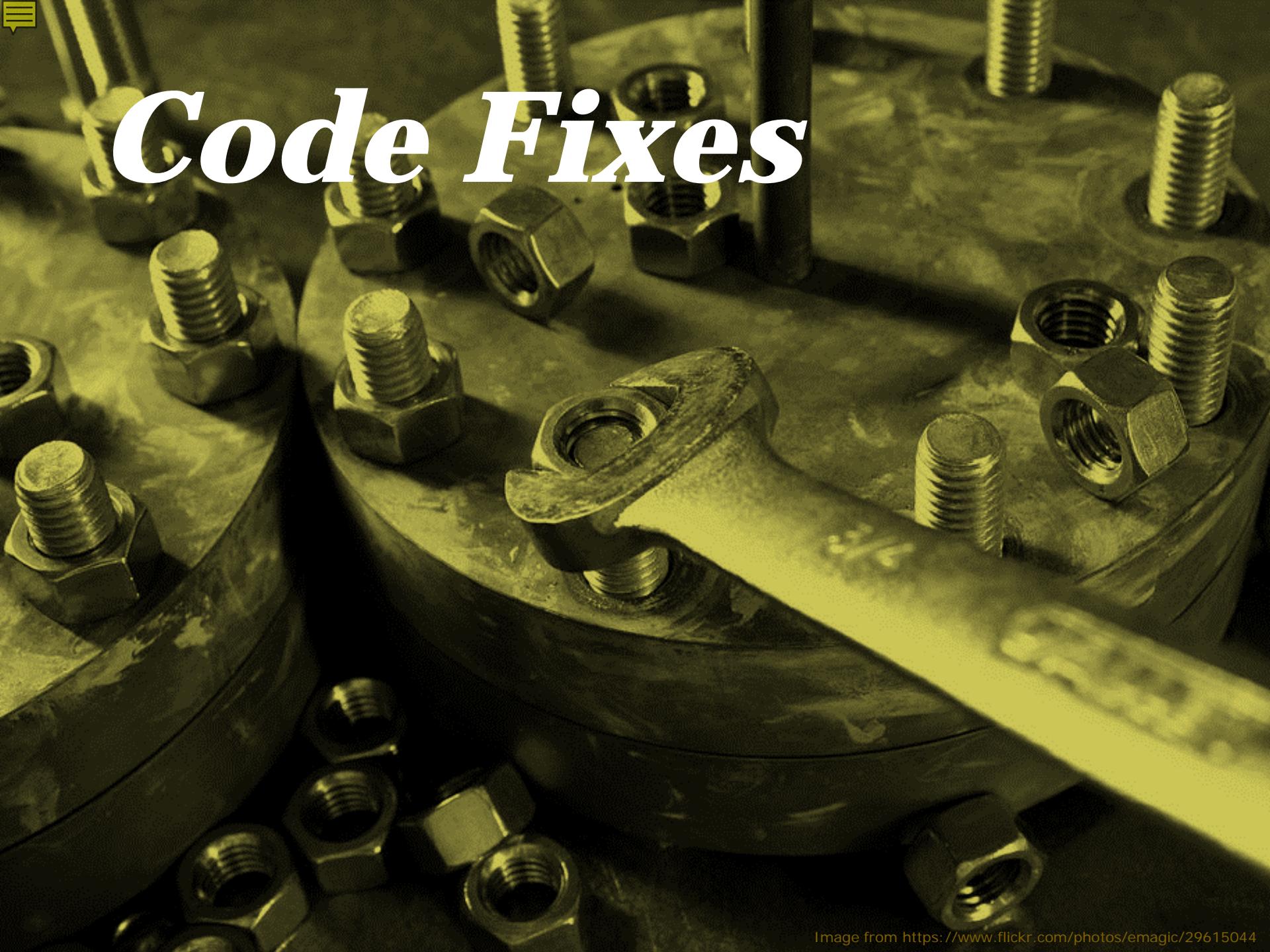
Analyzers

```
CompilationWithAnalyzers  
.GetAnalyzerDiagnosticsAsync()
```

Analyzers







Code Fixes

Code Fixes

Derived from

CodeFixProvider

Code Fixes

Must override

```
FixableDiagnosticIds { get; }
```

```
RegisterCodeFixesAsync(CodeFixContext)
```

Optionally

```
GetFixAllProvider()
```





Scripting



*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>