.NET Roslyn Compiler

•••

MAGIC NIGHT 2016/09/13

Announcements

- AWS Magic Night (IoT): September 20th
- AWS Magic Night (Mobile Notifications with AWS SNS): October 18th

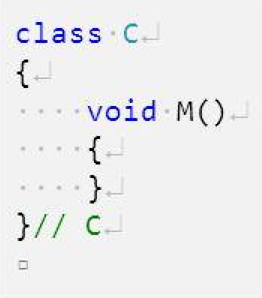
Magic Night Format

- Hands on experience
- Introduce a challenge
- Team up (random groups)
- Collaborate
- One hour warning
- Showcase your work

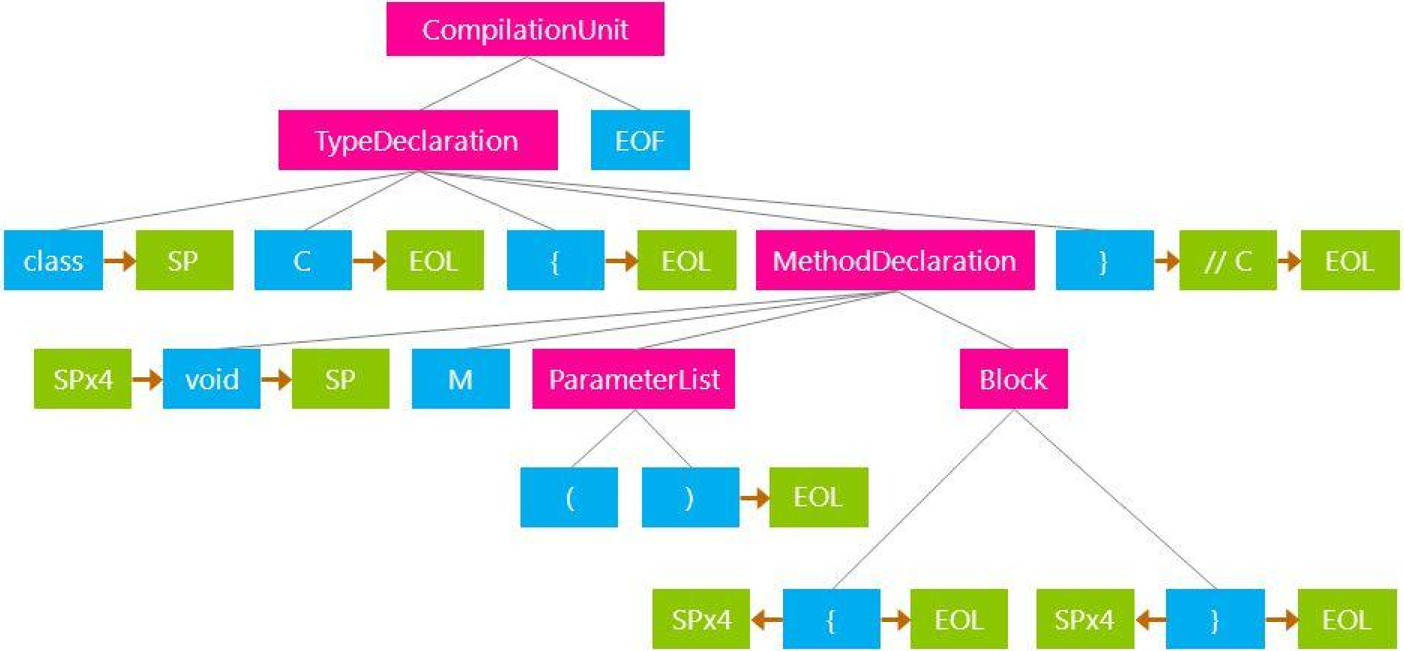
What is “Roslyn”?

- A set of open-source compilers and code analysis API’s for C#
- Lexical analysis
- Semantic analysis
- Dataflow analysis
- Dynamic compilation
- Code emission
- Visual Studio integration (issue detection and code fix)

Syntax Tree API – Trivia



```
class C
{
    void M()
    {
    }
}// C
```



API’s (https://github.com/dotnet/roslyn/wiki)

- Syntax API (SyntaxTree)
- Compilation API
  - SemanticModel
    - “What names are in scope at this location?”
    - “What members are accessible from this method?”
    - “What variables are used in this block of text?”
    - “What does this name/expression refer to?”
  - model.GetSymbolInfo()
  - model.GetTypeInfo()

Challenge

```
public static void Log(
        DateTime date,
        string formatString,
        params string[] arguments
) {
      Console.WriteLine(date + ": " + string.Format(formatString, arguments));
    }


static void Main(string[] args) {
      Log(DateTime.Now, "Hello, my name is {0} {1}", "Yuri");
    }
```

Roslyn to the rescue!

```
[AttributeUsage(AttributeTargets.Method)]
public class StringFormat : Attribute {

    //--- Fields ---
    public readonly string FormatParam;
    public readonly string ArgsParam;

    //--- Constructors ---
    public StringFormat(string formatParam = "format", string argsParam =
"args") {
        this.FormatParam = formatParam;
        this.ArgsParam = argsParam;
    }
}
```

```
[StringFormat(
        formatParam: "formatString",
        argsParam: "arguments"
)]
public static void Log(
        DateTime date,
        string formatString,
        params string[] arguments
) {
    Console.WriteLine(date + ": " + string.Format(formatString, arguments));
}
```

Demo

Challenge

1. Validate all method invocations of your “Log” method have a matching number of arguments between the format string and the argument list.

Extra Credit:

1. Validate that the attribute is being used correctly, meaning that the arguments that are specified in it, actually exist in the Log( ) method declaration.
2. Handle more difficult cases where the string is not directly provided in the Log( ) call, but instead a variable is being passed in.
3. Whatever you want to work on :)

Recommended Steps

1. Inspect all method invocations
2. For each method invocation, find the declaration of the method (Semantic Analysis)
3. Check if the declaration has our special “StringFormat” attribute
4. Find which arguments correspond to the “format” and the “args” arguments (positionally is easiest)
5. Validate those arguments in the current method invocation

References

1. Code: https://github.com/DotNetMagicNight/MagicNightRoslyn
2. Docs: https://github.com/dotnet/roslyn/wiki
3. Tutorials: https://msdn.microsoft.com/en-us/library/mt162308.aspx
4. Syntax Visualizer: https://roslyn.codeplex.com/wikipage?title=Syntax%20Visualizer

Have Fun!