Creating More Complex Diagnostics

Kathleen Dollard

@kathleendollard | www.CodeRapid.com
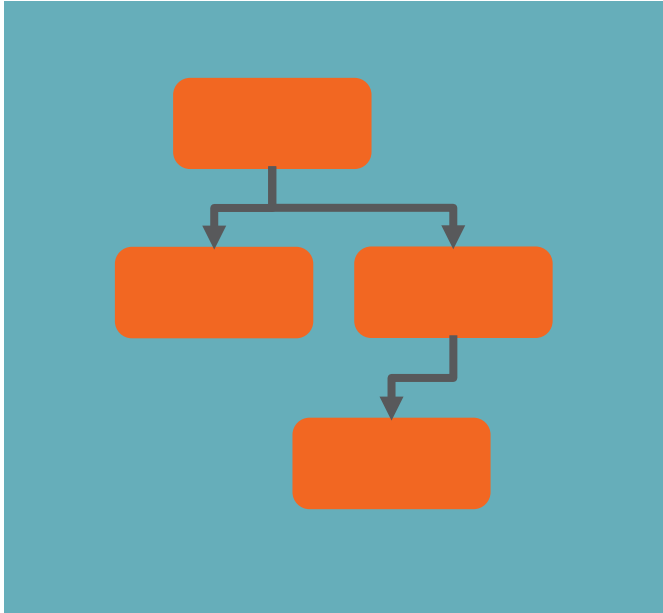
Using the Semantic Model

Why Seal Attribute Classes?

Performance:

- GetCustomAttribute searches for children

Security:

- Derived classes constructors run when retrieving assembly metadata

Correctness:

- Attributes on attribute can be changed in derived classes
  - Allowing multiples
  - Allowing in unexpected locations (example: class attribute on property)

```
[TestMethod]
[TestCategory("Analyzers")]
public void Empty_string_should_not_give_diagnostic()
```

```
[TestMethod(), TestCategory(testCategory: "Analyzers")]
public void Empty_string_should_not_give_diagnostic()
```

Attributes in Syntax Tree	Attributes in Semantic Tree
▪ Many variations, must roundtrip	▪ Samples above are identical
▪ Uses AttributeSyntax nested in AttributeListSyntax	▪ Simple list of attributes
▪ Names not resolved	▪ Names are resolved
	▪ Full inheritance hierarchy is available

Using Roslyn Services

Why Make Local Variables Constant?

Allows Optimization

Communicates intent

C#

Language Specification
Version 5.0

7.1.9 Constant expressions . .

8.5 Declaration statements
8.5.1 Local variable declarations
8.5.2 Local constant declarations

- Eeek! Lots of stuff!

- Single type or var keyword
- Initializer is optional
- One or more variables
- Types can be aliased

- Must have initializer
- Initializer must be first
- const must be first word
- var can't be used

Data Analysis Service

- Only local variables, parameters, and "this"

- Considers a statement, expression, or set of statements

- Available methods are:
  - Succeeded
  - AlwaysAssigned
  - Captured
  - DataFlowsIn
  - DataFlowsOut
  - ReadInside
  - ReadOutside
  - UnsafeAddressTaken
  - VariablesDeclared
  - WrittenInside
  - WrittenOutside

Debugging Analyzers

Summary

Semantic Model	▪ Declarations are resolved in the semantic model ▪ The inheritance hierarchy is available ▪ Attributes are a single simple list
Services	▪ DataFlowAnalysis is one of several services available ▪ Services provide difficult functionality, within limits
Helper Methods	▪ Helper methods support reuse, especially minutiae like managing trivia or formatting ▪ Well-named methods clarify even complex analysis