Project Roslyn

Exposing the C# and Visual Basic compilers’ code analysis

Dustin Campbell
Senior Program Manager
Microsoft
Project Codename “Roslyn”

• Scope
  • Managed compilers & language services
  • Code analysis APIs
  • Scripting APIs
  • Language service extensibility
  • Read-Eval-Print-Loop (REPL)

• Ship Date
  • v.Next + n, where n > 0
Language Service

Compiler API

Compiler Pipeline

Parser
Metadata Import
Symbols
Binder
IL Emitter

Formatter
Colorizer
Outlining
Navigate To
Object Browser
Completion List
Find All
References
Rename
Quick Info
Signature Help
Extract Method
Go To Definition
Edit and Continue

Navigate To
Object Browser
Completion List
Find All
References
Rename
Quick Info
Signature Help
Extract Method
Go To Definition
Edit and Continue
Design Choices

- Immutable
- Complete
- Resilient
- Efficient
Demo: Compiler APIs
Major Players
Syntax Nodes

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

SyntaxTree
GetRoot

CompilationUnit

TypeDeclaration

MethodDeclaration

ParameterList
Block
Major Players
Syntax Tokens

class C
{
    void M();
}
} // C
Major Players
Syntax Tokens

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

EOF
```cpp
class C {
  void M();
};

// C
```
Major Players

Compilations, References and Symbols

Compilation
- Symbols
- Diagnostics
- Emit

SyntaxTrees

References
Major Players

Semantics

GetSemanticModel(tree)

Compilation

SemanticModel

- Binding Information
- Conversion Classification
- Flow Analysis
- Overload Resolution
- Etc.
Major Players

Workspace, Solutions, Projects and Documents

- Host Environment
  - Events (e.g. key presses)
  - Apply
  - Workspace
  - Solution
  - Solution
  - Solution
  - Project
  - Project
  - Project
  - Document
  - Document

- Compilation
  - SyntaxTree
Demo: Building a Quick Fix
Next Steps

- Install the Roslyn CTP
- Try out the APIs
- Give us feedback!

Dev Center:
http://msdn.com/roslyn

Forum:

Email:
dustinca@microsoft.com