# **Using Roslyn (consuming Static and Sematic data)**

Executing scripts inside document:

Exposing the C# and VB compiler's code analysis <a href="http://msdn.microsoft.com/en-us/hh500769#Toc306015665">http://msdn.microsoft.com/en-us/hh500769#Toc306015665</a>

## 3.2 Obtaining a Syntax Tree

```
Inspector
 Command To Execute
                                                                                     Invoke and Result
      SyntaxTree tree = SyntaxTree.ParseCompilationUnit(
                                                                                                Execute
                       @"using System;
                                                                                     stop execution
                       namespace HelloWorld
                                                                                      Output
                           class Program
                                                                                      ● A
Z↓
                                                                                      static void Main(string[] args)
                                                                                         FileName
                                                                                         Options
                                                                                                 Roslyn.Compilers.
                                    Console.WriteLine(""Hello, World!"");
                                                                                         Root
                                                                                                 using System;
                                                                                         Text
   14 return tree;
   16 //using Roslyn.Compilers.CSharp;
   17 //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
   18 //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
                                                                                      FileName
```

```
ExpressionSyntax myExpression = Syntax.ParseExpression("1+1");
return myExpression;
//using Roslyn.Compilers.CSharp;
//02Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
//02Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
```

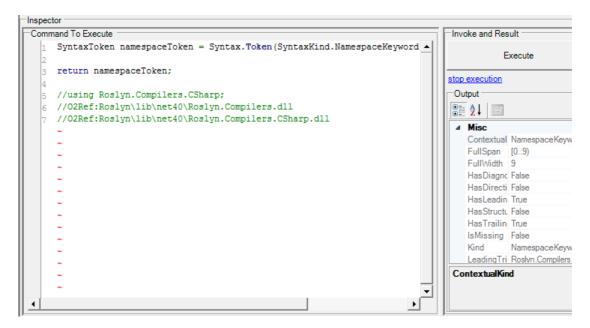
```
Inspector
                                                                                        Invoke and Result
Command To Execute
      ExpressionSyntax myExpression = Syntax.ParseExpression("1+1");
                                                                                                   Execute
     return myExpression;
                                                                                        stop execution
     //using Roslyn.Compilers.CSharp;
     //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
                                                                                        Output
     //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
                                                                                        FullSpan [0..3)
                                                                                            FullWidth 3
                                                                                            HasAnnota False
                                                                                            HasChildre True
                                                                                            HasDiagnc False
                                                                                            HasDirecti False
                                                                                            HasLeadin False
                                                                                            HasStructu False
                                                                                            HasTrailin False
                                                                                            IsMissing False
                                                                                            IsStructure False
                                                                                        FullSpan
```

```
StatementSyntax myStatement = Syntax.ParseStatement("for (int i = 0; i <
length; i++) { }");
return myStatement;
//using Roslyn.Compilers.CSharp;
//O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
//O2Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll</pre>
```



## 3.3 Manipulating Syntax Nodes

```
SyntaxToken namespaceToken = Syntax.Token(SyntaxKind.NamespaceKeyword);
return namespaceToken;
//using Roslyn.Compilers.CSharp;
//O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
```



```
Inspector
                                                                                      Invoke and Result
Command To Execute
     //the new 4.0 parameter syntax doesn't work in this O2 REPL environmen .
                                                                                                 Execute
     /*QualifiedNameSyntax namespaceName = Syntax.QualifiedName(
                                  left: Syntax. IdentifierName ("Roslyn"),
                                                                                      stop execution
                                   right: Syntax.IdentifierName("Compilers"))
     //so we have to do this instead
                                                                                       Output
     QualifiedNameSyntax namespaceName = Syntax.QualifiedName(
                                                                                       e di di
                                   Syntax. IdentifierName ("Roslyn"),
                                                                                       default (SyntaxToken),
                                                                                          Arity
                                   Syntax.IdentifierName("Compilers"));
                                                                                          DotToken
                                                                                          FullSpan (0.16)
                                                                                          FullWidth 16
   12 return namespaceName;
                                                                                          HasAnnota False
                                                                                          HasChildre True
   14 //using Roslyn.Compilers
                                                                                          HasDiagno False
   15 //using Roslyn.Compilers.CSharp;
                                                                                          HasDirecti False
   16 //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
                                                                                          HasLeadin False
   17 //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
                                                                                          HasStructi, False
                                                                                          HasTrailin False
                                                                                       Arity
```

```
NameSyntax namespaceName = Syntax.ParseName("Roslyn.Compilers");
return namespaceName;
//using Roslyn.Compilers
//using Roslyn.Compilers.CSharp;
//02Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
//02Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
```

```
Inspector
Command To Execute
                                                                                        Invoke and Result
     NameSyntax namespaceName = Syntax.ParseName("Roslyn.Compilers");
                                                                                                    Execute
     return namespaceName;
                                                                                        stop execution
                                                                                         Output
     //using Roslyn.Compilers
     //using Roslyn.Compilers.CSharp;
                                                                                         A ↓
      //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
                                                                                         //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
                                                                                            Arity
                                                                                            DotToken
                                                                                            FullSpan
                                                                                                     [0..16)
                                                                                            FullWidth 16
                                                                                            HasAnnota False
                                                                                            HasChildre True
                                                                                            HasDiagnc False
                                                                                            HasDirecti False
                                                                                            HasLeadin False
                                                                                            HasStructu False
                                                                                            HasTrailin False
                                                                                         Arity
```

```
return myNamespace.GetFullText();
//using Roslyn.Compilers
//using Roslyn.Compilers.CSharp;
//O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
//O2Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
```



```
Inspector
Command To Execute
                                                                                    Invoke and Result
     SyntaxToken namespaceToken = Syntax.Token(
                                                                                              Execute
                                   SyntaxKind.NamespaceKeyword,
                                   Syntax.Space);
                                                                                   stop execution
     NameSyntax namespaceName = Syntax.ParseName("Roslyn.Compilers");
                                                                                    Output
     NamespaceDeclarationSyntax myNamespace =
                                                                                    namespace Roslyn.Compilers
         Syntax.NamespaceDeclaration(namespaceToken, namespaceName);
     NamespaceDeclarationSyntax formattedNamespace = myNamespace.Format();
  10 return formattedNamespace.GetFullText();
  13 //using Roslyn.Compilers
  14 //using Roslyn.Compilers.CSharp;
  15 //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
  16 //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
```

## 3.3.2 Modifying an existing node

```
Inspector
                                                                                        Invoke and Result
Command To Execute
      SyntaxTree tree = SyntaxTree.ParseCompilationUnit(
                                                                                                   Execute
                       @"using System;
                                                                                       stop execution
                       namespace HelloWorld
                                                                                        Output
                            class Program
                                                                                        HasDirecti False
                                static void Main(string[] args)
                                                                                           HasLeadin True
                                                                                           HasStructu False
                                    Console.WriteLine(""Hello, World!"");
                                                                                           HasTrailin False
                                                                                           IsMissing False
                                                                                           IsStructure False
                       1");
                                                                                                    NamespaceDecla
                                                                                           Members Roslyn.Compilers
   15 NamespaceDeclarationSyntax oldNamespace = tree.Root.DescendentNodes()
                                                                                           Name HelloWorld
                                  .OfType<NamespaceDeclarationSyntax>()
                                                                                           Namespac
                                                                                                            names
                                  .First():
                                                                                           OpenBrace
                                                                                           Parent using System;
   19 return oldNamespace;
                                                                                        Name
   21 //using Roslyn.Compilers
```

```
SyntaxTree tree = SyntaxTree.ParseCompilationUnit(
                @"using System;
                namespace HelloWorld
                    class Program
                        static void Main(string[] args)
                            Console.WriteLine(""Hello, World!"");
NamespaceDeclarationSyntax oldNamespace = tree.Root.DescendentNodes()
                          .OfType<NamespaceDeclarationSyntax>()
                          .First();
NameSyntax namespaceName = Syntax.ParseName("Roslyn.Compilers");
NamespaceDeclarationSyntax newNamespace =
                    oldNamespace. Update (oldNamespace.NamespaceKeyword,
                                       namespaceName,
                                       oldNamespace.OpenBraceToken,
                                       oldNamespace.Externs,
                                       oldNamespace.Usings,
                                       oldNamespace.Members,
                                       oldNamespace.CloseBraceToken,
                                       oldNamespace.SemicolonTokenOpt);
return oldNamespace.Format().GetFullText() +
          "\n\n +
        newNamespace.Format().GetFullText();
//using Roslyn.Compilers
 //using Roslyn.Compilers.CSharp;
//O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
//O2Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
```

```
Inspector
Command To Execute
                                                                                Invoke and Result
                                                                          -
                                                                                                Execute
  19 NameSyntax namespaceName = Syntax.ParseName("Roslyn.Compilers
                                                                                stop execution
  21 NamespaceDeclarationSyntax newNamespace =
                                                                                 Output
                   oldNamespace.Update(oldNamespace.NamespaceKeyword
                                                                                namespace HelloWorld
                                                 namespaceName,
                                                 oldNamespace.OpenBrace
                                                                                  class Program
                                                 oldNamespace.Externs,
                                                                                     static void Main(string[] args)
                                                 oldNamespace.Usings,
                                                 oldNamespace.Members,
                                                                                      Console.WriteLine("Hello, World!");
                                                 oldNamespace.CloseBrac
                                                 oldNamespace.Semicolor
  29
  31 return oldNamespace.Format().GetFullText() +
                                                                                  amespace Roslyn.Compilers
               "\n\n" +
   33
             newNamespace.Format().GetFullText();
                                                                                  class Program
  34
                                                                                    static void Main(string[] args)
  35 //using Roslyn.Compilers
  36 //using Roslyn.Compilers.CSharp;
                                                                                      Console.WriteLine("Hello, World!");
   37 //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
    8 //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
```

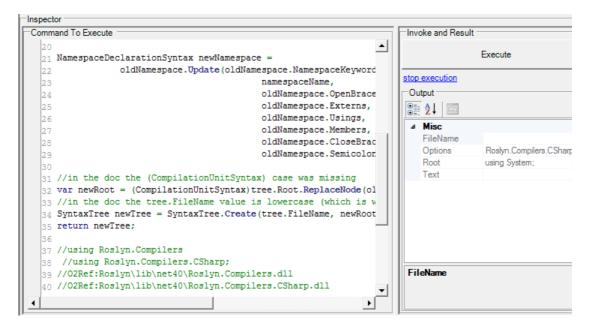
### 3.3.3 Replacing a node

```
SyntaxTree tree = SyntaxTree.ParseCompilationUnit(
                @"using System;
                namespace HelloWorld
                    class Program
                        static void Main(string[] args)
                            Console.WriteLine(""Hello, World!"");
NamespaceDeclarationSyntax oldNamespace = tree.Root.DescendentNodes()
                           .OfType<NamespaceDeclarationSyntax>()
                           .First();
NameSyntax namespaceName = Syntax.ParseName("Roslyn.Compilers");
NamespaceDeclarationSyntax newNamespace =
                    oldNamespace. Update (oldNamespace.NamespaceKeyword,
                                        namespaceName,
                                        \verb"oldNamespace.OpenBraceToken",
                                        oldNamespace.Externs,
                                        oldNamespace. Usings,
                                        oldNamespace.Members,
                                        oldNamespace.CloseBraceToken,
                                        oldNamespace.SemicolonTokenOpt);
SyntaxNode newRoot = tree.Root.ReplaceNode(oldNamespace, newNamespace);
return newRoot.Format().GetFullText();
//using Roslyn.Compilers
 //using Roslyn.Compilers.CSharp;
//O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
//O2Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
```

```
Inspector
                                                                               Invoke and Result
Command To Execute
                                                                         •
                                                                                               Execute
  19 spaceName = Syntax.ParseName("Roslyn.Compilers");
                                                                               stop execution
  21 ationSyntax newNamespace =
                                                                                Output
  22 Namespace.Update(oldNamespace.NamespaceKeyword,
                                                                               usina System:
                                namespaceName,
                                oldNamespace.OpenBraceToken,
                                                                                  nespace Roslyn.Compilers
                                oldNamespace.Externs,
                                                                                 class Program
                                oldNamespace.Usings,
                                oldNamespace.Members,
                                                                                   static void Main(string[] args)
                                oldNamespace.CloseBraceToken,
                                oldNamespace.SemicolonTokenOpt);
  29
                                                                                     Console.WriteLine("Hello, World!");
  31 oot = tree.Root.ReplaceNode(oldNamespace, newNamespace);
   33 Format().GetFullText();
  35 Compilers
  36 .Compilers.CSharp;
   37 lib\net40\Roslyn.Compilers.dll
     lib\net40\Roslyn.Compilers.CSharp.dll
```

### 3.3.4 Creating a Syntax Tree

```
SyntaxTree tree = SyntaxTree.ParseCompilationUnit(
                @"using System;
                namespace HelloWorld
                    class Program
                        static void Main(string[] args)
                            Console.WriteLine(""Hello, World!"");
NamespaceDeclarationSyntax oldNamespace = tree.Root.DescendentNodes()
                          .OfType<NamespaceDeclarationSyntax>()
                          .First();
NameSyntax namespaceName = Syntax.ParseName("Roslyn.Compilers");
NamespaceDeclarationSyntax newNamespace =
                    oldNamespace. Update (oldNamespace. NamespaceKeyword,
                                       namespaceName,
                                       oldNamespace.OpenBraceToken,
                                       oldNamespace.Externs,
                                       oldNamespace.Usings,
                                       oldNamespace.Members,
                                       oldNamespace.CloseBraceToken,
                                       oldNamespace.SemicolonTokenOpt);
//in the doc the (CompilationUnitSyntax) case was missing
var newRoot = (CompilationUnitSyntax)tree.Root.ReplaceNode(oldNamespace,
newNamespace);
//in the doc the tree.FileName value is lowercase (which is wrong)
SyntaxTree newTree = SyntaxTree.Create(tree.FileName, newRoot,tree.Options);
return newTree;
//using Roslyn.Compilers
  //using Roslyn.Compilers.CSharp;
//O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
```



### 4.2 Obtaining a Compilation

```
Inspector
Command To Execute
                                                                              Invoke and Result
                                   Console.WriteLine(""Hello, World!
                                                                                             Execute
                               1
                                                                             stop execution
                                                                              Output
  12
                                                                              14 SyntaxTree tree = SyntaxTree.ParseCompilationUnit(sourceText)
                                                                              Assembly
                                                                                              HelloWorld, Version=0.0
  16 AssemblyFileReference mscorlib = new AssemblyFileReference(ty
                                                                                 DynamicType dynamic
                                                                                 GlobalNamespa <global namespace>
  18 Compilation compilation = Compilation.Create("HelloWorld",
                                                                                 GlobalUsings
                                                      null.
  1.9
                                                                                 HostObjectType
                                                      new [ tree ],
                                                                                 IsSubmission
                                                      new MetadataRefe
                                                                                 MetadataFilePrc (none)
                                                                                 ObjectType
                                                                                             object
  23 return compilation;
                                                                                              Roslyn.Compilers.CSharp
                                                                                 Options
                                                                                 Previous Submis (none)
  25 //using Roslyn.Compilers
                                                                                 ReferenceResol Roslyn, Compilers, Metada
  26 //using Roslyn.Compilers.CSharp;
  27 //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
                                                                              Assembly
  28 //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
```

```
var sourceText=@"using System;
                namespace HelloWorld
                    class Program
                        static void Main(string[] args)
                            Console.WriteLine(""Hello, World!"");
SyntaxTree tree = SyntaxTree.ParseCompilationUnit(sourceText);
AssemblyFileReference mscorlib = new AssemblyFileReference(typeof
(object).Assembly.Location);
Compilation compilation = Compilation.Create("HelloWorld")
                                     .AddReferences(mscorlib)
                                     .AddSyntaxTrees(tree);
return compilation;
//using Roslyn.Compilers
  //using Roslyn.Compilers.CSharp;
//O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
//O2Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
```

```
Inspector
                                                                               Invoke and Result
Command To Execute
      var sourceText=@"using System;
                                                                                              Execute
                       namespace HelloWorld
                                                                              stop execution
                            class Program
                                                                               Output
                                                                               static void Main(string[] args)
                                                                                Assembly
                                                                                                HelloWorld, Version=0.0
                                    Console.WriteLine(""Hello, World!
                                                                                  DynamicType dynamic
                                                                                  GlobalNamespa <global namespace>
                                                                                  GlobalUsings
                                                                                  HostObjectType
                                                                                  IsSubmission
  14 SyntaxTree tree = SyntaxTree.ParseCompilationUnit(sourceText)
                                                                                  MetadataFilePrc (none)
                                                                                  ObjectType
                                                                                               object
  16 AssemblyFileReference mscorlib = new AssemblyFileReference(ty
                                                                                               Roslyn.Compilers.CSharp
                                                                                  Options
                                                                                  Previous Submis (none)
   18 Compilation compilation = Compilation.Create("HelloWorld")
                                                                                  ReferenceResol Roslyn, Compilers, Metada
                                               .AddReferences (mscorlib)
                                               .AddSyntaxTrees(tree);
                                                                               Assembly
   21 return compilation;
```

#### 4.3 Obtaining Symbols

```
var sourceText=@"using System;
                namespace HelloWorld
                    class Program
                        static void Main(string[] args)
                            Console.WriteLine(""Hello, World!"");
SyntaxTree tree = SyntaxTree.ParseCompilationUnit(sourceText);
AssemblyFileReference mscorlib = new AssemblyFileReference(typeof
(object).Assembly.Location);
Compilation compilation = Compilation.Create("HelloWorld")
                                     .AddReferences (mscorlib)
                                     .AddSyntaxTrees(tree);
NamespaceSymbol globalNamespace = compilation.GlobalNamespace;
foreach (Symbol member in globalNamespace.GetMembers())
{
      member.str().info();
return globalNamespace;
//using Roslyn.Compilers
  //using Roslyn.Compilers.CSharp;
//O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
//O2Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
```

```
Inspector
                                                                                   Invoke and Result
 Command To Execute
                                                                                                   Execute
    14 SyntaxTree tree = SyntaxTree.ParseCompilationUnit(sourceText)
                                                                                  stop execution
    16 AssemblyFileReference mscorlib = new AssemblyFileReference(ty
                                                                                   Output
   18 Compilation compilation = Compilation.Create("HelloWorld")
                                                                                   .AddReferences (mscorlib)
   1.9
                                                                                      IsNamespace
                                                                                                    True
                                 .AddSyntaxTrees(tree);
                                                                                      IsOverride
                                                                                                    False
                                                                                      IsSealed
                                                                                                    False
   22 NamespaceSymbol globalNamespace = compilation.GlobalNamespace
                                                                                      IsStatic
                                                                                                    True
                                                                                      IsSynthesized
                                                                                                    False
   24 foreach (Symbol member in globalNamespace.GetMembers())
                                                                                      IsType
                                                                                                    False
   25 [
                                                                                      IsVirtual
                                                                                                    False
   26
           member.str().info();
                                                                                      Kind
                                                                                                    Namespace
   27 }
                                                                                                    C#
                                                                                      Language
                                                                                                    {Roslyn.Compilers.CShar
                                                                                      Locations
   29 return globalNamespace;
                                                                                      Name
    30 //using Roslyn.Compilers
                                                                                      OriginalDefinitic <global namespace>
    31 //using Roslyn.Compilers.CSharp;
    32 //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
    33 //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
Log Viewer
[3:23:41 AM] INFO: HelloWorld
[3:23:41 AM] INFO: Microsoft
[3:23:41 AM] INFO: AssemblyRef
[3:23:41 AM] INFO; This Assembly

    □ Debug

                               ✓ Info 64,504 kb | 66,352 kb
                                                                                                                  GC C
```

```
var sourceText=@"using System;
                namespace HelloWorld
                    class Program
                        static void Main(string[] args)
                            Console.WriteLine(""Hello, World!"");
SyntaxTree tree = SyntaxTree.ParseCompilationUnit(sourceText);
AssemblyFileReference mscorlib = new AssemblyFileReference(typeof
(object).Assembly.Location);
Compilation compilation = Compilation.Create("HelloWorld")
                                     .AddReferences (mscorlib)
                                     .AddSyntaxTrees(tree);
NamespaceSymbol globalNamespace = compilation.GlobalNamespace;
NamespaceSymbol systemNamespace = globalNamespace.GetMembers("System")
                                  .First() as NamespaceSymbol;
NamedTypeSymbol stringType = systemNamespace.GetMembers("String")
                             .First() as NamedTypeSymbol;
return stringType;
//using Roslyn.Compilers
  //using Roslyn.Compilers.CSharp;
//O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
```

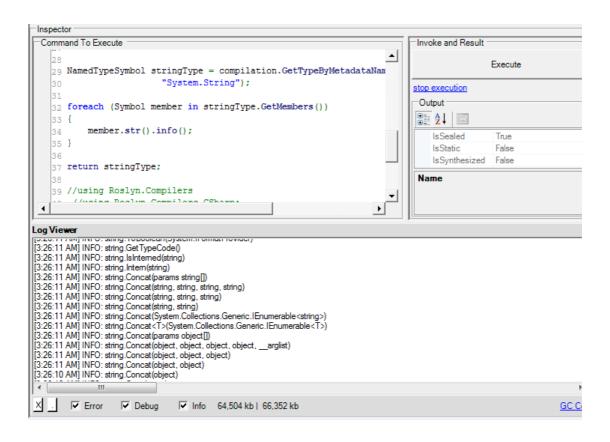
```
Inspector
                                                                     Invoke and Result
  Command To Execute
                                                                                   Execute
    16 AssemblyFileReference mscorlib = new AssemblyFileReference(ty
                                                                     stop execution
    18 Compilation compilation = Compilation.Create("HelloWorld")
                             .AddReferences (mscorlib)
                                                                      Output
    1.9
                            .AddSvntaxTrees(tree):
                                                                      IsSealed
                                                                                    True
    22 NamespaceSymbol globalNamespace = compilation.GlobalNamespace
                                                                         IsStatic
                                                                                    False
                                                                         IsSynthesized
                                                                                    False
                                                                         IsType
                                                                                    True
    25 NamespaceSymbol systemNamespace = globalNamespace.GetMembers (
                                                                         IsValueType
                                                                                    False
                            .First() as NamespaceSymbol;
                                                                         IsVirtual
                                                                                    False
                                                                         Kind
                                                                                    NamedType
    28 NamedTypeSymbol stringType = systemNamespace.GetMembers("Stri
                                                                                    C#
                                                                         Language
                       .First() as NamedTypeSymbol;
                                                                         Locations
                                                                                    {Roslyn,Compilers.CShar
    30 return stringType;
                                                                         MemberNames
                                                                         Name
    32 //using Roslyn.Compilers
                                                                                           S
                                                                         Original Definitio string
    33 //using Roslyn.Compilers.CSharp;
    34 //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
                                                                      Name
    35 //O2Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
 Log Viewer
[3:23:41 AM] INFO: Hello World
[3:23:41 AM] INFO: Microsoft
 [3:23:41 AM] INFO: AssemblyRef
[3:23:41 AM] INFO: This Assembly
       ▼ Error
                          ✓ Info 64,504 kb | 66,352 kb

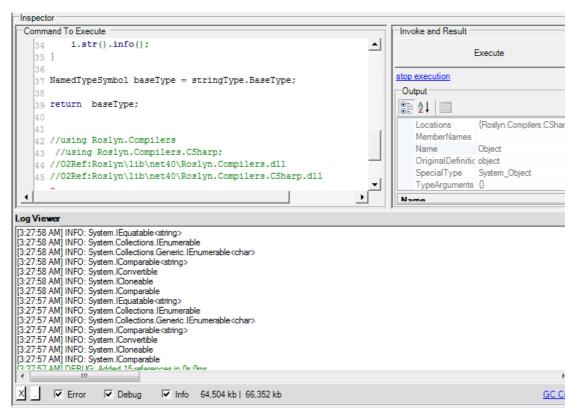
    □ Debug

                                                                                               GCC
var sourceText=@"using System;
                    namespace HelloWorld
                          class Program
                               static void Main(string[] args)
                                    Console.WriteLine(""Hello, World!"");
                    } ";
SyntaxTree tree = SyntaxTree.ParseCompilationUnit(sourceText);
AssemblyFileReference mscorlib = new AssemblyFileReference(typeof
(object).Assembly.Location);
Compilation compilation = Compilation.Create("HelloWorld")
                                               .AddReferences(mscorlib)
                                               .AddSyntaxTrees(tree);
NamespaceSymbol globalNamespace = compilation.GlobalNamespace;
NamespaceSymbol systemNamespace = globalNamespace.GetMembers("System")
                                           .First() as NamespaceSymbol;
NamedTypeSymbol stringType = compilation.GetTypeByMetadataName(
                                    "System.String");
foreach (Symbol member in stringType.GetMembers())
```

```
member.str().info();
}
return stringType;

//using Roslyn.Compilers
    //using Roslyn.Compilers.CSharp;
//02Ref:Roslyn\lib\net40\Roslyn.Compilers.dll
//02Ref:Roslyn\lib\net40\Roslyn.Compilers.CSharp.dll
```





#### 4.4 Asking Semantic Questions



•

•

.

.

.

.

•