

Advanced Reflection & Metaprogramming

Jb Evain <jbevain@novell.com>

Mono Team, Novell, Inc.

Metaprogramming

“Metaprogramming is the writing of computer programs that write or manipulate other programs (or themselves) as their data, or that do part of the work at compile time that would otherwise be done at runtime.”

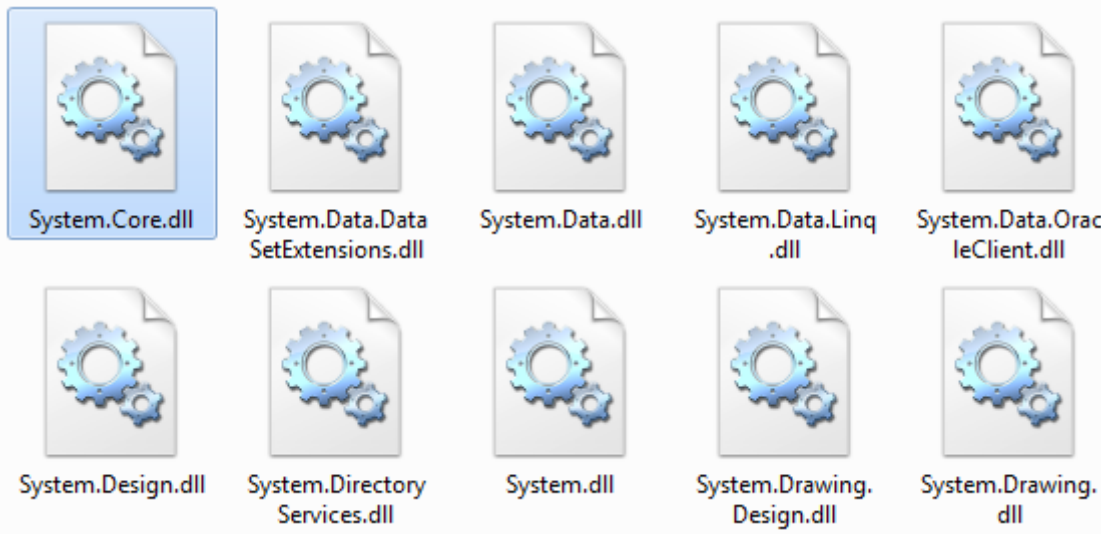
<http://en.wikipedia.com/wiki/>

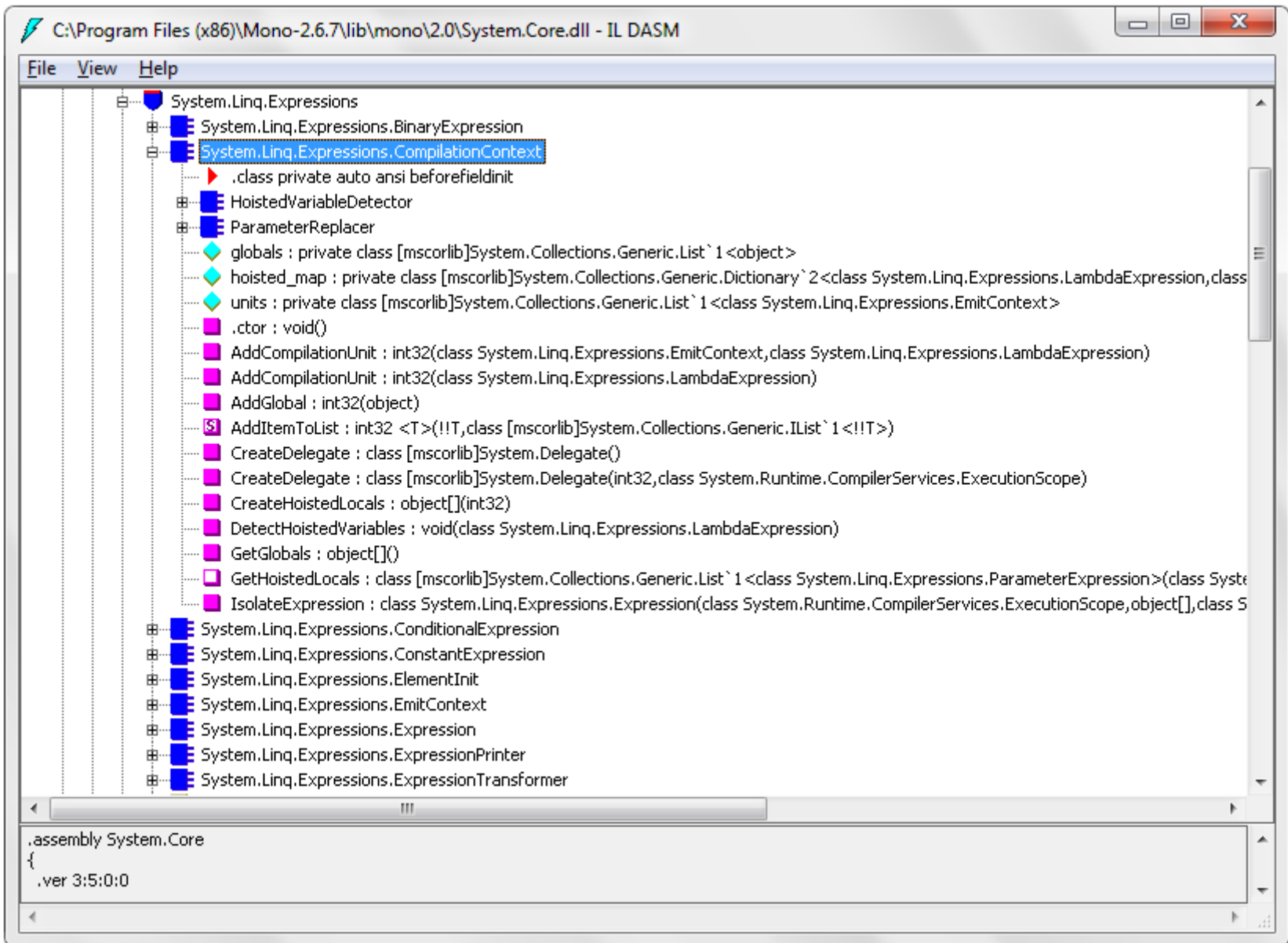
Metaprogramming

~~Languages / Compile time~~
~~metaprogramming~~

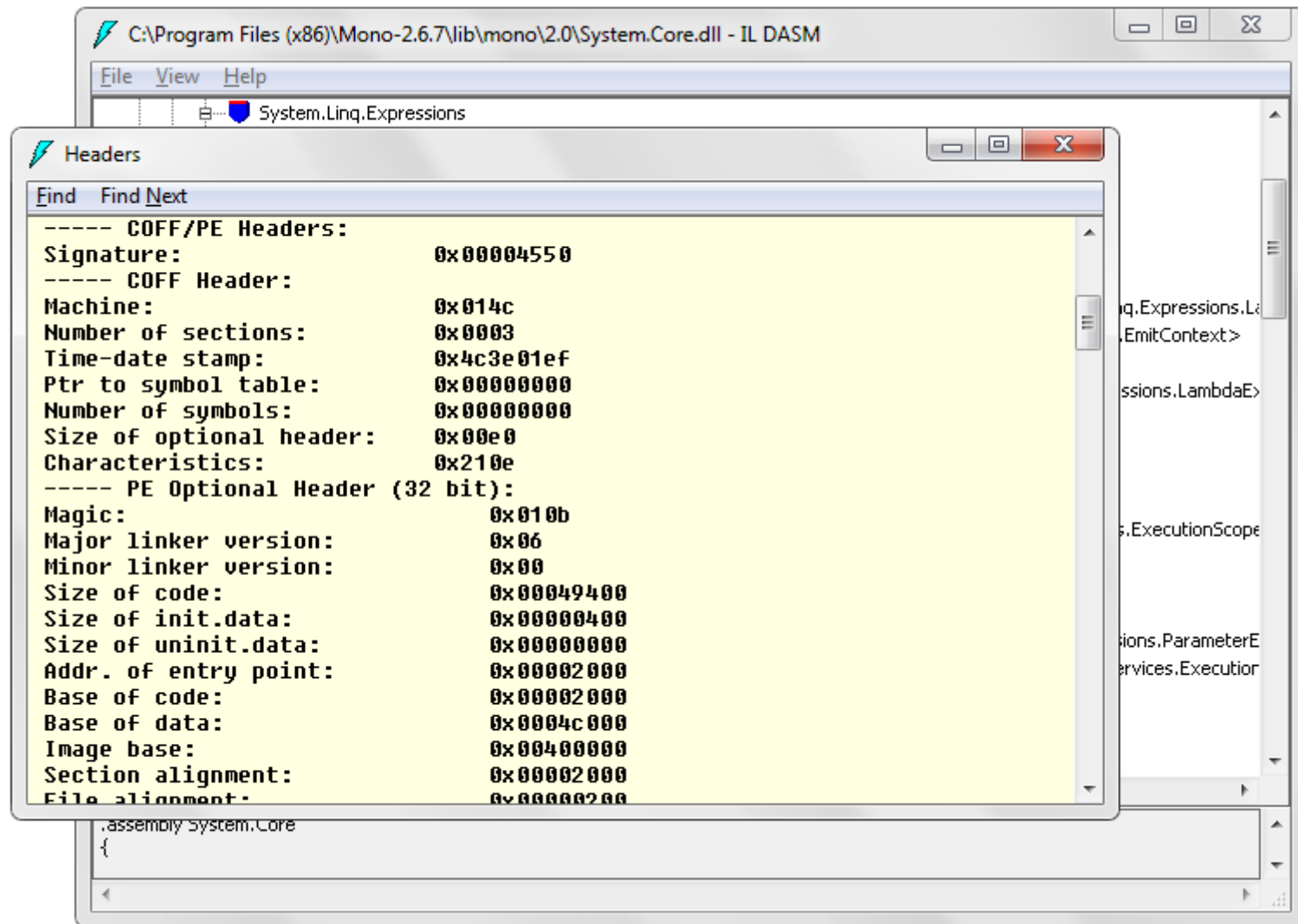
what this talk is not about

programs as data

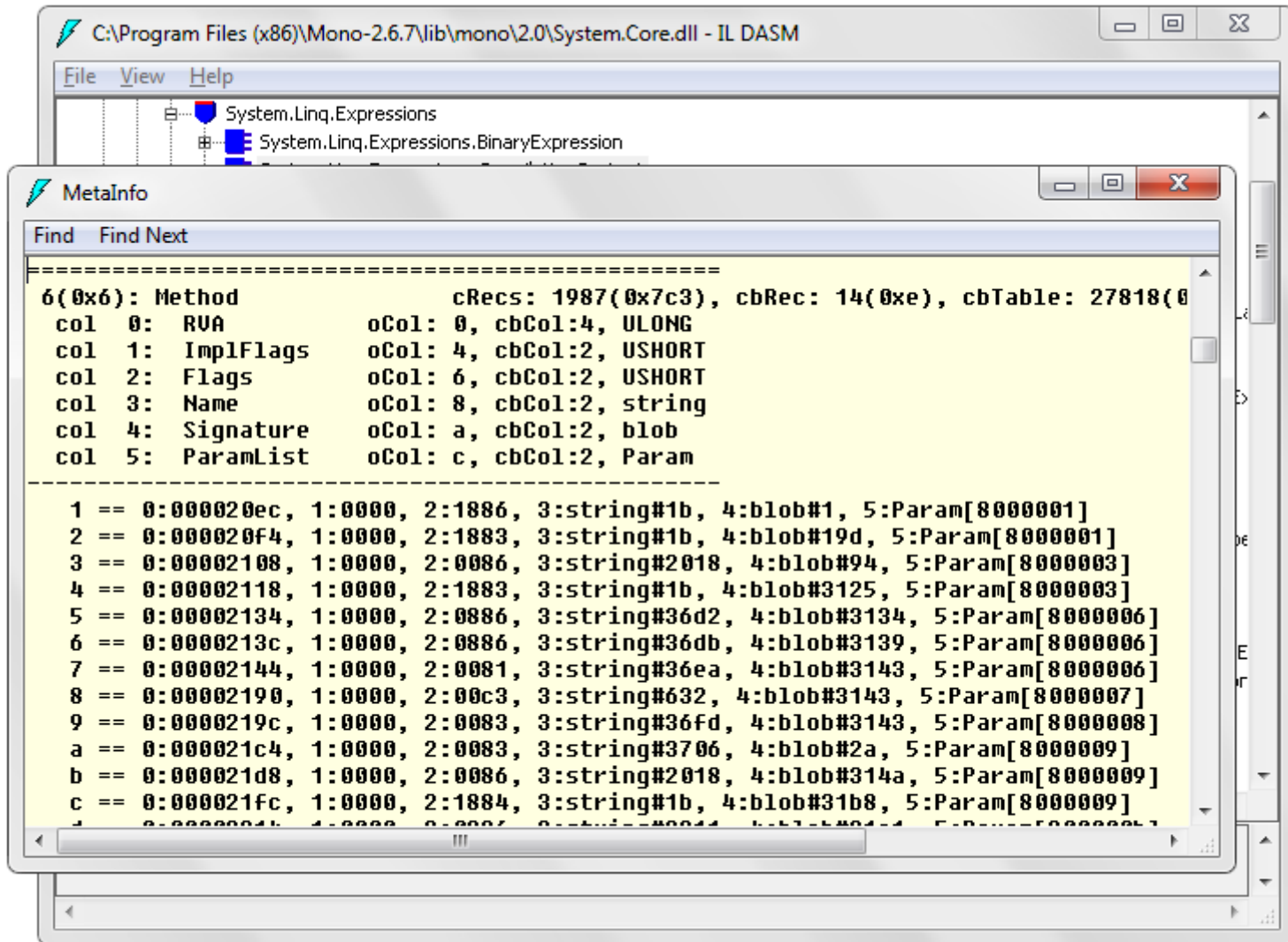




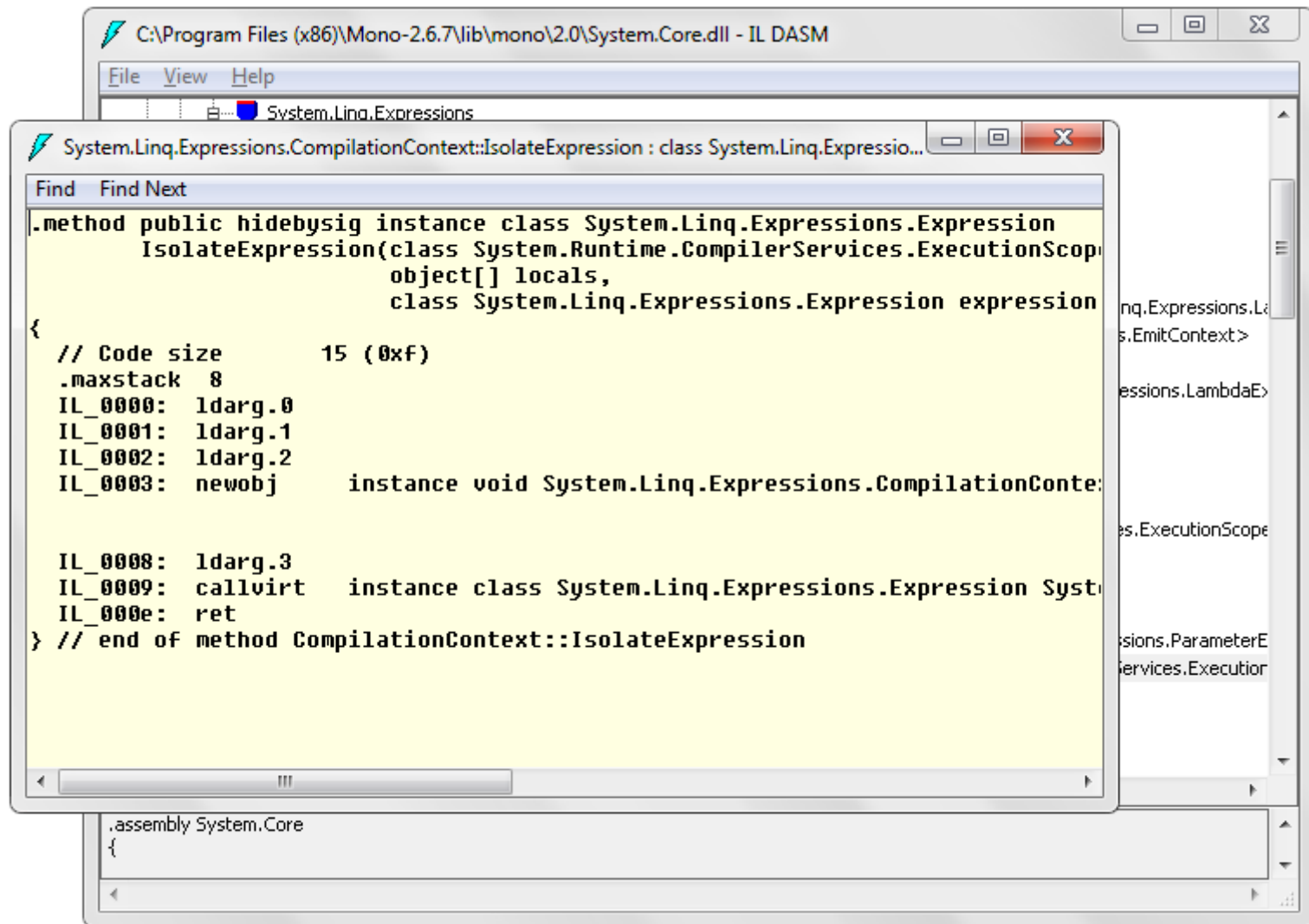
metadata



headers



tables



CIL

- System.Linq.Expressions
 - BinaryExpression
 - CompilationContext
 - Base Types
 - Derived Types
 - HoistedVariableDetector
 - ParameterReplacer
 - .ctor()
 - AddCompilationUnit(LambdaExpression) : Int32
 - AddCompilationUnit(EmitContext, LambdaExpression)
 - AddGlobal(Object) : Int32
 - AddItemToList<T>(T, IList<T>) : Int32
 - CreateDelegate() : Delegate
 - CreateDelegate(Int32, ExecutionScope) : Delegate
 - CreateHoistedLocals(Int32) : Object[]
 - DetectHoistedVariables(LambdaExpression) : Void

Disassembler

```
public Expression IsolateExpression(ExecutionScope scope, object[] locals, Expression expr)
{
    return new ParameterReplacer(this, scope, locals).Transform(expression);
}
```

```
public Expression IsolateExpression(ExecutionScope scope, object[] locals, Expression expr)
```

Declaring Type: System.Linq.Expressions.CompilationContext

Assembly: System.Core, Version=3.5.0.0



Step through this assembly in the Visual Studio debugger using .NET Reflector Pro

Install VS add-in



Standard ECMA-335

5th Edition / December 2010

**Common Language
Infrastructure (CLI)
Partitions I to VI**

System.Reflection

builtin

System.Reflection

- `Object.GetType ()`
- `Type.GetTypeFromHandle (...)`
- `MethodBody.GetILAsByteArray ()`

Mono.Reflection

lightweight reflection helper

Mono.Reflection

- Runtime IL disassembler
- IL pattern matching
- Property backing field

Mono.Cecil

Swiss Army Knife for assemblies

Cross Platform

- Mono
- .net >= 2.0
- CompactFramework >= 2
- Silverlight >= 2
- WP7

Why

- Complete metadata access
- No AppDomain bloating
- Collectible data
- No tie to a specific mscorlib.dll
- Ability to modify existing assemblies

Why not

- `System.Reflection.Emit's`
 - `AssemblyBuilderAccess.Run`
 - `AssemblyBuilderAccess.RunAndCollect`

The strong name problem

Similar projects

- Microsoft's CCI
- IKVM.Reflection[.Emit]

usage



@nikhilk
Nikhil Kothari

Said it before, will say it again: Mono Cecil simply rocks. It should have been in the framework. <http://bit.ly/cpKhWS> Thanks @jbevain

6 Sep via MetroTwit  Unfavorite  Retweet  Reply

achievement

a glimpse at the future

Milo

A level higher

links

- <http://twitter.com/jbevain>
- <http://evain.net/blog>

- <http://github.com/jbevain/cecil>
- <http://groups.google.com/group/mono-cecil>
- <http://github.com/jbevain/mono.reflection>

questions ?