

Outils de développement pour .NET Framework

Under Construction
En Construction

Didier Donsez

Université Joseph Fourier (Grenoble 1)

IMA IMAG/LSR/ADELE

Didier.Donsez@imag.fr

Didier.Donsez@ieee.org

Petit rappel sur .NET et C#

■ .NET

- Développement Multi langage
 - C#, C++, Java Script, Eiffel, Component Pascal, APL, Cobol, Oberon, Perl, Python, Scheme, Smalltalk, Standard ML, Haskell, Mercury, Oberon et Java/J++
- CIL (Common Intermediate Language)
- CTS (Common Type System)
- CLI (Common Language Infrastructure)
- CLR (CLI Runtime implémenté par MicroSoft)
 - JIT, pré-JIT (à l'installation, ou développement)

■ C# : le langage « Post-Java »

- « Synchronisation » de Java et de C++

■ Standardisation ECMA (European Computer Manufacturers Association)

- <http://www.ecma.ch>
- ECMA-334 CLI (Format COFF, CTS, Metadata, ...)
- ECMA-335 C#

Non Microsoft .NET

■ Motivations

- .NET sur des OS non Windows (Unix, Linux, MacOS X, ...)
- Implémentations libres
- Outils libres

■ Project

- ROTOR (Microsoft) sauf Linux (Shared Sources)
- Mono project (Ximian)
- DotGNU (Free Software Foundation)

Implémentations

■ Microsoft

- Commerciales
 - .NET CLR
 - Compact .NET CLR
- Code partagé
 - « Rotor » : Shared Source CLI
 - 3.6 Mloc (10,721 fichiers)
 - <http://msdn.microsoft.com/net/sscli>*
 - GC moins performant, JIT différent
 - Une sous partie du Runtime (pas de WebForm, ADO, ASP.NET, ...)

■ Mono

■ DotGNU Portable .NET

Compilateurs



■ Microsoft

■ Mono

- Compilateur et JIT efficace
- Peu de processeurs

■ DotGNU

- GCC Front End
- Linux et dizaine d'OS
- Nombreux processeurs

Outils du SDK

- Assembly Generation Utility (al.exe)
- Assembly Registration Utility (gac.exe)
- MSIL Assembler (ilasm.exe)
- MSIL Disassembler (ildasm.exe)
- C++ Compiler (cl.exe)
- C# Compiler (csc.exe)
- Visual Basic Compiler (vbc.exe)
- J# Compiler (en extra)
- PE File Format Viewer (dumpbin.exe)
- Type Library Exporter (tlbexp.exe)
- Type Library Importer (tlbimp.exe)
- XML Schema Definition Tool (xsd.exe)
- Shared Name Utility (sn.exe)
- Web Service Utility (wsdl.exe)

Décompilateurs

■ MSIL Disassembler (Ildasm.exe)

- a companion tool to the MSIL Assembler (Ilasm.exe).
Ildasm.exe takes a portable executable (PE) file that contains Microsoft intermediate language (MSIL) code and creates a text file suitable as input to Ilasm.exe.

Décompilateurs

The screenshot displays the ILDASM tool interface. The top window shows the decompiled code for the Main method, and the bottom window shows the decompiled code for the constructor.

```

CSharpTest.exe - ILDASM
Fichier  Affichier  Aide
CSharpTest.exe
MANIFEST
CSharpTestNamespace
CSharpTest
class private auto ansi beforefieldinit
currentDate : private valueType [mscorlib]System
ctor : void()
Main : void()

-EntryPoint
// Code size      49 (0x31)
-maxstack 2
.locals init (class CSharpTestNamespace.CSharpT
IL_0000: newobj
instance void CSharpTestN
IL_0005: stloc.0
IL_0006: ldstr
"Hello there."
IL_000b: call
void [mscorlib]System.Cons
IL_0010: ldstr
"Today is "
IL_0015: ldloc.0
valueType [mscorlib]Syste
IL_0016: ldflD
[mscorlib]System.Datetime
IL_001b: box
string [mscorlib]System.St
IL_0020: call
void [mscorlib]System.Cons
IL_0025: call
int32 [mscorlib]System.Cor
IL_002a: call
IL_002f: pop
IL_0030: ret
// end of method CSharpTest::Main

CSharpTest::ctor : void()
-method private hidebysig specialname rtspecialna
instance void .ctor() cil managed
{
// Code size      18 (0x12)
-maxstack 2
IL_0000: ldarg.0
instance void [mscorlib]Sy
IL_0001: call
IL_0006: ldarg.0
valueType [mscorlib]Syste
IL_0007: call
valueType [mscorlib]Syste
IL_000c: stfld
valueType [mscorlib]Syste
IL_0011: ret
// end of method CSharpTest::ctor

```

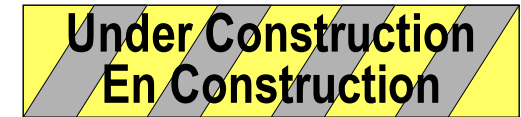

Ofuscateur



■ Dotfuscator

- (<http://www.preemptive.com/dotfuscator>)

Profilage



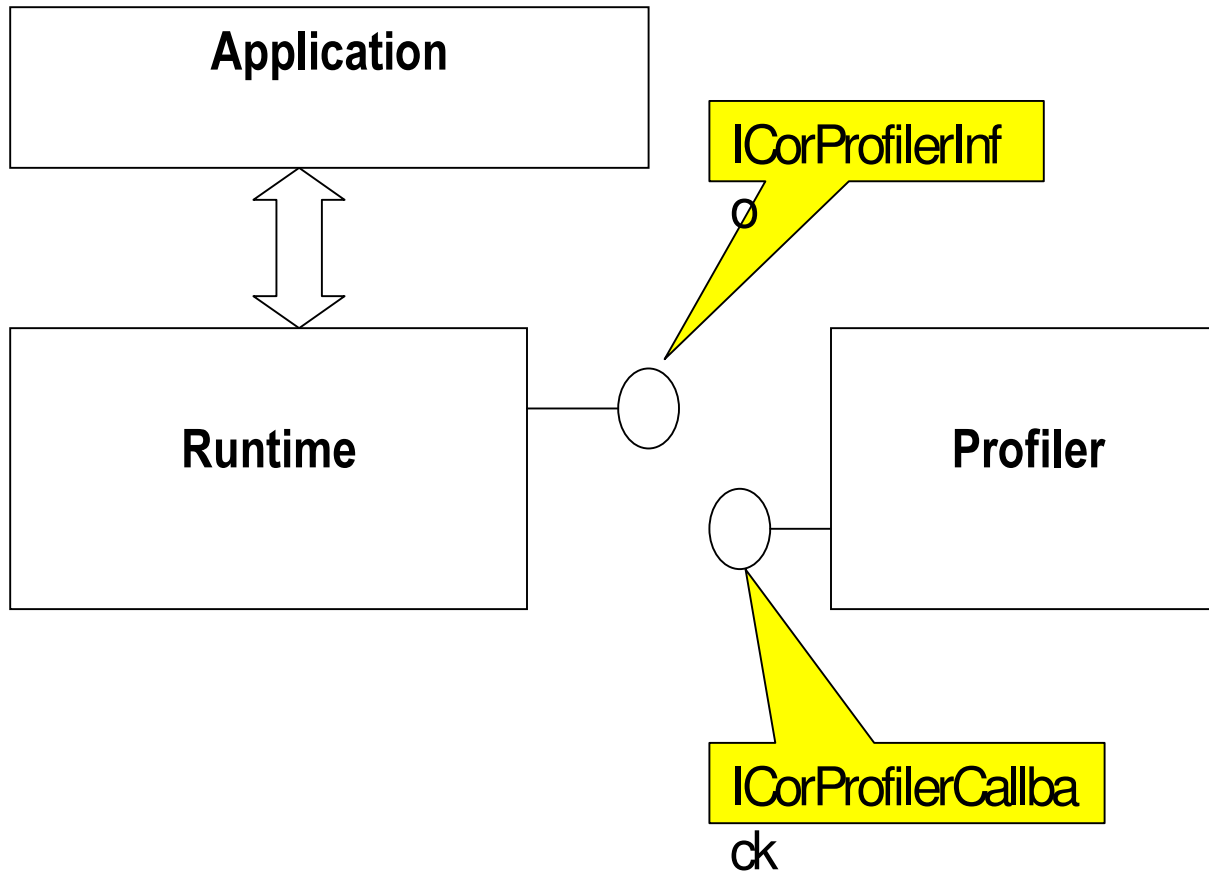
■ Identification

- des points chauds (*bottleneck, hot spot*)
- des fuites mémoires ...

■ Outils

Profilage

■ Il existe un namespace



Génération (ou émission) dynamique de code

■ Motivations

- AOP, Dynamic Proxy (Réseau, BD, ...), ...

■ namespace System.Reflection.Emit

- Permet la construction d'assemblies, modules, classes à la volée

■ Voir

- <http://www.dina.dk/~sestoft/rtcg/rtcg.pdf>
- Rechercher Cisternino & Kennedy, Language independent program generation

Exemple de Génération de code (i)

```
private static Type CreateHelloWorld(AppDomain appDomain, AssemblyBuilderAccess access) {  
  
    AssemblyName assemblyName = new AssemblyName();  
    assemblyName.Name = "EmittedAssembly";  
  
    // Create the dynamic assembly.  
    AssemblyBuilder assembly = appDomain.DefineDynamicAssembly(assemblyName, access);  
  
    // Create a dynamic module  
    ModuleBuilder module = assembly.DefineDynamicModule("EmittedModule");  
  
    // Define a public class named "HelloWorld" in the assembly.  
    TypeBuilder helloWorldClass = module.DefineType("HelloWorld", TypeAttributes.Public);  
  
    // Define a private String field named "Greeting" in the type.  
    FieldBuilder greetingField = helloWorldClass.DefineField(  
        "Greeting", typeof(String), FieldAttributes.Private);  
}
```

Exemple de Génération de code (ii)

```
// Create the constructor.
Type[] constructorArgs = { typeof(String) };
ConstructorBuilder constructor = helloWorldClass.DefineConstructor(
    MethodAttributes.Public, CallingConventions.Standard, constructorArgs);

// Generate IL for the method. The constructor calls its superclass constructor.
// The constructor stores its argument in the private field.
ILGenerator constructorIL = constructor.GetILGenerator();
constructorIL.Emit(OpCodes.Ldarg_0);
ConstructorInfo superConstructor = typeof(Object).GetConstructor(new Type[0]);
constructorIL.Emit(OpCodes.Call, superConstructor);
constructorIL.Emit(OpCodes.Ldarg_0);
constructorIL.Emit(OpCodes.Ldarg_1);
constructorIL.Emit(OpCodes.Stfld, greetingField);
constructorIL.Emit(OpCodes.Ret);
```

Exemple de Génération de code (iii)

```
// Create the GetGreeting method.
```

```
MethodBuilder getGreetingMethod = helloWorldClass.DefineMethod("GetGreeting",  
    MethodAttributes.Public, typeof(String), null);
```

```
// Generate IL for GetGreeting.
```

```
ILGenerator methodIL = getGreetingMethod.GetILGenerator();  
methodIL.Emit(OpCodes.Ldarg_0);  
methodIL.Emit(OpCodes.Ldfld, greetingField);  
methodIL.Emit(OpCodes.Ret);
```

```
// Bake the class HelloWorld.
```

```
return(helloWorldClass.CreateType());  
}
```

Execution

- Assembly Loader
- Security
- Class Loader
- IL to Native Compiler
 - Native.exe+GC Table

Test

■ Unitaire

- CLR
 - NUnit

- ASP.NET
 - NUnitWeb

<http://kristopherjohnson.net/cgi-bin/twiki/view/KJ/NUnitWeb>

- Web Services

Normes de programmation

■ C#

- ???
- Style Checkers ??

■ VB

Sécurité



- Gestion de certificats
- Signature de code
- Code sellé ?? *sealed* non réversible
- Vérifieur de code
- Pré-vérifieur de code
 - Vérifie en outre la présence de unmanaged section

Gestion de projets

■ Makefile

- nmake (.NET SDK)

■ Tâches (optionnelles) ANT

- Le .NET SDK doit être installé; variables d'environnement configurées
- <csc> compilateur C#
- <ilasm> assembleur d'IL
- <WsdItoDotnet> générateur de stub/skel SOAP en C#, VB, ..
- Exemple
 - ```
<csc optimize="true" debug="false" docFile="documentation.xml" warnLevel="4"
unsafe="false" targetType="exe" incremental="false" definitions="RELEASE"
excludes="src/unicode_class.cs"
mainClass = "MainApp" destFile="NetApp.exe" />
```

## ■ NAnt (<http://nant.sourceforge.net/>)

- Un portage de ANT sur .NET

# Convertisseurs Java vers C# / .NET

- Aztec J2CS, ArtinSoft JLCA, DotNetJ, ...
- Microsoft JCLA (Java Language Conversion Assistant)
  - Assistant pour VisualStudio .NET de Conversion
    - Java ou J++ vers J# ou C#
      - 90 % de appels au JDK 1.1.4 convertis

| Java technology  | Upgrade to J#    | Upgrade to C#        |
|------------------|------------------|----------------------|
| Java language    | Java language    | C# language          |
| Applet           | Not converted    | Windows Form control |
| JavaBean         | JavaBean         | C# class             |
| AWT frame        | AWT frame        | Windows Form         |
| WFC Form         | WFC Form         | Windows Form         |
| Compiled library | Compiled library | Not converted        |
| Resource file    | ResX file        | ResX file            |

# Migration J2EE vers .NET

## ■ J2EE

- Servlets, JSP
- EJB
  - Session Bean
  - Entity Bean
    - CMP
    - BMP
- JDBC
- Procédures Stockés
  - Oracle PL/SQL, ...

## ■ .NET

- ASP .NET
- ADO .NET
  - Transact SQL
- Web Services

# Ateliers (CASE)

- Microsoft VisualStudio .NET
  
- Microsoft WebMatrix
  - Gratuit (intègre Cassini (Web Server))
  - Pas de complétion au codage !
  
- Eclipse + plugin
  
- Autres (Inprise ??)

# Mono Project (Ximian)

<http://www.go-mono.com>

## ■ Motivation

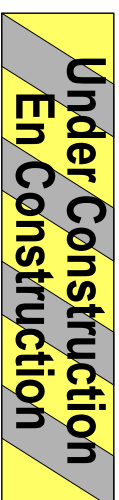
- Effort pour créer une version libre de .NET Framework.
- Pour Linux (donc MacOSX)

## ■ Outils

- Compilateur C#
- Runtime
  - JIT (Linux/x86)
  - Interpreteur (Linux/x86, Linux/PPC, S390 + en cours pour StrongARM, SPARC)
- Bibliothèques de classes.
- Implémentations de ADO.NET et ASP.NET



# DotGNU



# Serveurs HTTP

## ■ Motivations

- Support aux ASP .NET

## ■ Serveurs

- MicroSoft IIS
- MS Cassini (Gratuit)

# CORBA et .NET

- Il y a assez d'info dans la CLI pour créer des stubs et de squelettes (Miguel de Icaza)

# J2EE et .NET

## ■ DotNetJ

- Permet à des clients .NET d'accéder à une application J2EE/EJB via des .NET Remoting
  - 2 options de Canal .NET Remoting
    - Canal IIOP
    - Canal personnalisé (ObjectWeb/CAROL)
  - Testé avec JOnAS (en partenariat avec ObjectWeb)
- Remarque :
  - Les WebServices ne permettent pas le passage d'objets par référence

# Web Services et .NET

■ SOAP

■ WSDL

■ UDDI

■ XSD

■ Interopérabilité

- Apache AXIS, WebLogic, WebSphere ...

# Outils SDK pour les WS

## ■ wsdl.exe

- Génère un stub SOAP à partir d'un WSDL
  - `wSDL /language:vb http://localhost/pmcalc/pmcalc.asmx?wsdl`
  - `wSDL /language:cs http://localhost/pmcalc/pmcalc.asmx?wsdl`

## ■ soapsuds.exe

- Génère un WSDL à partir des Metadata

# Bases de Données et .NET

## ■ API DataSet

## ■ SGBD

- SQL Server
- MSDE ( MS SQL Server 2000 Desktop Engine )
  - Gratuit pour les tests

# .NET et Temps Réel

- Voir article Lutz, M.H.; Laplante, P.A, '*C# and the .NET framework: ready for real time?*', IEEE Software Volume: 20, Issue: 1, Jan/Feb 2003, pp 74- 80



# Embarqué et Nomade

## ■ .NET compact Framework

- .NET pour cibles Windows CE

## ■ .NET SmartCard (<http://www.hiveminded.com/>)

- Implémentation de la CLI pour les cartes à puce

## ■ .NET Framework for X-Box

- cible les consoles de jeu
- Disponibilité ???

# .NET Compact Framework



## ■ .NET Compact Framework

- Version allégé de .NET Framework
- Cible les profils CDC et CDLC
- fonctionnement offline
  - Exemple: cache de ligne SQL (ADO)
- RTE
  - Core CLI, réseau, XML, Web Services, ADO .NET
  - Garbage collector : simple Mark and Sweep
  - JIT MSIL → Natif
    - au premier appel
    - Cache de code JIT
- Environnement matériel
  - StrongARM, MIPS, x86, SH4, Xscale, ...
  - Windows CE
  - RAM : 1.5 Mo minimum

# .NET SmartCard

<http://www.hiveminded.com/>

- Implémentation de la CLI adapté à la carte
- Multi-applications
- Développement Multi-langage : C#, J#, VB, Jscript, Perl, ...
- Caractéristiques
  - Isolation
    - Application Domain de .NET
  - Transactions
    - Multi-niveaux ?
  - Garbage Collector
    - Mark and Sweep (sans marquage en EEPROM)
  - Communication
    - Inter-applications
      - Channel : flux d'octets bidirectionnel
    - Terminal-Application
      - APDU, .NET Remoting, Javacard 2.2 RMI

# OlyMars

## ■ ~Entity Bean CMP

# Extra

## ■ API par COM+

- Utilise des wrappers
- Transaction (coordinateur MTS), Sécurité, ...

# Bibliographie

- Beaucoup d'ouvrages sur C# et .NET
  - <http://www.oreilly.com/dotnet>
  - ...
- David Stutz, Ted Neward, Geoff Shilling, Shared Source CLI Essentials, O'Reilly, March 2003 (est.), ISBN 0-596-00351-x

# Sites

- <http://microsoft.com/net>
- <http://www.dotnetguru.org>
- <http://www.go-mono.com>
- <http://dotgnu.org>
- <http://www.ecma.org>
  
- <http://www.codeproject.com/dotnet>
  - Fournit pas mal d'exemples

MCours.com