

[1] Technical Overview of .NET
FRUUG
14 January, 2001
Tom Cargill
.Profile Consulting
www.profcon.com/

[2] N Years Ago This Month
6: Here, talking about Java.
19: First C++ program.
32: First programming job.

[3] Goal
I got curious about .NET.
Original C# manual [Ho hum.]
PLDI papers from Microsoft Research [Aha!]
I'm here to share what I've learned.

[4] What is .NET?
At the top:
Hype and smoke, mostly.
My Services (a.k.a. Hailstorm)
In the middle:
Programming languages/tools/frameworks
XML
Standard Library
At the bottom:
A virtual machine, like Java's.
CLI - Common Language Infrastructure

[5] .NET Value Proposition
Program at a higher, more robust level of abstraction.
Runtime checks, garbage collection, etc., ('managed').
Trade machine cycles for brain cycles.
x2-3 productivity gain with C# vs C++ in ASP.NET
Scott Guthrie, Conference.NET 8/15/01
(Sound like Java?)

[6] Portability
Java's other value proposition.
Much better than any prior technology.
.NET *could* do the same.

[7] Litigation/Standards

Microsoft licenses and extends Java; Sun sues; etc.

Sun backs out of ISO and ECMA.

Microsoft creates its own.

Microsoft gives C# and CLI to ECMA.

www.ecma.ch/ecma1/MEMENTO/tc39.htm

Then it gets grey

Where is the line between the standards and MS?

What patents does MS hold?

[8] (Some) History

COM3/Lightning -> CLR (Common Language Runtime)

1997

COOL -> C#

c2.com/cgi/wiki?ProjectCool

XSP -> ASP.NET

1998

Microsoft's spin is that .NET is a successor to COM.

I can find no COM in .NET

I see it as a successor to Java.

[9] The .NET Machine

An object virtual machine.

Much like the Java Virtual Machine.

Many instructions are the same.

Less programming language specific.

It supports somewhere between 2 and 20 languages.

Designed primarily for native execution, with interpretation as an option.

Opposite of the JVM

Includes:

Unsafe mode, code versioning and authentication

[10] Virtual versus Abstract Machine

Purists reserve the term *virtual* machine for an engine running on itself.

Like classical IBM VM, or VMWare.

So, .NET is an *abstract* machine.

Java destroyed this nomenclatural distinction.

[11] .NET versus Java

Gosling's characterization:

"Imitation is the sincerest form of flattery."

My characterization:

Microsoft thinks: Java was a nice research project; now, here's the product.

[12] .NET Languages

Primary

C#, VB

Secondary

(Live|Java|ECMA|J)Script, C++, Java?

Tertiary

APL, Cobol, Component Pascal, Eiffel, Fortran, Fujitsu Cobol, Haskell, Mercury, Oberon, Oz, Perl, Python, RPG, Scheme, Smalltalk, Standard ML, ... ???

[13] Common Language Infrastructure (CLI)

Type System (CTS)

Instruction Set (CIL)

Metadata, i.e. symbol tables

CLR - Common Language Runtime

Microsoft's commercial *implementation* of CLI spec.

Microsoft also has a 'shared source' implementation, called CLI.

[14] Common Type System (CTS)

Class, single inheritance

Interface

Structure: value type

Enumeration

Property: getter/setter method pair

Attribute: ~aspect

Delegate: ~method pointer

[15] Common Language Specification (CLS)

A *subset* of the CLI/CTS that enables cross-language binding.

E.g. no unsigned types

A CLS-compliant compiler's output interoperates with that of others.

A method in one language can call another.

A class in one language can *inherit* from another.

[16] Detail

How do you reference an identifier in another language if it's a *keyword* in yours?

In C#, escape as @id, like

```
x.@switch("on");
```

[17] ECMA Standards

European Computer Manufacturers Association
Technical Committee 39

Five key documents

- CLI Part I Architecture
- CLI Part II Metadata (symbol tables)
- CLI Part III CIL (instruction set)
- CLI Part IV Library
- C# Specification

[18] Implementations

Microsoft
Associates
Open Source

[19] Microsoft

Beta
VM
Library
C#, VB, C++ compilers
J#
ASP.NET
Compilers from associates
Eiffel, ...

[20] Open Source Implementations

Portable.net
gcc based C# compiler (C# -> CIL)
VM in C
Libraries
www.southern-storm.com.au/portable_net.html
Mono
C# compiler in C#
VM
Libraries
www.go-mono.com/
DotGNU
"DotGNU will be a complete replacement for the .NET strategy."
www.dotgnu.org/html2.0/index.html

[21] C# Programming Language

The language that most closely fits the VM.
An overview of C# gives a good intuitive feel for the VM.

[22] C# Hello World

Look carefully:

```
public
class Hello {
static void Main(string[] argv) {
System.Console.Out.WriteLine("Hello");
}
}
```

[23] Compile and Run

[Win2K/Cygwin]

Compile

```
$ echo Hello*
Hello.cs
$ csc Hello.cs
$ echo Hello.*
Hello.cs Hello.exe
```

Run

```
$ Hello.exe
Hello
$
```

[24] Disassemble

The .exe contains CIL bytecodes!

```
$ ildasm Hello.exe
...
.class public auto ansi beforefieldinit Hello extends
[mscorlib]System.Object
{
.method private hidebysig static void Main(string[] argv)
cil managed
{
.entrypoint
.maxstack 2
IL_0000: call class [mscorlib]System.IO.TextWriter
[mmscorlib]System.Console::get_Out()
IL_0005: ldstr "Hello"
IL_000a: callvirt instance void
[mmscorlib]System.IO.TextWriter::WriteLine(string)
IL_000f: ret
}
```

[25] C# is Java++--

Remove

(Non-static) Inner class

Checked exception

Add

Delegate: (composite) method pointer

Property: field syntax with method semantics

Event: notification

Attribute: ~aspect

Boxing: implicit wrapping

enum

struct: aggregate *value*

Operator overloading

Unsigned types

unsafe: C-style pointers

Minutiae

foreach

goto instead of break/continue label

decimal

[26] Deployment

.NET code is deployed in an *assembly*.

Like an exploded archive that retains its table-of-contents (*manifest*).

An assembly is a *self-describing* set of files.

A file can be a simple data resource or a *module*.

A module contains metadata and bytecode.

.dll or .exe (in .NET format)

One module contains the *manifest*.

[27] Application Deployment (Theory)

1. `cp -r` a tree of assemblies.

2. Invoke a .exe within the tree.

3. *Private* assemblies are in directories below the .exe.

Logically akin to static linking.

4. *Shared* assemblies are in a common *cache*.

The Registry is not involved. :))

[28] Global Assembly Cache : (

The GAC

Closest Java analogy: \$JAVA_HOME/jre/lib/ext

A *per computer* collection of assemblies.

Open question: Can more than one version of .NET be installed?

(VMWare?)

A directory that *isn't* a directory that *is* a directory.

[29] Code Versioning (Theory)

A shared assembly carries a 4x32-bit version number.

`major.minor.revision.build`, like `4.0.1381.6`

At build-time, each assembly records the versions of others on which it depends.

At dynamic load-time, a resolution policy (in XML) binds consistently.

Exact match, rebinding, etc.

The GAC may contain more than one version of a given assembly.

Applications that require different versions of common libraries may cohabit.

[30] Code Versioning (Practice?)

An assembly corresponds to a Java classloader, sort of.

A type can appear in more than one assembly.

Absolute type identity is a type-assembly *pair*.

A type can be loaded more than once.

If it's in more than one assembly.

If more than one version of an assembly is loaded.

It appears to be at least as complicated as Java.

If you think that Java isn't complicated, see Bill Connor.

[31] Code Authentication

RSA 1024 bit keys

SHA digests

'Strong' assembly name has public key.

Guarantee that run-time codebase matches that of compile-time.

[32] Application Domain

Machine is partitioned into domains.

Domain is unit of resource management.

Like a Java class loader, sort of.

Domain is unit of security.

[33] What Sits on Top

ASP.NET

A web server engine/framework implemented in .NET.

My Services (Hailstorm)

Network services that manipulate users' data stored on servers.

Kerberos/Passport

SOAP

No *technical* connection to the virtual machine.

[34] Conclusion
.NET ~= Java++