

# Linux.NET

**A**s many people know, Microsoft came onto the Internet bandwagon quite late into the game. However, once they restructured their business plans to take note of the WWW, they have made it their business to have a finger in every pie that was available. It's no accident that Hotmail is the world's most popular webmail system and MSN is the world's most popular website! The latest initiative from Redmond has been christened .NET, and is much more than just a marketing ploy.

## What is .NET?

.NET isn't just the name Microsoft are giving to many of their products, so although *Visual Studio.NET* is out, and *Windows Server.NET* almost made it to release (it's now been renamed to *Windows Server 2003*, "to avoid confusion"), .NET is a technology, not a product in itself.

.NET is a set of technologies designed to take advantage of the Internet. As such, it works over the Web using Web Services, it is heterogeneous in that programs written for .NET compile down to an intermediate language not native machine code, and also the .NET interpreter system is designed to catch common flaws exploited over the Internet, such as buffer overruns.

Perhaps the most important part of .NET is its heterogeneous nature – it can convert a variety of source languages (such as C++, Java, and Microsoft's new language, C#) into intermediate code (known as CIL, Common Intermediate Language), which is then hot-spot compiled into native code on the execution platform. As a result, .NET is capable of running the same executables on a variety of platforms – wherever the interpreter is available.

If you're a Java convert, this will all sound like old news to you – after all, Java does much the same. Indeed, reading through C# code and the same program in Java will show quite how similar the two systems can be. .NET, though, was *designed* to accept multiple languages (unlike Java, which wasn't, but can anyway) and so makes its class libraries available to all languages. What this means is that a programmer who uses language X because it has libraries Y no longer need worry, as each language is cross-compatible. In fact, this cross-language compatibility extends to inheritance: you can design a class in C# then inherit from that class using Eiffel#, and you can even debug across the two languages simultaneously.

.NET includes a large collection of class libraries that allow easy access to XML, databases, GUIs, and such. A complete .NET implementation includes replicated class libraries, a compiler to CIL, and of course an CIL runtime interpreter. Collectively, these are known as the .NET Framework. Generally, C# is chosen as the basic language to compile, because it was designed to implement all parts

**Microsoft is pushing its .NET principle hard – but how does it affect Linux users? PAUL HUDSON investigates...**

of .NET – the language can be thought of as a means rather than as an end.

## Linux and .NET

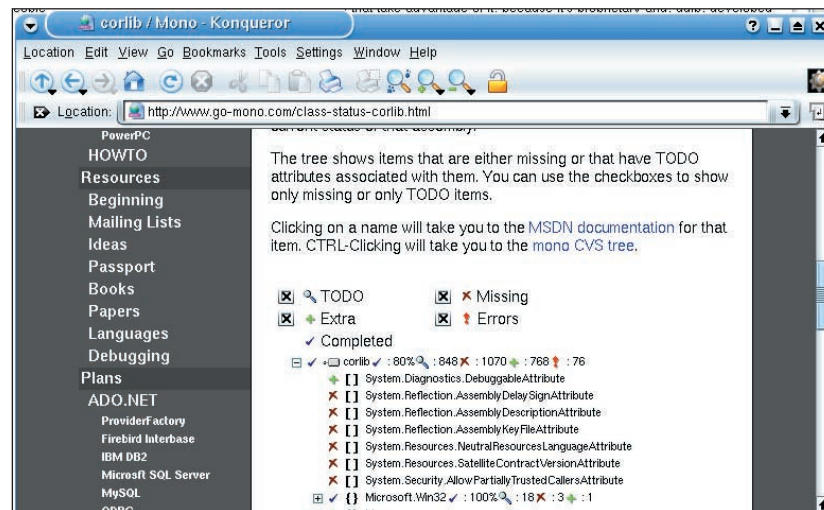
So, .NET is a technology that might be useful for Linux users. After all, if applications written for Windows are cleanly compiled down to CIL, they can be run on Linux natively. Two groups of developers recognise this possibility, so there are two free software projects currently being developed that provide .NET compatibility: DotGNU and Mono. The Mono project is being led by Miguel de Icaza, of GNOME fame, and co-founder of Ximian, which has lent some extra weight to Mono development. It's important to note that Mono and DotGNU aren't competing with each other – officially, DotGNU is designed to enable decentralised services and authentication, whereas Mono is there to provide the compiler, class libraries, and runtime.

Once Mono is ready for primetime, most Windows .NET applications should run fine on Linux. Other than the obvious result of "great, lots more programs for Linux are available, so people might be more likely to switch", there's also the side-effect that it would make programming for Windows and Linux exactly the same – a programmer trained to write C# on Windows could write exactly the same code on a Linux box and have it work exactly the same way. So, adding quality .NET support to Linux increases not only the number of applications that are available, but also the number of people capable of writing new applications.

## Why use .NET?

Using a .NET-compatible language today allows you to take advantage of its cross-platform nature – as long as you don't use vendor-specific class libraries in your code, it should be portable to Windows, Solaris, etc, while still retaining all the

**Want to know what Mono supports as of now? The Mono site tells you which of the class libraries are supported and what's missing in a very straightforward manner.**



functionality – including the GUI. Also, you can benefit from .NET's cross-language nature, allowing you to distribute work over a group of people, each of whom are able to use the language they feel most comfortable with – at the end of the day, the compiled code will work together seamlessly.

One advantage to using .NET is the option to use *managed code*. This is where you place a lot of your programming responsibility into the hands of the .NET interpreter. The interpreter will perform automatic garbage collection, and also detect attempted buffer overflows, and halt them in their tracks. The obvious disadvantage to using managed code is that it takes control out of the hands of programmers – I for one like to have complete control over my code, and consider it a little sloppy to rely on garbage collection to clean up after yourself.

For those programmers willing to take on the .NET challenge, there's certainly a great deal of support – not the least of which is the set of class libraries that the .NET Framework includes. These class libraries allow easy access to particularly complicated and regularly used aspects of web programming. The most useful libraries are Windows Forms and XML, which give .NET-compatible languages much of the same functionality as their Java versions.

One particularly important thing I wish to make clear is that, in my opinion, you shouldn't feel 'dirty' using .NET. All too often I hear people saying that we shouldn't use .NET or technologies that take advantage of it, because it's proprietary and, gulp, developed by MS. It is my fear that this is representative of an increasing feeling of "not invented here" among Linux developers (particularly those who spell the Redmond company 'Micro\$oft') – that is, if software is developed by a big company, it should be shunned. Perhaps these people forget the roots of GNU and Linux, which were a group of people wanting to take existing proprietary UNIX and re-implement it as free software – precisely what DotGNU and Mono are doing for .NET. .NET is a technology that represents a big leap forward for programming languages: try to judge it on its individual merits without worrying about who invented it.

## Be careful using .NET

It's important to remember that though its functionality is great, there are potential problems with .NET programming. Firstly, like Java, .NET is designed to be cross-platform, however, unlike Java, it is very easy to tie .NET programs to a specific platform – especially Windows. The cross-platform problem has several parts: OS differences, Windows-specific features, and native calls.

Firstly, OS differences comes down to specific things such as reading files. For example, whereas `c:\program files\foobar` might exist on Windows, it will not exist on Linux. Similarly, you may need to watch case sensitivity now and then, as this is often an issue, particularly with files. Windows-specific features include the Windows Forms GUI toolkit, the Windows registry, etc. Mono has compatibility for Windows Forms built-in, however it's only currently working solidly on Windows – Linux/OS X support is shaky. The problem here is that Windows Forms weren't designed for cross-platform compatibility, but instead they were designed to allow developers to switch to .NET whilst still retaining as

## YOU CAN QUOTE ME ON THAT...

### What people have to say about Mono and DotGNU

**"With Mono and DotGNU, we hope to provide good alternatives to components of .NET, ones that will respect your freedom, and your privacy. You will be able to use the facilities of Mono and DotGNU either with, or without, the Internet, and using servers of your choice."**  
**RICHARD STALLMAN**, GNU project founder

**"I want to be able to deliver four times as many free software applications with the**

**same resources, and I believe that this is achievable with these new technologies."**  
**MIGUEL DE ICAZA**, founder of GNOME and Mono lead developer

**".NET is truly multi-language and the role of the framework is to provide a reasonable target to which all current languages can map"**

**BERTRAND MEYERM**, creator of the Eiffel language



**Installing Mono on your machine is usually painless. Debian users should go to [www.debianplanet.org/mono/](http://www.debianplanet.org/mono/) and follow the instructions.**

much of traditional Windows GUI programming as possible.

The final problem is native calls using Platform Invocation Services, or *pinvoke*. This allows a .NET-managed program to call a unmanaged program, such as a Windows DLL. Naturally this means leaving the cross-platform safety of .NET, so *pinvoke* isn't guaranteed to work, even if the shared library is available. This works in the same way as Java's Native Interface (JNI), and can be a great benefit as well as a great pain in the backside!

## Conclusion

If MS has its way, .NET is certainly here to stay. Even if MS decides not to support it in the future, it's still a great technology and one that Linux users should be happy to take hold of and use. People regularly cite "Ah, but MS might introduce patents!" and the like as reasoning not to use .NET, but really – you have nothing to fear. Even if MS do introduce patents into .NET v2 (or whatever they call it), it won't stop Mono developers working with what they have already, and it certainly won't stop Mono developers creating their own implementation of proprietary code – look how well the *Samba* people have done.

Give .NET a try, and attempt to forget that it's a MS idea. We think you'll be pleasantly surprised! ■