

# Steve Scott's Column

At our London meeting in March I presented a session introducing Microsoft's new .NET framework. As the session generated quite a lot of interest, I thought I would try to summarise the session in print for those who weren't there. The article is intended to explain the purpose or position of .NET so that people can be aware of what it is. For this reason, in some areas I have massively simplified things for the sake of understanding rather than complete technical accuracy.



supported across languages, for example, and exceptions raised in some C# code can be caught by some Visual Basic Code. The CLR also allows for inheritance across languages meaning a component developed in one .NET language can be inherited from and extended using any other .NET language. This is achievable because in .NET all things are self-describing in the form of embedded METADATA.

## .NET - what is it?

To quote Dr GUI from Microsoft's .NET web site (<http://msdn.microsoft.com/net/>). "Putting your finger on just what .NET is and isn't at any point in time is hard. But if you understand some of the key motivating forces behind .NET, it makes it a lot easier to get an idea of what it is."

So with this classic quote in mind we will take a look at some of these "motivating forces".

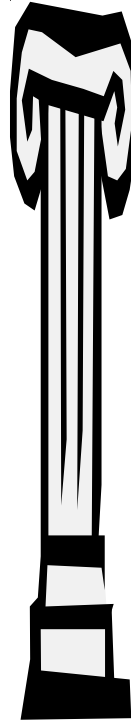
## The .NET Framework

The .NET framework is an environment, on which you can build and deploy the next generation of Windows applications. The .NET framework is designed to sit on top of the operating system and provide a common set of services to any application that chooses to use it. To put it in Delphi terminology, the .NET framework wraps operating systems' functionality into a set of objects in the same way as the VCL. Also like the VCL the framework adds additional functionality on top of the core operating system services. The difference is that unlike the VCL, the .NET framework is designed to be generic across languages and possibly in the future platform. (Microsoft would say the .NET is already cross platform as they see 95, 98, ME, NT & 2000 as different platforms).

So .NET comprises a number of services and facilities that provide a comprehensive skeleton on which to put the flesh of your application's functionality. The framework can be broken down into a number of sections that we will look at individually.

## The Common Language Runtime

The core of .NET is the common language run time (CLR). The CLR is an environment in which all .NET application will run. Any language that wishes to make use of .NET will need to compile down to what Microsoft have called IL or "Intermediate Language". This IL code will then be taken and compiled using the CLR. The thing to point out here is that the final output will be fully executable machine code, .NET is NOT interpreted. The Common Language runtime supplies .NET applications with a host of features that would normally be implemented by individual languages. The most obvious of these are common types, automatic garbage collection, structured exception handling and security. So by default any .NET language would automatically get these features. What it also means is that these features are



## The .NET Base Classes

On top of the CLR sits the .NET base classes. This is a framework that supplies a whole host of objects to achieve common application purposes. This is the VCL like part of the .NET framework. The base classes are split into a number of libraries or namespaces as they have been named under .NET. These namespaces include ADO.NET, the new and quite radically changed version of ADO. Also included are namespaces for dealing with XML, multi-threading, Input/Output, Networking, Security, Diagnostics and others. These namespaces basically contain any classes that might be needed, regardless of the type of development to be undertaken.

## Windows Forms

On top of the base classes sits another set of namespaces that contain classes specifically developed for Windows application development. This set of libraries, currently known as "Windows Forms", contains namespaces of Controls, for Drawing (GDI Etc) and Windows Application Services. This is the section where you will find the .NET equivalent of TForm and TListBox only these classes are available to any .NET language.

## ASP.NET

ASP.NET is the collective name given to the set of .NET namespaces that have been put together for Internet/Web Development. They include new classes based around Web Forms designed to allow point and click RAD Web development for web pages, just like we enjoy with windows forms. Also in this section are the classes to develop Web Services. Web services are designed to be small black box suppliers of functionality whose features (Methods & Properties) are available via what used to be called SOAP but is now called XML protocol.

## .NET Tools

What we have currently looked at is the .NET framework, what is also needed are tools that allow you to develop .NET applications. Microsoft will obviously be providing tools in the form of the next version of Visual Studio called Visual Studio.NET. This will supply five .NET languages: C, C++, VB, C# and Jscript. Other vendors are being encouraged to produce .NET languages that exist either in their own right or as Visual Studio plug-ins. There are currently 17 .NET languages including PERL, Pascal and COBOL. The big question has to be: what about Delphi?

Where will Borland go with this technology? As I write, Borland have made no official announcements as to their intent on this subject, although a reliable source at Borland has told me “Borland are going to be supporting .NET and are working closely with Microsoft to ensure that Borland are able to do this.”

## **What Next?**

If you want to start playing with .NET you can either download the BETA 1 framework (111MB or 11 x 10MB downloads) from <http://msdn.microsoft.com/downloads/default.asp?URL=/code/sample.asp?url=/msdn-files/027/000/976/msdncompositedoc.xml> or get it off MSDN. Visual Studio.NET Beta 1 is also available with MSDN

universal. Hundreds of .NET sites are springing up all over the web and some very active newsgroups frequented by the .NET development team are available at [msnews.microsoft.com](http://msnews.microsoft.com).

I hope this very high level and brief introduction to .NET has at least allowed you to understand the basics of this new and exciting technology.

*Steve Scott is CTO of AscotFirst.Com, based in Gloucestershire and Ireland. A seasoned Delphi and Web trainer, speaker and developer, Scotty is BUG London Group Leader, and father of lots of children.*