VB.NET TUTORIAL

.NETDefined

Before getting deeply into the subject we will first know how Businesses are related to Internet, what .NET means to them and what exactly .NET is built upon. As per the product documentation from a Business perspective, there are three phases of the Internet. The First phase gets back to the early 1990's when Internet first came into general use and which brought a big revolution for Businesses. In the First phase of the Internet Businesses designed and launched their Website's and focused on the number of hits to know how many customers were visiting their site and interested in their products, etc. The Second phase is what we are in right now and in this phase Businesses are generating revenue through Online Transactions. We are now moving into the Third phase of the Internet where profit is the main priority. The focus here is to Businesses effectively communicate with their customers and partners who are geographically isolated, participate in Digital Economy and deliver a wide range of services. How can that be possible? The answer, with .NET.

What is .NET ?

Many people reckon that it's Microsoft's way of controlling the Internet, which is false. .NET is Microsoft's strategy of software that provides services to people any time, any place, on any device. An accurate definition of .NET is, it's an XML Web Services platform which allows us to build rich .NET applications, which allows users to interact with the Internet using wide range of smart devices (tablet devices, pocket PC's, web phones etc), which allows to build and integrate Web Services and which comes with many rich set of tools like Visual Studio to fully develop and build those applications.

What are Web Services?

Web Services are the applications that run on a Web Server and communicate with other applications. It uses a series of protocols to respond to different requests. The protocols on which Web Services are built are summarized below:

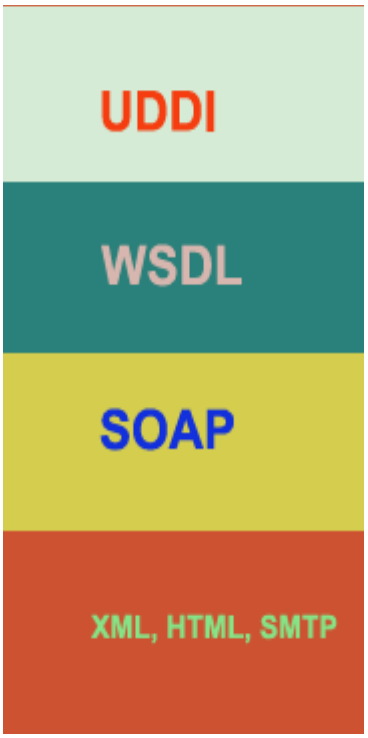
UDDI: Stands for Universal Discovery and Description Integration. It's said to be the Yellow Pages of Web Services which allows Businesses to search for other Businesses allowing them to search for the services it needs, know about the services and contact them.

WSDL: Stands for Web Services Description Language, often called as whiz-dull. WSDL is an XML document that describes a set of SOAP messages and how those messages are exchanged.

SOAP: Stands for Simple Object Access Protocol. It's the communication protocol for Web Services.

XML, **HTTP** and **SMTP**: Stands for Extensible Markup Language, Hyper Text Transfer Protocol and Simple Message Transfer Protocol respectively. UDDI, WSDL and SOAP rely on these protocols for communication.

The image below shows the order of the protocols on which Web Services are built:



Example of a Web Services Application

Let's say a customer accesses a Website and buys something. The Web services of the business will communicate with the inventory system to see if there is enough stock to fulfill the order. If not, the system can communicate with the suppliers to find one or all of the parts that make up the order before filling the order. At all stages the customer will be kept informed via messages. The end result is a seamless system communicating and exchanging information easily regardless of the platform they are all running on. The business don't need to worry about going to the wrong supplier because it asks the Web service running on the supplier system what it does. And the business doesn't have to worry about the other system's methods of handling data because they communicate via SOAP and XML.

Real World Application

Microsoft's passport service is an example of a .NET service. Passport is a Web-based service designed to make signing in to Websites fast and easy. Passport enables participating sites to authenticate a user with a single set of sign-in credentials eliminating the need for users to remember numerous passwords and sign-in names. You can use one name and password to sign in to all .NET Passport-participating sites and services. You can store personal information in your .NET Passport profile and, if you choose, automatically share that information when you sign in so that participating sites can provide you with personalized services. If you use Hotmail for your email needs then you should be very much familiar with the passport service.

To find out more about how Businesses are implementing Web Services and the advantages it is providing please visit Microsoft's Website and check out the case studies published.

What is .NET Built On?

.NET is built on the Windows Server System to take major advantage of the OS and which comes with a host of different servers which allows for building, deploying, managing and maintaining Web-based solutions. The Windows Server System is designed with performance as priority and it provides scalability, reliability, and manageability for the global, Web-enabled enterprise. The Windows Server System integrated software products are built for interoperability using open Web standards such as XML and SOAP.

Core Windows Server System Products include :

SQL Server2000: This Database Server is Web enabled and is designed with priority for .NET based applications. It is scalable, easy to manage and has a native XML store.

Application Center 2000: This product is designed to manage Web Applications.

Commerce Server 2000: This powerful Server is designed for creating E-Commerce based applications.

Mobile Information Server: This Server provides real-time access for the mobile community. Now Outlook users can use their Pocket PC's to access all their Outlook data while they are moving.

Exchange Server 2000: This is a messaging system Server and allows applications on any device to access information and collaborate using XML.

BizTalk Server 2000: This is the first product created for .NET which is XML based and allows to build business process that integrate with other services in the organization or with other Businesses.

Internet Security and Acceleration Server 2000: This Server provides Security and Protection for machines. It is an integrated firewall and Web cache server built to make the Web-enabled enterprise safer, faster, and more manageable.

Host Integration Server 2000: This Server allows for the Integration of mainframe systems with .NET.

When developing real world projects if you don't know how to use the above mentioned Server's which are built for .NET based applications do not worry. Your System Administrator is always there to help you.

Host Integration Server 2000: This Server allows for the Integration of mainframe systems with .NET.

When developing real world projects if you don't know how to use the above mentioned Server's which are built for .NET based applications do not worry. Your System Administrator is always there to help you.

.NET and XML

There is a lot of connection between XML and .NET. XML is the glue that holds .NET together. XML looks similar to HTML which is readable and text-based. XML is a method of putting structured data into a text file. XML is the specification for defining the structure of the document. Around this specification a whole family of optional modules are being developed. The reason why XML is linked so much to .NET is, it's platform independent and is well supported on any environment. To move the data contained in an XML file around different organizations using different software on different platforms it should be packed it into something. That something is a protocol like SOAP.

About SOAP

SOAP, Simple Object Access Protocol is a simple, lightweight protocol for exchanging information between peers in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelop that describes what is in the message and how it should be processed, a set of encoding rules and a convention for representing remoteprocedure calls and responses.

.NET vs Java

Many of us wonder what .NET has to do with Java. Is there any relation between them? Are they similar? and so on. I even hear some people say .NET is Microsoft's answer to Java. I think every language has its own pros and cons. Java is one of the greatest programming languages created by humans. Java doesn't have a visual interface and requires us to write heaps of code to develop applications. On the other hand, with .NET, the Framework supports around 20 different programming languages which are better and focus only on business logic leaving all other aspects to the Framework. Visual Studio .NET comes with a rich visual interface and supports drag and drop. Many applications were developed, tested and maintained to compare the differences between .NET and Java and the end result was a particular application developed using .NET requires less lines of code, less time to develop and lower deployment costs along with other important issues. Personally, I don't mean to say that Java is gone or .NET based applications are going to dominate the Internet but I think .NET definitely has an extra edge as it is packed with features that simplify application development.

I hope the information above puts some light on the technology aspects behind .NET and helps you in getting started.

.NET Framework

.NET is a "Software Platform". It is a language-neutral environment for developing rich .NET experiences and building applications that can easily and securely operate within it. When developed applications are deployed, those applications will target .NET and will execute wherever .NET is implemented instead of targeting a particular Hardware/OS combination. The components that make up the .NET platform are collectively called the .NET Framework.

The .NET Framework is a managed, type-safe environment for developing and executing applications. The .NET Framework manages all aspects of program execution, like,

allocation of memory for the storage of data and instructions, granting and denying permissions to the application, managing execution of the application and reallocation of memory for resources that are not needed.

The .NET Framework is designed for cross-language compatibility. Cross-language compatibility means, an application written in Visual Basic .NET may reference a DLL file written in C# (C-Sharp). A Visual Basic .NET class might be derived from a C# class or vice versa.

The .NET Framework consists of two main components:

- Common Language Runtime (CLR)
- Class Libraries

Common Language Runtime (CLR)

The CLR is described as the "execution engine" of .NET. It provides the environment within which the programs run. It's this CLR that manages the execution of programs and provides core services, such as code compilation, memory allocation, thread management, and garbage collection. Through the Common Type System (CTS), it enforces strict type safety, and it ensures that the code is executed in a safe environment by enforcing code access security. The software version of .NET is actually the CLR version.

Working of the CLR

When the .NET program is compiled, the output of the compiler is not an executable file but a file that contains a special type of code called the Microsoft Intermediate Language(MSIL), which is a low-level set of instructions understood by the common language run time. This MSIL defines a set of portable instructions that are independent of any specific CPU. It's the job of the CLR to translate this Intermediate code into a executable code when the program is executed making the program to run in any environment for which the CLR is implemented. And that's how the .NET Framework achieves Portability. This MSIL is turned into executable code using a JIT (Just In Time) complier. The process goes like this, when .NET programs are executed, the CLR activates the JIT complier. The JIT complier converts MSIL into native code on a demand basis as each part of the program is needed. Thus the program executes as a native code even though it is compiled into MSIL making the program to run as fast as it would if it is compiled to native code but achieves the portability benefits of MSIL.

Class Libraries

Class library is the second major entity of the .NET Framework which is designed to integrate with the common language runtime. This library gives the program access to runtime environment. The class library consists of lots of prewritten code that all the applications created in VB .NET and Visual Studio .NET will use. The code for all the elements like forms, controls and the rest in VB .NET applications actually comes from the class library.