About Web Forms in .NET Framework

By Balaji

About Web Forms in .NET Framework by Balaji

About Web Forms in .NET Framework

Web pages in ASP.Net are called ASP.NET Web Forms which have certain server controls such as text, dropdown list, checkboxes, and buttons. An ASP.NET Web Form looks similar to the web forms in HTML. The only difference is that in ASP.NET, the Web Forms runs at the server side and in HTML the web forms runs at the client side. Apart from this difference an ASP.NET Web Form has more features than an ordinary HTML web form such as:

· The code blocks are processed on the server

· The entire page in ASP.NET is compiled when it is requested for the first time. When you make subsequent requests, the page is not compiled but shown directly in your browser

· ASP.NET Web Forms can contain page directives. Page directives allow you to set the default language and user controls tags for the entire page. You can also turn off session state and ViewState management using page directives

· An ASP.NET Web Form can contain both user controls and Server side Includes (SSIs)

· An ASP.NET Web Form though run on the server can contain client side script such as JavaScript or Jscript

An ASP.NET Web Form first gets compiled into Intermediate Language (IL) and later to the machine language. Not just the server−side code gets compiled but every control and element in the page gets compiled. This enables the ASP.NET Web Form to be in any language that is compatible with Common Language Runtime (CLR) engine. Another advantage with the code getting translated to IL is that the user can use the application in any browser. In addition, you can reduce the processors and server's load when you compile the ASP.NET Web Forms in ASP.NET, thereby, increasing the user's productivity time. Thus ASP.NET allows you to create Web Forms that are platform independent,

language independent, and browser independent.

To access online version of the above article, go to http://www.dotnet−guide.com/webforms.html

Visit http://www.dotnet−guide.com for a complete introduction to .NET framework. Learn about ASP.NET, VB.NET, C# and other related technologies.

Getting Started with Visual Studio.Net

By Balaji

Getting Started with Visual Studio.Net by Balaji

Getting Started with Visual Studio.Net

Visual Studio.Net is a comprehensive Integrated Development Environment (IDE) that is extensively used to develop ASP.Net web applications. In addition, it allows you to create standalone applications, mobile applications and eXtensible Markup Language (XML) Web Services. Visual Studio.Net provides a common platform to build, compile, and run an application. Visual Studio.Net comes with .Net Framework that allows you to install Common Language Runtime (CLR), class libraries, and ASP.NET. Visual Studio.Net has drag−and−drop capabilities for all the controls that make it user−friendly.

Visual Studio.Net comes in three editions: Professional, Enterprise Developer, and Enterprise Architect. The Professional edition helps you to easily build and deploy Windows, Web, and Mobile applications. The Professional edition, that has built−in ADO.NET and Visual Database Tools, provide support for the creation of professional data−driven software. The Enterprise Developer allows you to use the .NET Framework and Microsoft Windows Server 2003 to develop distributed applications with improved deployment capabilities, security, reliability, and performance.

As Visual Studio.Net includes Common Language Runtime engine, the applications in Visual Studio.Net are language independent and platform independent. You can program the applications with languages such as Visual Basic.Net, Visual C#.Net, Visual C++.Net, and Visual J#.Net. In addition you can use languages such as COBOL, FORTRAN, Java, Eiffel, and Mercury in your applications. You can also use scripting languages such as VBScript and JavaScript.

Visual Studio.Net 2003 now has some of the advanced features such as integration with .Net Compact Framework, support for latest web services, and integration with Enterprise Instrumentation Framework. The integration with .Net Compact Framework allows you to develop and deploy applications for smart devices. Visual Studio.Net 2003 supports some of the most advanced web services such as WS−Routing, WS−Security, WS−Attachments, and Direct Internet Message Encapsulation. Similarly, the integration of Enterprise Instrumentation Framework with Visual Studio.Net 2003 allows you to monitor the application while you run it.

All the features mentioned above are available in all the editions of Visual Studio.Net. For the integration of Enterprise Instrumentation Framework, you have to download it from MSDN.

To access online version of the above article, go to http://www.dotnet−guide.com/visualstudio.html

Visit http://www.dotnet−guide.com for a complete introduction to .NET framework. Learn about ASP.NET, VB.NET, C# and other related technologies.