

ASP.NET

Guia do Desenvolvedor

Felipe Cembranelli

Novatec Editora

www.novateceditora.com.br

1

Introdução à plataforma .NET

A plataforma Microsoft .NET

.NET é uma nova plataforma de software para desenvolvimento de aplicações. Ela oferece suporte para diversas linguagens de programação, as quais compartilham uma biblioteca de classes que oferece serviços básicos. Aplicações .NET não são compiladas em código nativo de máquina, como, por exemplo, código Intel x86, e sim em uma linguagem intermediária chamada MSIL (Microsoft Intermediate Language), que é executada em um ambiente de execução (máquina virtual) chamado Common Language Runtime (CLR).

O Framework .NET

O Framework .NET é um conjunto de ferramentas para criar, construir e testar aplicações .NET. Consiste em dois componentes principais: Common Language Runtime (CLR) e as bibliotecas de classe Base Class Library (BCL). A BCL oferece recursos para uma série de necessidades de desenvolvimento, como E/S de arquivo e banco de dados, XML e SOAP. O Framework .NET pode ser obtido em www.microsoft.com/net. Veja na figura 1.1 uma visão geral do Framework .NET.

Common Language Runtime (CLR)

A CLR é um ambiente de tempo de execução (runtime) que realiza tarefas, tais como: gerenciamento de memória, coleta de lixo, segurança, tratamento de erro, controle de versão e suporte de instalação. Realiza a interface entre a aplicação e o sistema operacional. O código que é executado nesse ambiente de runtime é chamado de **Código Gerenciado** (“**Managed Code**”), enquanto aquele que é executado fora é chamado de **Código Não Gerenciado** (“**Unmanaged Code**”).

Common Language Specification (CLS)

A CLS é uma especificação que estabelece as regras que um compilador deve seguir para que seu código compilado seja capaz de ser executado dentro da CLR. Dessa maneira, o código compilado de diferentes linguagens de programação pode ser executado dentro do mesmo ambiente de tempo de execução.

Códigos gerados por um compilador compatível com a CLS podem ser utilizados por qualquer outro aplicativo também gerado por um compilador compatível com a CLS. Isso significa, por exemplo, que uma classe escrita em uma linguagem pode ser derivada em uma outra linguagem, e é possível instanciar em uma linguagem um objeto escrito em uma outra linguagem .NET (interoperabilidade entre linguagens).

Common Type System (CTS)

O ambiente .NET oferece um sistema unificado de tipos que é utilizado por todas as linguagens compatíveis com a CLS. Uma variável do tipo “String” possui as mesmas características, como estrutura e número de bytes, tanto no VB.NET quanto no C#. O Common Type System (Sistema Comum de Tipo) define as regras que o CLR segue com relação aos tipos básicos da linguagem. Com um modelo orientado a objetos, esse sistema unificado de tipos permite estender os tipos predefinidos, adicionando novos tipos por meio de herança, além de permitir a interação entre programas escritos em diferentes linguagens de programação.



Figura 1.1 – Visão geral dos elementos do Framework .NET.

Comunicação entre diferentes linguagens

O Common Language Runtime (CLR) oferece suporte para a interoperabilidade entre diferentes linguagens de programação, desde que essas linguagens obedeçam à especificação CLS. Isso significa que um componente desenvolvido em VB.NET pode herdar métodos de um componente escrito em C#, por exemplo.

Microsoft Intermediate Language (MSIL)

Na plataforma .NET, o código-fonte é compilado para código intermediário denominado Microsoft Intermediate Language (MSIL). Com o código MSIL são criados os metadados, contendo informações autodescritivas do código, como os tipos declarados e os métodos implementados. O MSIL e os metadados são armazenados em um mesmo arquivo executável portátil (PE). Os metadados substituem a biblioteca de tipo (Type Library) e a IDL (Interface Definition Language) do modelo COM/COM+. O ambiente de tempo de execução (CLR) é o responsável pela conversão do código em MSIL para o código binário de máquina. A conversão é feita por compiladores just-in-time (padrão) ou em tempo de instalação do assembly.

Assembly

Os assemblies são executáveis (DLLs ou Exe) utilizados para “empacotar” aplicações no ambiente .NET, possuindo todas as informações necessárias para o funcionamento da aplicação ou do componente. Assemblies facilitam a distribuição/instalação e o controle de versão, além de definir as configurações de segurança. É composto basicamente de:

- Código compilado MSIL da classe (pode possuir uma ou mais classes).
- Metadados de Tipos (Type Metadata): contém informações sobre os tipos no assembly com um mapeamento para o módulo contendo o tipo.
- Arquivos utilizados pela aplicação (.gif, .bmp etc.).
- Manifesto: descreve como os elementos do assembly se relacionam uns com os outros, informações sobre a versão e referências a outros assemblies.

Namespace

Namespace é um recurso utilizado para organizar hierarquicamente classes, evitando ambigüidades e simplificando referências a estas na utilização de bibliotecas de classes.

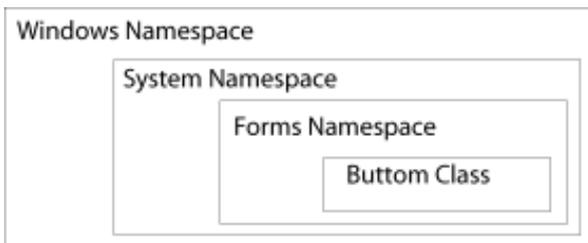


Figura 1.2 – Representação da hierarquia de namespaces.

Por exemplo, a classe `Button` está contida no namespace `System.Windows.Forms`. Isto significa que `Button` está contida no namespace `Forms`, que por sua vez está contido no namespace `Windows`, o qual está contido no namespace raiz `System`. A declaração completa `System.Windows.Forms` é chamada de “Fully Qualified Name”.

Biblioteca de classes do Framework .NET

O Base Class Library (BCL) é uma extensa coleção de classes, interfaces e estruturas que podem ser utilizadas por qualquer linguagem .NET. Veja a seguir uma descrição dos principais namespaces e suas respectivas classes que podem ser utilizados no ASP.NET.

System

Contém classes fundamentais e classes-base que definem tipos de dados, eventos e manipuladores de eventos, interfaces, atributos e exceções. Oferece serviços de conversão de tipos de dados, método de manipulação de parâmetros, métodos matemáticos, métodos de chamada de programas locais e remotos, gerenciamento do ambiente de aplicação e supervisão de aplicações gerenciadas e não gerenciadas.

Principais classes: `Array`, `Console`, `Environment`, `Exception`, `Math`, `Object`, `OperatingSystem`, `Random`, `String`, `Boolean`, `Byte`, `Char`, `Decimal`, `Double`, `DateTime`, `Type` e `TimeZone`.

System.Web

Fornecer classes e interfaces para comunicação browser/server. Esse namespace inclui a classe `HttpRequest`, que oferece informação sobre a requisição HTTP corrente, a classe `HttpResponse`, que gerencia a saída HTTP para o cliente, e a classe `HttpServerUtility`, que dá acesso às funcionalidades do servidor. `System.Web` também oferece classes para manipulação de cookie, transferência de arquivo, informação de exception e controle de cache.

Principais classes: `HttpApplication`, `HttpBrowserCapabilities`, `HttpContext`, `HttpCookie`, `HttpRequest`, `HttpResponse`, `HttpServerUtility` e `TraceContext`.

System.Web.UI.HtmlControls

Contém classes que permitem a utilização de controles HTML no servidor. Os controles HTML rodam no servidor e retornam tags HTML-padrão suportadas por qualquer browser. Isto permite a programação de elementos HTML em uma página Web.

Principais classes: `HtmlAnchor`, `HtmlButton`, `HtmlForm`, `HtmlImage`, `HtmlInputButton`, `HtmlInputCheckBox`, `HtmlTable`, `HtmlSelect` e `HtmlTextArea`.

System.Web.UI.WebControls

Contém classes que permitem a utilização de controles de Servidor Web. Os controles de servidor Web executam no servidor e incluem controles de formulário, tais como botões e caixas de texto, e controles mais sofisticados, como Calendar.

Principais classes: AdRotator, Button, Calendar, CheckBox, DataGrid, DataList, DropDownList, FontInfo, HyperLink, Image, ImageButton, Label, ListBox, Panel, RadioButton, Style, Table, TextBox, Xml.

Onde obter o Framework .NET

Para trabalhar com o ASP.NET, é preciso instalar o Microsoft .NET Framework Software Development Kit (SDK), que pode ser obtido em <http://www.microsoft.com/net>.

A instalação do Framework .NET SDK é feita por meio do seguinte arquivo:

- SetupSdk.exe: (138 MB) Microsoft .NET Framework Software Development Kit (SDK) – inclui o Framework .NET e todas as ferramentas para compilar, depurar e executar aplicações .NET, além de documentação, exemplos e ferramentas de linha de comando.

Requerimentos de instalação

Software

- Sistema Operacional.
 - Microsoft Windows NT 4.0 com SP 6a.
 - Microsoft Windows 2000 (SP 2 recomendado).
 - Microsoft Windows XP Professional.
- Microsoft Internet Explorer 5.01 ou superior (pode ser obtido em <http://www.microsoft.com/windows/ie/default.htm>).
- Microsoft Data Access Components 2.7 (MDAC): inclui componentes de acesso a dados como o ADO.NET (pode ser obtido em <http://www.microsoft.com/data>).
- Internet Information Services 5.0 (IIS) ou superior: servidor Web que acompanha o Windows 2000 (deve ser instalado antes do Framework .NET).

Hardware

- Pentium 133 MHz com 128 MB (mínimo).
- Pentium 133 ou superior com 256 MB ou mais (recomendado).

Instalando o Framework .NET em produção

Pode ser feito por meio do seguinte arquivo (obtido em www.microsoft.com/net):

- Setupredist.exe: (20.4 MB) Microsoft .NET Redistributable Package – contém somente o necessário para executar aplicações .NET (CLR, Class Library e ASP.NET).



Figura 1.3 – Página inicial da documentação do Framework .NET.