

Utilizando Themes no ASP.NET 2.0 – Parte 2

por **Ricardo Oneda**
28 de novembro de 2005

Introdução

Na primeira parte desta série de artigos, vimos o que são temas e como definir *skins* para serem aplicados nos controles do ASP.NET. O exemplo feito determinava a aparência para todos os controles de um determinado tipo que estivessem na página na qual o tema foi aplicado. Por exemplo, se houvessem 3 controles do tipo *TextBox*, todos teriam a aparência definida no arquivo **Skin1.skin**. Mas nem sempre é isso o que se deseja. Neste artigo veremos como ter mais de um *skin* para um mesmo tipo de controle.

Obs: este artigo é baseado na versão Beta 2 do .NET Framework 2.0/Visual Studio 2005. O conteúdo está sujeito a alterações até o lançamento da versão final dos produtos.

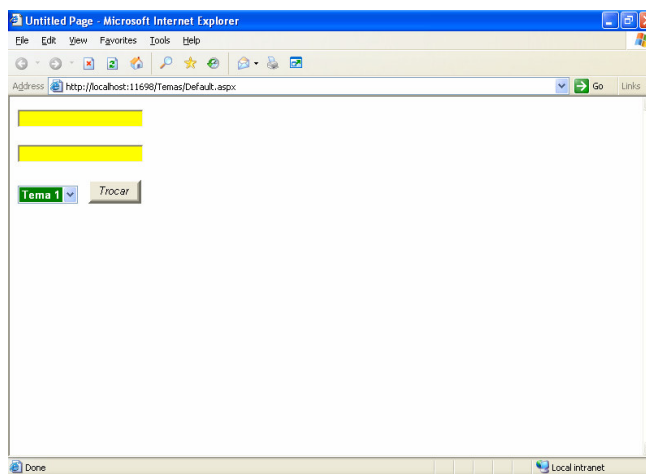
Default Skin X Named Skin

Como dito anteriormente, até agora definimos *skins* que se aplicam a todos os controles de um determinado tipo. Este tipo de *skin* é o mais simples e é conhecido como **Default Skin**, ou seja, *skin* padrão. Para que possamos ter mais de um *skin* definido para um mesmo tipo de controle, precisamos utilizar os **Named Skin**, ou seja, *skin* nomeados. Os **Named Skin** possuem o atributo **SkinID** definido na declaração do *skin* e permitem que controles do mesmo tipo possuam *skins* diferentes.

Abra o Web Site **Temas** criado na primeira parte deste artigo. No *Solution Explorer*, clique com o botão direito sobre a página **Default.aspx** e escolha a opção *View Designer*. Adicione um segundo *TextBox* com as seguintes propriedades:

Controle	ID	Text
TextBox	txtCaixa2	-

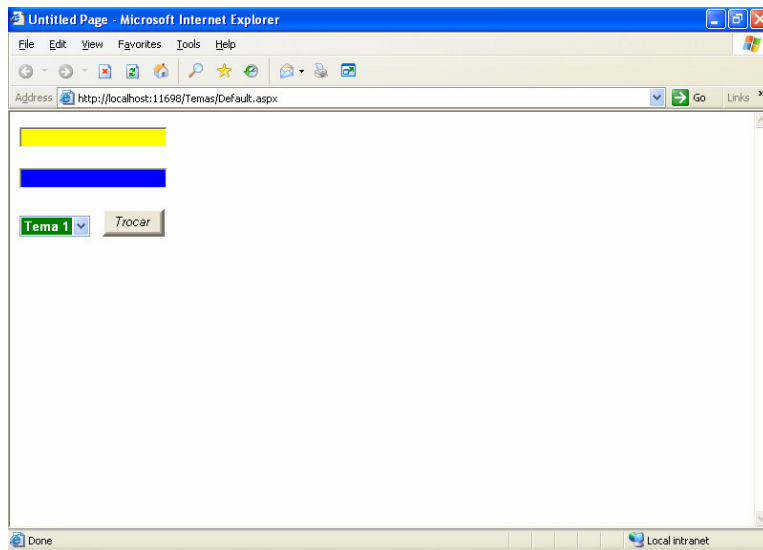
No *Solution Explorer*, clique com o botão direito sobre a página **Default.aspx** e escolha *View in Browser*. Você verá que os dois controles *TextBox* possuem a mesma aparência. Isso acontece porque ambos estão utilizando o **Default Skin**:



Agora, abra o arquivo **Skin1.skin** no Visual Studio 2005 e acrescente a seguinte linha:

```
<asp:TextBox SkinID="CaixaTexto2" ForeColor="White" BackColor="Blue" Runat="Server" />
```

Note a propriedade **SkinID**: é ela que define um *skin* como sendo **Named Skin**. Volte para o *Designer* da página **Default.aspx** e configura a propriedade **SkinID** do controle **txtCaixa2** para **CaixaTexto2**. Estamos dizendo ao ASP.NET que este controle específico irá utilizar um **Named Skin** chamado **CaixaTexto2**, ao invés de utilizar o **Default Skin**. No *Solution Explorer*, clique com o botão direito sobre a página **Default.aspx** e escolha *View in Browser* e perceba a diferença:



Conclusão

Neste artigo vimos como podemos ter mais de um *skin* definido para um mesmo tipo de controle através do uso de **Named Skins**, deixando nossa aplicação muito mais flexível e personalizável. No próximo artigo veremos o que são e quando utilizar **StyleSheetThemes**.

Referências

Código fonte do exemplo deste artigo

<http://www.msdnbrasil.com.br/sharepedia/visualizarartigo.aspx?id=62124>

Utilizando Themes no ASP.NET 2.0 - Parte 1

<http://www.msdnbrasil.com.br/sharepedia/visualizarartigo.aspx?id=62128>

Creating Web Application Themes in ASP.NET 2.0

<http://msdn.microsoft.com/asp.net/reference/ui/default.aspx?pull=/library/en-us/dnvs05/html/themes.asp>

ASP.NET Themes and Skins Overview

<http://msdn2.microsoft.com/en-us/library/ykzx33wh.aspx>

Sobre o Autor

Ricardo Oneda Pereira de Toledo é formado em Processamento de Dados pela Faculdade de Tecnologia de São Paulo, é *MCAD – Microsoft Certified Application Developer*, com ênfase em aplicações Web com C# e Banco de Dados SQL Server 2000 e é *MVP – Most Valuable Professional* em ASP/ASP.NET. Atualmente trabalha como Analista de Sistemas, desenvolvendo aplicações Web nas linguagens C e C++. Visite o meu blog em <http://thespoke.net/blogs/oneda/default.aspx>