

Utilizando Themes no ASP.NET 2.0 – Parte 4

por **Ricardo Oneda**
18 de dezembro de 2005

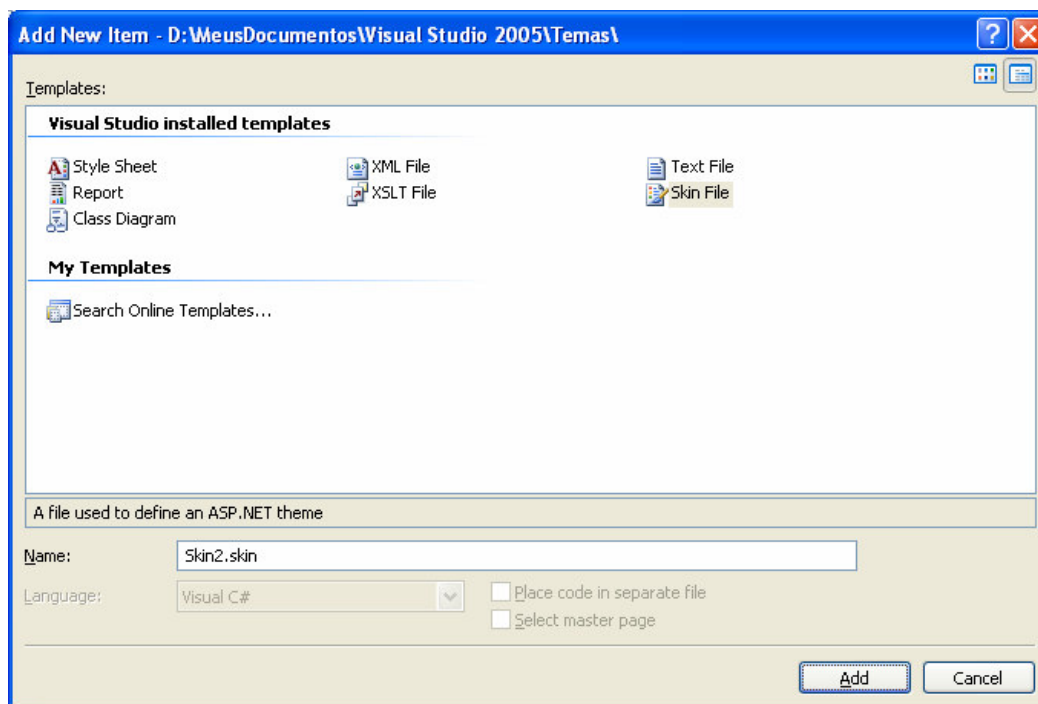
Introdução

Nos últimos artigos, foram mostrados o que são os **Themes** do ASP.NET 2.0 e como podemos utilizá-los, de forma flexível, para manter a aplicação consistente no que se refere a sua apresentação visual. Nesta, que é a última parte da série de artigos, será mostrado como que a aplicação web pode carregar um **Theme** dinamicamente, em tempo de execução.

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.

Adicionando um novo arquivo Skin

Se você fez o exemplo da parte 3 deste artigo, primeiro é necessário desfazer uma alteração. No Visual Studio 2005, abra o **Web Site Temas** criado na parte 1 deste artigo. Configure a propriedade **Theme** da página **Default.aspx** para **Tema1** e deixe vazia a propriedade **StyleSheetTheme**. Feito isso, para que nossa aplicação tenha dois arquivos **Skin**, no *Solution Explorer*, clique com o botão direito sobre a pasta **App_Themes**, escolha *Add Folder → Theme Folder* e dê o nome **Tema2** para a pasta. Clique com o botão direito sobre a pasta **Tema2** e escolha *Add New File*. Na janela que se abre, escolha *Skin File* e dê o nome **Skin2.skin**, conforme figura, e clique em *Add*:



No arquivo *skin* que foi criado, adicione o código abaixo:

```
<asp:DropDownList BackColor="Black" ForeColor="white" Font-Bold="true" runat="server" />
<asp:TextBox BackColor="Gray" Runat="Server" />
<asp:Button BorderWidth="5px" Runat="Server"/>
<asp:TextBox SkinID="CaixaTexto2" ForeColor="White" BackColor="Red" Runat="Server" />
```

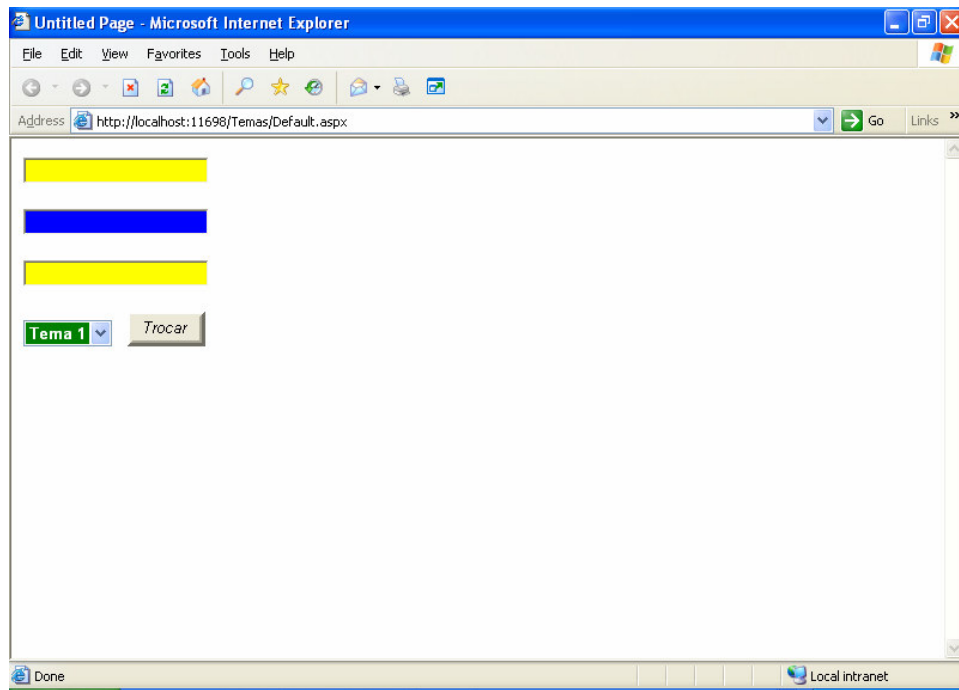
Perceba que este código é muito parecido com o código do arquivo **Skin1.skin**, criado na primeira parte do artigo. As únicas diferenças são os valores dos atributos dos controles.

Vá para o *Design* da página **Default.aspx**, selecione o controle *DropDownList* **ddlTema** e configure a propriedade **AutoPostBack** para **true**. Isso fará com que seja feito um *post* da página a cada vez que selecionarmos um valor do **DropDownLost**. Agora, abra o arquivo Default.aspx.cs e adicione o seguinte código para tratamento do evento **PreInit**:

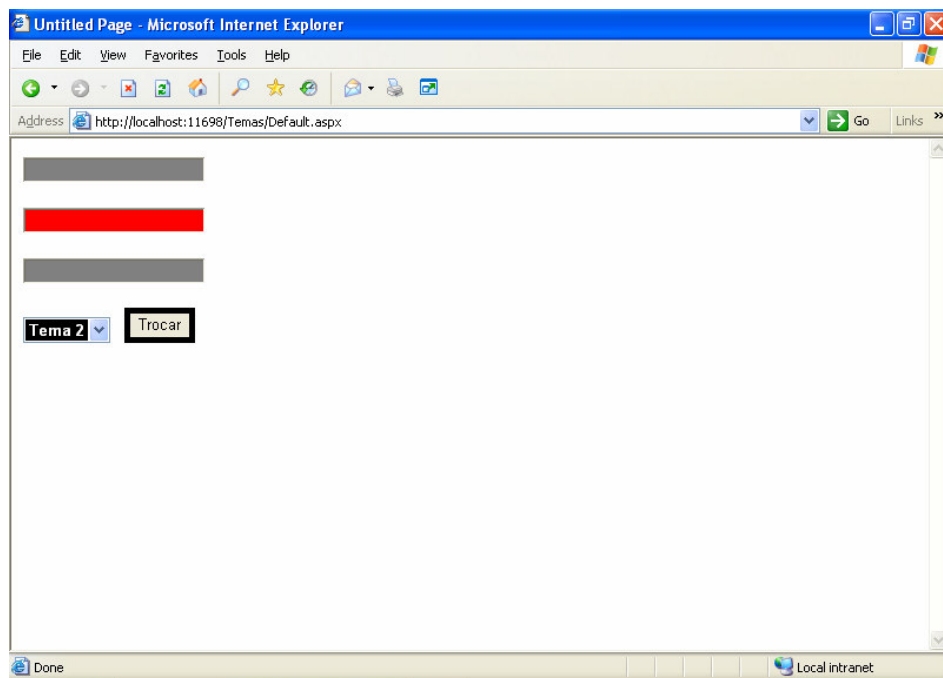
```
void Page_PreInit(Object sender, EventArgs e)
{
    if ((Request["ddlTema"] != null))
    {
        if (Request["ddlTema"].ToString() == "Tema 1")
        {
            Page.Theme = "Tema1";
        }
        else
        {
            Page.Theme = "Tema2";
        }
    }
}
```

O código acima associa um tema à propriedade **Theme** da página de acordo com o item selecionado pelo usuário. Um **Theme**, assim como uma **Master Page**, só pode ser associado a uma página antes do evento **Init**. Assim, o melhor momento para definir um **Theme** dinamicamente é no evento **PreInit**, como visto.

No *Solution Explorer*, clique com o botão direito sobre a página **Default.aspx** e escolha *View in Browser*:



Agora, selecione **"Tema 2"** no *DropDownList* e perceba que o novo tema é aplicado à página:



Conclusão

Nesta última parte desta série de artigos, vimos como associar um tema dinamicamente. Isso como podemos deixar as aplicações ASP.NET personalizáveis de forma fácil e rápida.

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>

Utilizando Themes no ASP.NET 2.0 - Parte 2

<http://www.msdnbrasil.com.br/sharepedia/visualizarartigo.aspx?id=62145>

Utilizando Themes no ASP.NET 2.0 - Parte 3

<http://www.msdnbrasil.com.br/sharepedia/visualizarartigo.aspx?id=62163>

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/onedade/default.aspx>