

Por que eu ainda não escrevi nenhum artigo sobre o SilverLight ?

Principalmente porque estavam ocorrendo tantas mudanças em um intervalo de tempo tão pequeno que achei temerário tratar do assunto enquanto uma versão estável não fosse oficialmente liberada. Atualmente temos o **SilverLight na versão 3.0** e é sobre esta versão que irei tratar neste artigo.

Para poder criar aplicações usando **SilverLight 3.0** você deve possuir o **Visual Studio 2008** com **service pack 1** instalado e além disso deve possuir os seguintes recursos: (Se não tiver o service pack 1 pegue aqui: [Microsoft Visual Studio 2008 Service Pack 1 \(Installer\)](#))

- [Microsoft® Silverlight™ 3 Tools for Visual Studio 2008 SP1](#)
- [Install Microsoft Expression Blend 3 + SketchFlow Trial](#) - Permite criar graficamente interfaces com o usuário para aplicações **Silverlight 3**;
- [Install Deep Zoom Composer](#) - Permite que você prepare imagens para uso com o característica **Deep Zoom** (imagens dentro de imagens) no SilverLight 3;
- [Download Silverlight Toolkit](#) - Contém controles SilverLight, componentes e utilitários que podem ser baixados e usados em suas aplicações SilverLight 3;
- [Download .NET RIA Services](#) - Simplifica o padrão tradicional de n-camadas trazendo junto as plataformas ASP .NET e SilverLight 3. .NET RIA Services fornece um padrão para escrever a lógica da aplicação que roda em mid-tier e controles para acessar dados para consultas, alterações e outras operações.

Você pode obter todas as informações para quem esta iniciando além de vídeo aulas, artigos e dicas no site: <http://silverlight.net/getstarted/>

Vamos começar tentando dar uma definição clara sobre o que vem a ser o **SilverLight**.

O **SilverLight** teve origem na tecnologia "**WPF/E**" (*E de Everywhere*), originalmente uma extensão do projeto **WPF - Windows Presentation Foundation**, e por isso possui um subconjunto de funções do **WPF**.

Nota: O WPF tem o objetivo de criar aplicações com uma interface rica usando a linguagem XAML que rodam no Windows, em aplicações desktop e também possível criar aplicações WPF que rodam em um navegador (aplicações XBAPs). Todas elas rodam apenas na plataforma Microsoft.

O **SilverLight** é uma tecnologia que permite criar aplicações para web com uma interface rica, sendo que todas as suas funcionalidades são cross-browser e rodam em diversas plataformas. Atualmente o SilverLight esta disponível para **Windows e Mac** e roda nos navegadores **Internet Explorer , FireFox e Safari**.

Além disso o projeto **MoonLight** que está em desenvolvimento tem o objetivo de permitir rodar o **SilverLight no Linux**.

O **SilverLight** também usa a linguagem XAML e os controles muito semelhante aos do WPF e possui também a independência entre o código e a interface da aplicação de forma que a interface é colocada em um arquivo XML chamado **XAML** (*Extensible Application Markup Language*) e o código fica em um arquivo distinto escrito na linguagem **.net** que você escolheu para desenvolver: **VB .NET** ou **C#**.

Podemos pensar no **Framework SilverLight** como uma combinação de 3 arquiteturas: o *plug-in de navegador*, o *framework de apresentação* e o *.NET Framework*.

Com esta combinação o SilverLight se propõe a preencher a lacuna entre a interface do usuário (UI), usando uma linguagem declarativa (**XAML**) e programação funcional usando o .NET Framework.

O plug-in SilverLight por ser muito leve fornece a interação necessária com o navegador permitindo que a aplicação SilverLight rode em múltiplas plataformas. Ele precisa ser instalado pelo usuário antes que as aplicações possam ser visualizadas no navegador.

As aplicações SilverLight são implementadas como **objetos embutidos em uma página web**. Quando o navegador encontra um objeto silverlight na página, o plug-in faz o download de um pacote **XAP** (*Um arquivo XAP é basicamente uma aplicação SilverLight compilada.*) a partir do servidor web que contém os binários e os recursos para a aplicação SilverLight e então começa a execução do código no interior da página web.

As aplicações SilverLight são executadas como aplicações do lado do cliente sem a necessidade de dar um refresh no navegador para atualizar a interface. (*Como possuem a .NET Framework embutida as aplicações SilverLight podem ser integradas com controles e serviços do servidor.*)

Resumindo : As aplicações SilverLight são empacotadas em um arquivo com extensão XAP e executadas no navegador do cliente através de um **plug-in (4 MB)** instalado na máquina local. (*Por questões de segurança o SilverLight em acesso limitado a máquina do usuário.*)

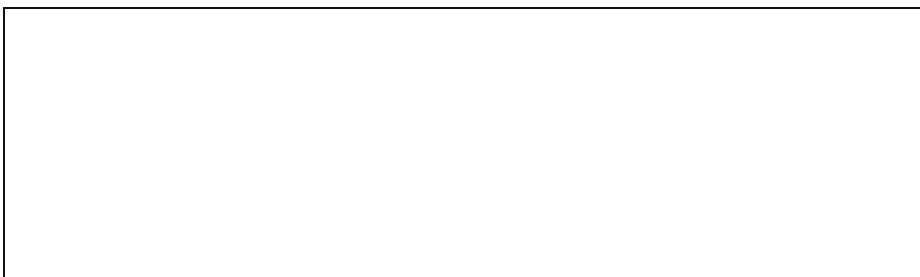
Criando o primeiro programa SilverLight

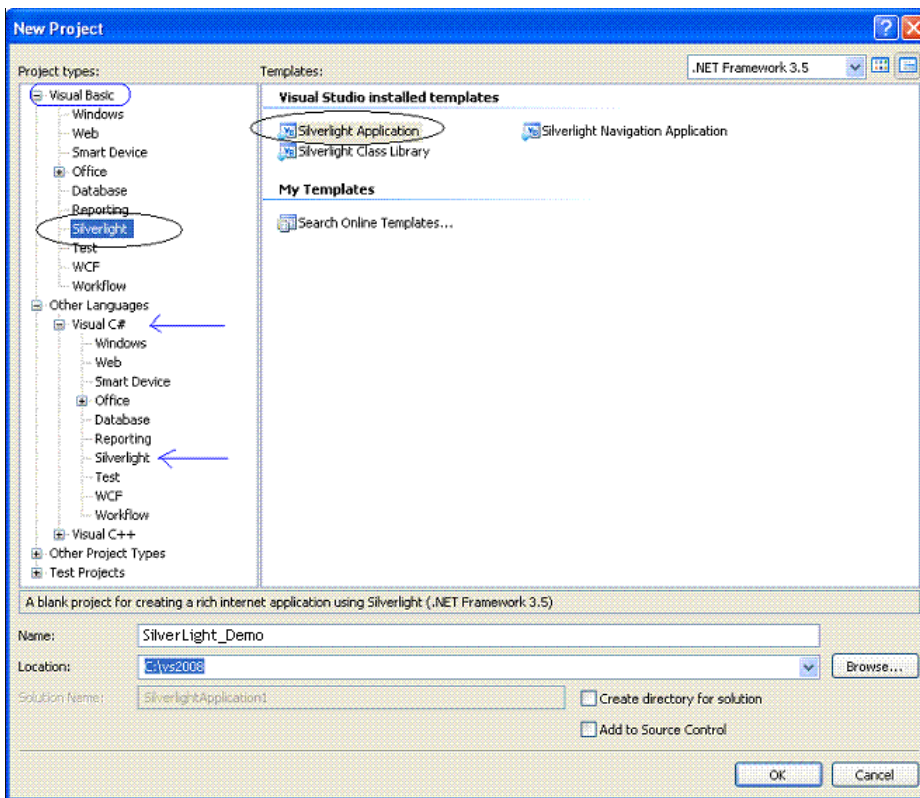
Vamos iniciar criando uma aplicação SilverLight bem simples para podermos analisarmos a estrutura básica de um projeto SilverLight.

Abra o **Visual Studio 2008** e no menu **File** selecione **New Project**;

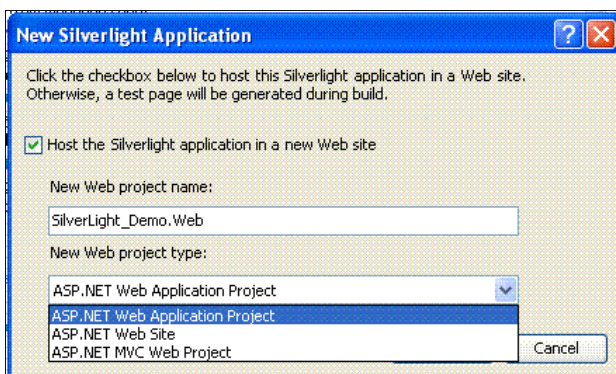
Na janela New Project vamos selecionar aceitar o valor para a versão 3.5 da plataforma .NET, selecionar a linguagem **Visual Basic** e o tipo de projeto **SilverLight** e na janela **Templates** selecionar **SilverLight Application** informando o nome **SilverLight_Demo**;

Nota: Observe que temos as mesmas opções para linguagem **Visual C#**.





Ao clicar no botão **OK** será apresentada a janela a seguir:

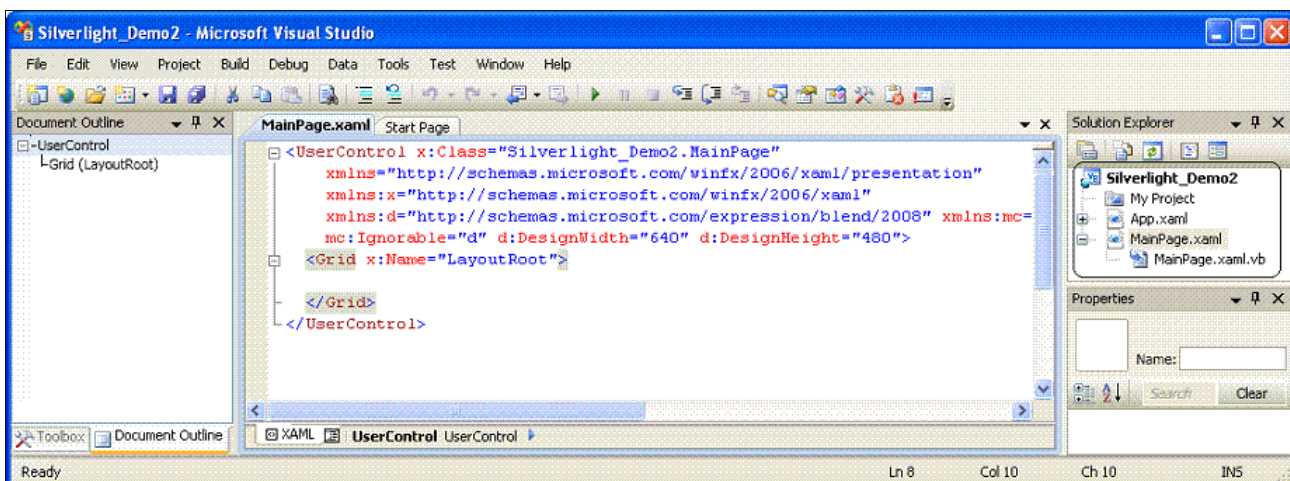


Nesta janela vamos definir como a aplicação **SilverLight** será executada pois um projeto SilverLight não pode ser executado sozinho:

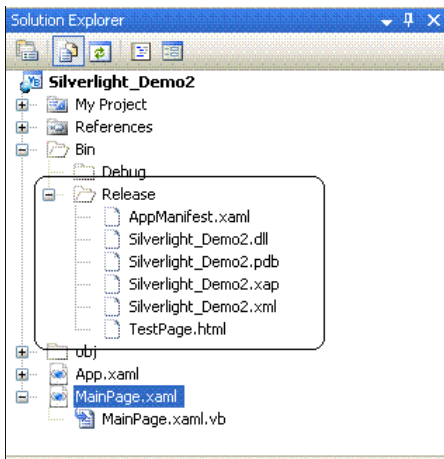
Temos duas opções:

1- Se **não marcarmos** a opção **Host the SilverLight application in a new Web site**, o próprio projeto irá gerar uma página HTML para ser o host da aplicação;

Se escolhermos esta opção teremos o projeto exibido na figura abaixo. **Mas aonde esta a página HTML que irá hospedar a aplicação SilverLight no Projeto ?**



Se compilarmos a aplicação e ativarmos a visualização de todos os arquivos iremos ver na pasta bin a página HTML gerada para hospedar o projeto. Ela não pode ser modificada pois será gerada a cada compilação.



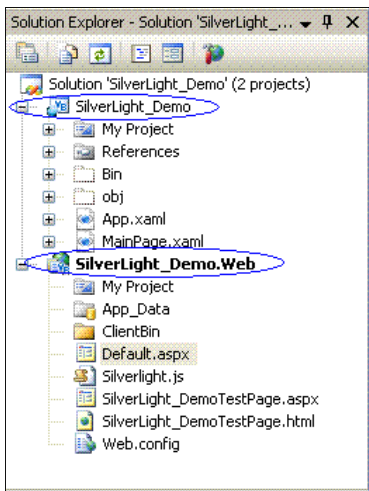
Mas então qual a vantagem em se usar esta opção ?

Embora esta opção seja menos flexível ela pode ser usada quando queremos hospedar o projeto SilverLight em um servidor que não tenha ASP .NET (Servidor Apache).

Dessa forma nesta opção temos que colocar apenas o arquivo XAP e a página HTML no servidor.

O arquivo XAP já contém todos os recursos para executar o projeto SilverLight como ja mencionei anteriormente.

2- Se **marcamos** a opção, será criado na Solution um web site que será o host da aplicação. Além disso podemos escolher o tipo de projeto que será usado onde um projeto do tipo ASP .NET Web Application fara com que um projeto ASP .NET hospede o controle SilverLight. Esta opção é indicada para criar um controle SilerLight integrado ao projeto ASP .NET.



Escolhendo esta opção teremos a criação de dois projetos em uma única solução: O projeto ASP .NET e o controle SilverLight conforme mostra a figura ao lado.

- **SilverLight_Demo** é o controle SilverLight;

- **SilverLight_Demo.Web** é o projeto ASP .NET que hospeda o controle;

Observe que no projeto ASP .NET temos uma pasta chamada **ClientBin**. **Qual a sua finalidade ?**

Na pasta **ClientBin** será colocado o projeto **SilverLight** após a sua compilação. Vamos dar um **Build** na aplicação e espisar a pasta **ClientBin** que agora esta vazia...

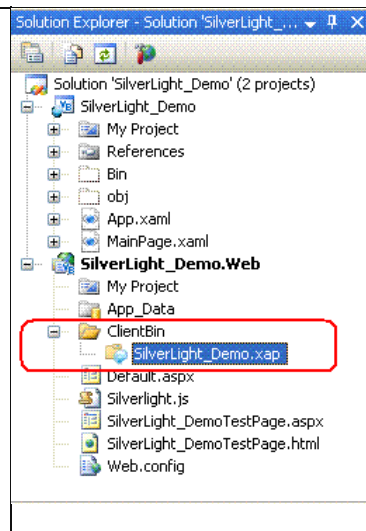
Observe que a após a compilação temos na pasta **ClientBin** o arquivo **SilverLight_Demo.xap** conforme a figura ao lado;

O arquivo **SilverLight_Demo.xap** contém todos os recursos necessários para executar o controle SilverLight. Na verdade ele é um arquivo zipado renomeado para XAP.(É este arquivo que o plug-in SilverLight baixa e executa.)

Se você renomear o arquivo para **SilverLight_Demo.xap** e abrir irá visualizar o seguinte conteúdo:

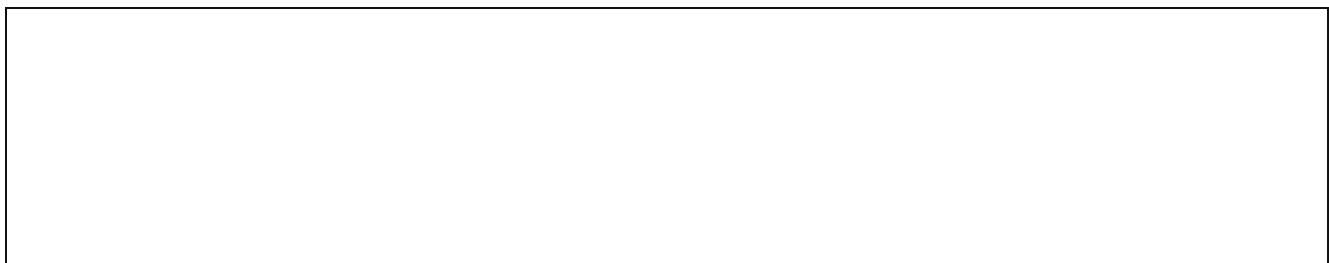
Nome	Tamanho	Comprimido	Tipo	Modificado	CRC32
Pasta					
AppManifest.xaml	373	208	Windows Markup File	2/10/2009 09:19	39611ED7
SilverLight_Demo.dll	10.240	3.493	Extensão de aplicativo	2/10/2009 09:19	63F1A6D3

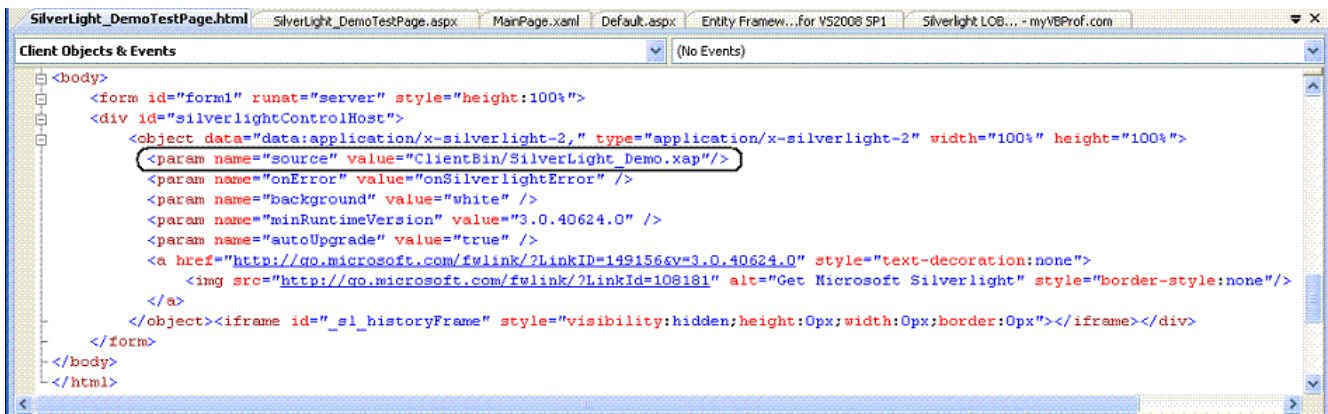
Total 10.613 bytes em 2 arquivos



Observe também que temos uma página **SilverLight_DemoTestPage.aspx** e uma página **SilverLight_DemoTestPage.html**. Essas são as duas formas pelas quais podemos exibir uma interface **SilverLight** no navegador.

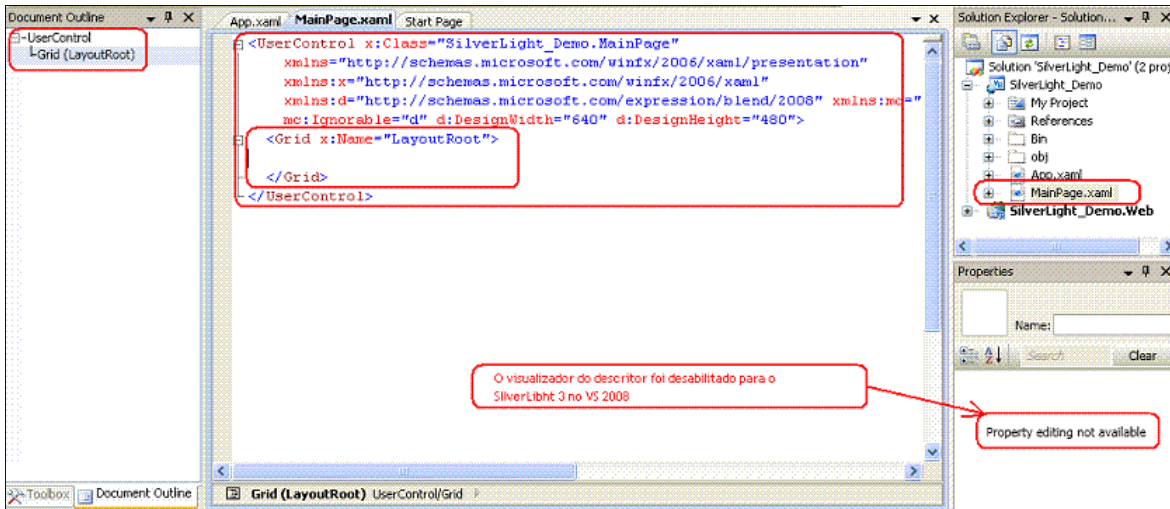
Vamos abrir a página **SilverLight_DemoTestPage.html** e observar o trecho de código na página que faz a chamada ao arquivo **SilverLight_Demo.xap** na pasta **ClientBin**:





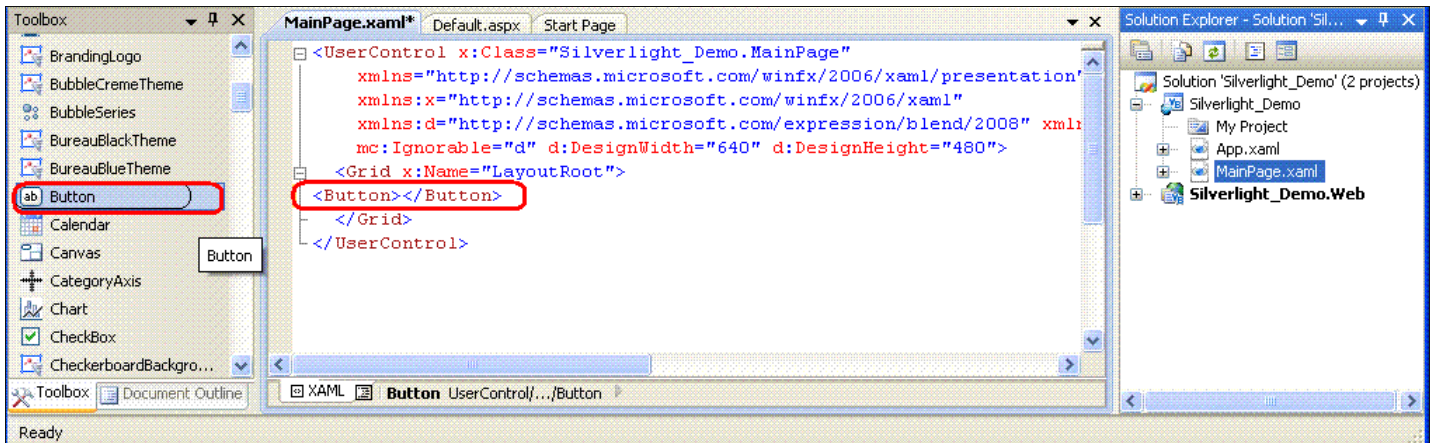
Se espiarmos o arquivo **SilverLight_DemoTestPage.aspx** veremos também o mesmo tipo de chamada ao arquivo **XAP** que é o projeto **SilverLight**.

Vamos abrir agora o arquivo **MainPage.xaml**:



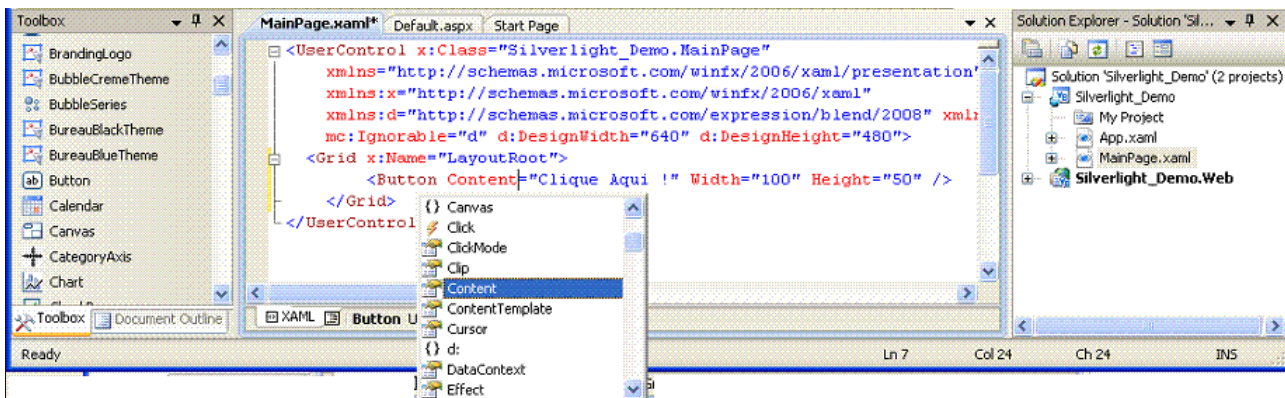
Podemos ver que a página **SilverLight** é um **UserControl** e dentro do **UserControl** temos um **Grid**. Como infelizmente, o designer foi desabilitado para o SilverLight 3 no VS 2008 (a equipe da Microsoft decidiu isso devido a diversos bugs no designer) não temos a visualização da página.

Apenas para mostrar os controles para o **SilverLight** abra a **ToolBox** e arraste um controle **Button** para o interior do **Grid** conforme a figura abaixo:



Uma outra forma de incluir objetos e posicionar o mouse onde o controle deve ser incluído e digitar diretamente o comando desejado. O **IntelliSense** do Visual Studio irá exibir uma relação com os **controles/comandos** sugerindo um valor a partir do texto que você digitou.

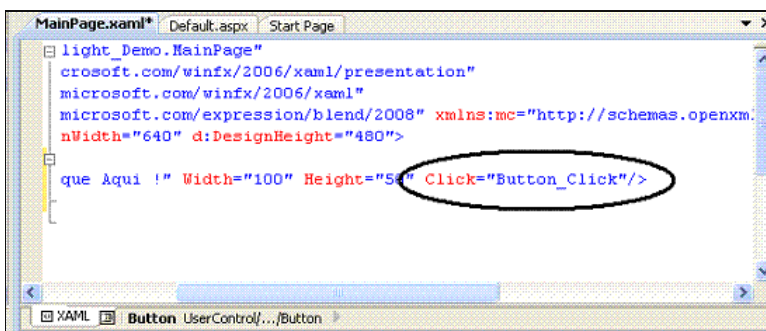




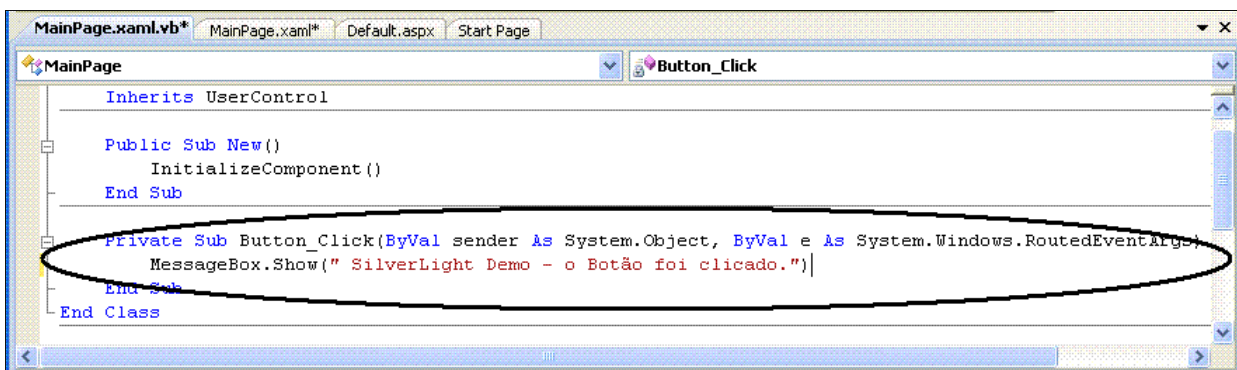
Vamos agora incluir código em nosso programa usando o editor XAML, digite a palavra **Click=**, e você verá o IntelliSense sugerir a criação de um manipulador de eventos (**Event Handler**) conforme a figura abaixo:



Clicando sobre **<New Event Handler>** será criado o manipulador com o nome **Button_Click**;



Podemos agora clicar com o botão direito do mouse e selecionar a opção **Navigate to Event Handler**. Isso nos levará a página de código do arquivo **MainPage.xaml.vb** e incluir o código conforme a figura abaixo:

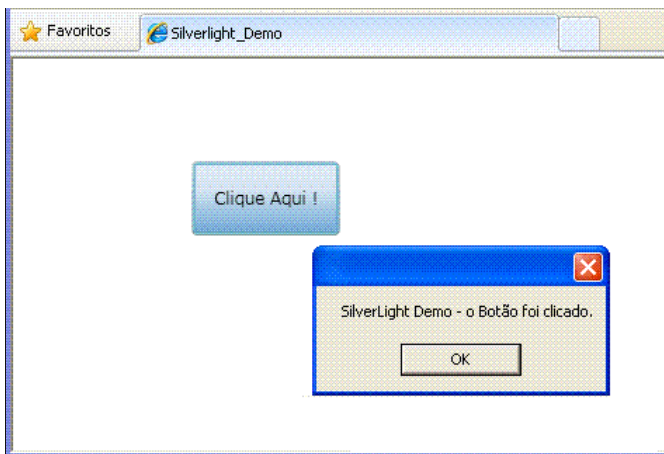


Vamos executar o projeto mas antes clique com o botão direito do mouse sobre o projeto **Silverlight_Demo.Web** e selecione a opção **Set as Startup Project**

A seguir clique com o botão direito sobre a página **Silverlight_DemoTestPage.aspx** e selecione a opção **Set as Startup Page**.

Executando o projeto iremos obter:





Pronto ! Acabamos de criar nossa primeira aplicação SilverLight.

Observe que estivemos trabalhando diretamente com a linguagem XAML uma linguagem de marcação. Mas eu não disse que com o SilverLight poderíamos criar interfaces com alta qualidade ? Como ?

Usando o Expression Blend

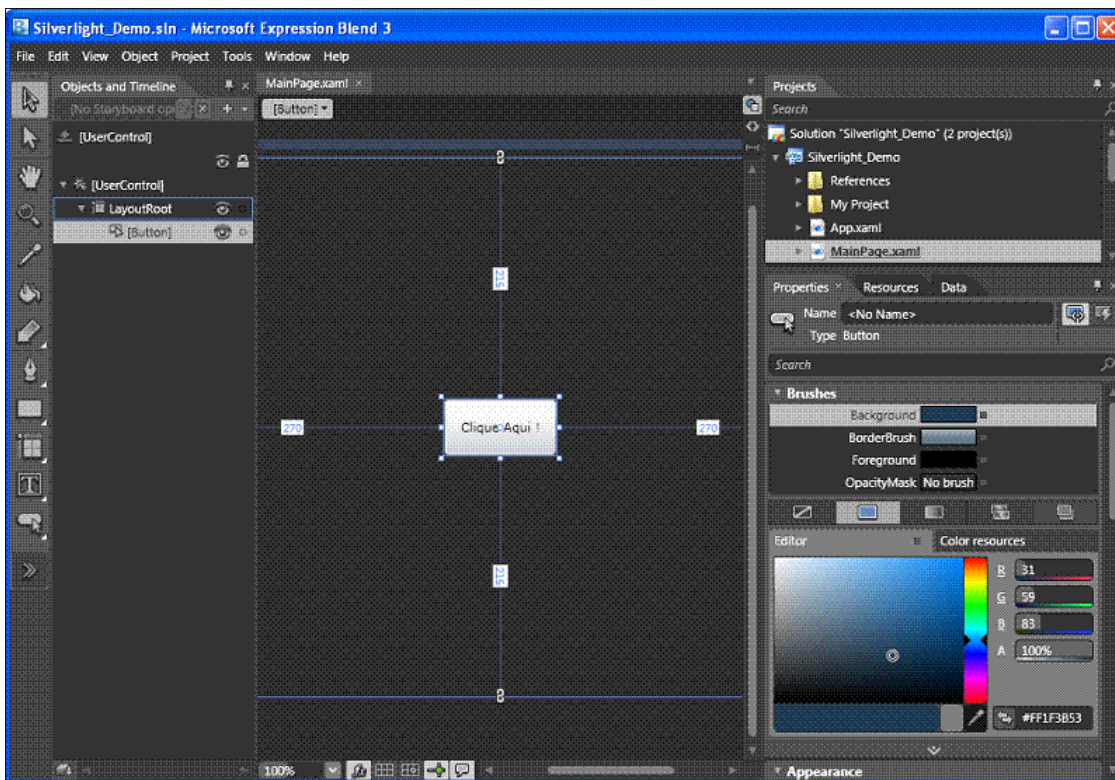
Usando um editor especializado chamado **Expression Blend**. Vamos então usá-lo.

Na janela Solution Explorer vamos clicar com o botão direito sobre o arquivo **MainPage.xaml** e selecionar a opção **Open in Expression Blend**.

O **Expression Blend** é uma ferramenta de design usada para desenhar interfaces **XAML** a partir da qual podemos criar as interfaces visuais de altíssima qualidade com recursos de animações e gráficos 3D nas aplicações SilverLight. Eu estou usando a versão trial : [Install Microsoft Expression Blend 3 + SketchFlow Trial](#)

Será apresentada uma janela de segurança, você deve clicar na opção **Yes** para abrir a página no Expression Blend.

Feito isso o **Expression Blend** será aberto exibindo a nossa página XAML com o controle Button usado conforme a figura abaixo:

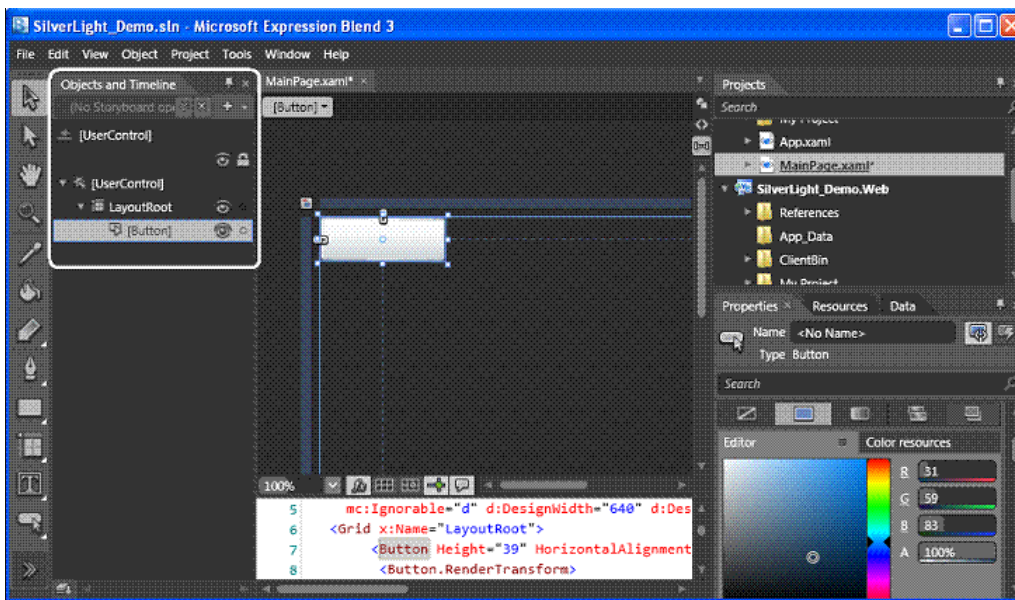


Embora a primeira vista possa parecer confusa a estrutura de janelas e menus é parecida com a do Visual Studio. Senão vejamos:

- **No lado esquerdo temos as ferramentas e os controles que são usados para criar a interface como na Toolbox;**
- **No lado direito temos a janela Project , igual a janela Solution Explorer;**
- **Abaixo temos a janela Properties que exhibe as propriedades do controle selecionado;**
- **Na base inferior temos a janela exibindo o código XAML da página;(Para alterar esta configuração no menu View -> Active Document View selecione o tipo de visualização com a qual deseja trabalhar.)**

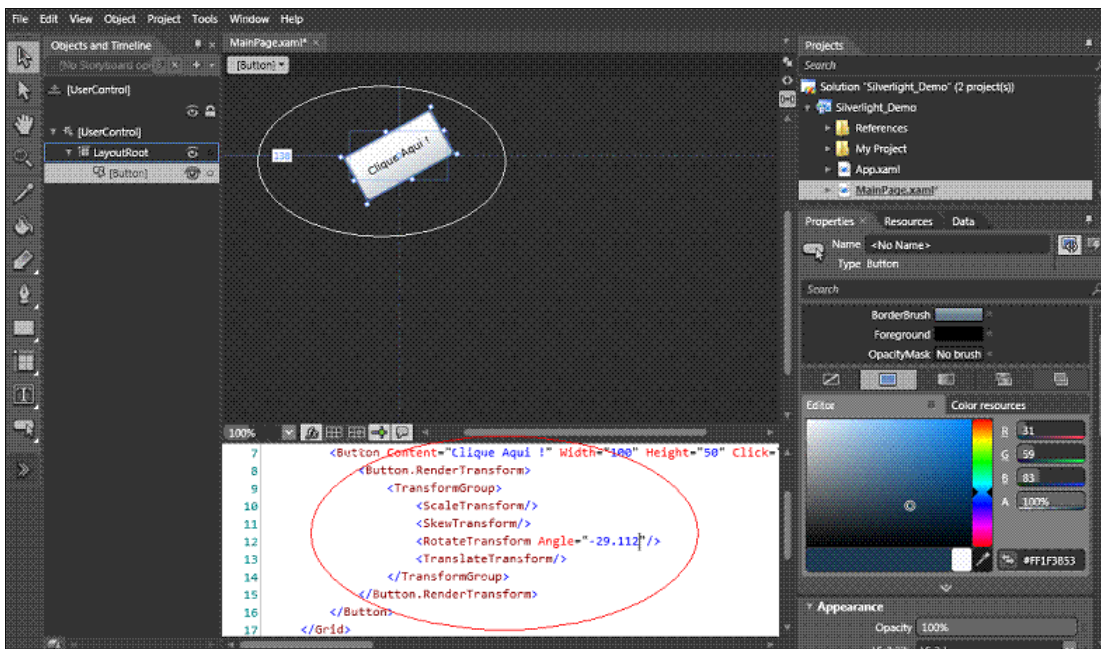
É claro que existem muito mais recursos para criação de interfaces no Expression Blend e por isso você vai encontrar outras janelas com ferramentas específicas para esta finalidade.

A janela **Objects and TimeLine** é muito útil para exibir uma visualização hierárquica dos controles na página, para ativá-la selecione o menu Windows a opção e marque-a, você verá a janela conforme a figura abaixo:



Qualquer alteração feita no projeto **Expression Blend** se reflete na solução do Visual Studio e vice-versa e qualquer alteração visual gera o código **XAML** correspondente.

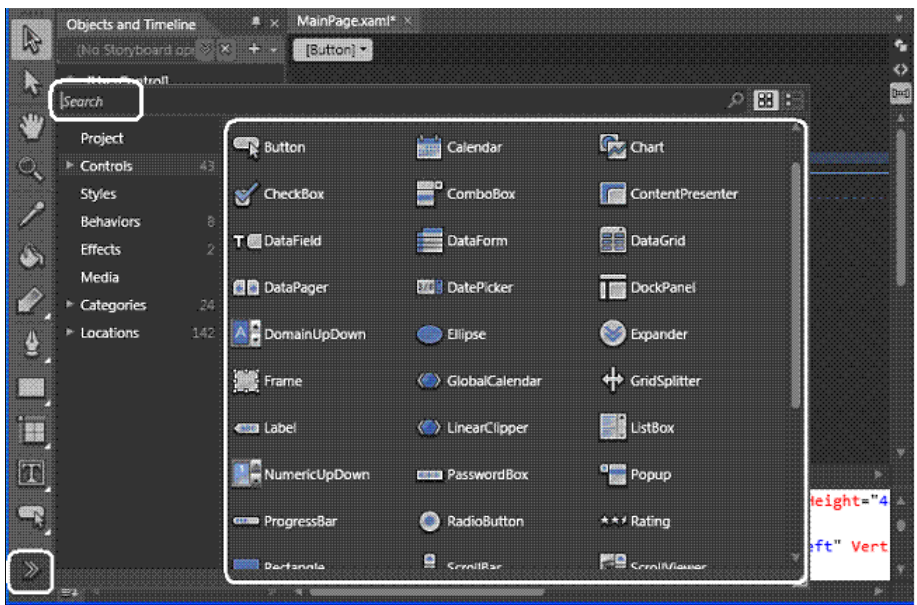
Veja na figura abaixo que aplicamos um efeito ao controle Button, fazendo a rotação do mesmo posicionando o mouse na parte inferior esquerda do Button e rodando para baixo, e vemos no descritor **XAML** o código gerado correspondente a esta ação;



Você vai perceber ao trabalhar com o Expression Blend que pode configurar o leiaute de exibição das janelas de diversas formas e que a ferramenta apresenta tantos recursos para a criação de interfaces ricas que **'ficamos um pouco perdidos'** em um primeiro contato. Nada que um pouco de estudo com paciência não resolva.

Selecione o ícone **Assets** na **ToolBox** para visualizar os controles disponíveis. Veja que temos uma caixa **Search** para ajudar a encontrar um controle tamanha a quantidade deles:





Para incluir um controle na página selecione o controle clicando duas vezes sobre o mesmo. Para excluir um controle, na janela **Objects and TimeLine**, clique com o botão direito sobre o controle e selecione a opção **Delete**.

Vou expor alguns conceitos básicos sobre a ferramenta Expression Blend mas recomendo quem deseja explorar a fundo a ferramenta a leitura de material especializado e livros sobre o Expression Blend.

Gerenciando o Layout no Expression Blend

Como você já observou as interfaces no **SilverLight** podem ser criadas usando diretamente a linguagem **XAML** digitando no local apropriado os comandos com a ajuda do IntelliSense ou usando o editor **Expression Blend** que é um ambiente integrado visual com diversas ferramentas e controles para alcançar o mesmo objetivo.

Em ambos os casos o tratamento da interface está baseado em **Panels** ou painéis que são usados para posicionar os controles na página e no SilverLight temos os seguintes **Panels**:

- **Canvas;**
- **StackPanel;**
- **Grid;**

O **Canvas** é o **Panel** mais simples e permite posicionar os controles em posições fixas através de coordenadas **Left e Top**;

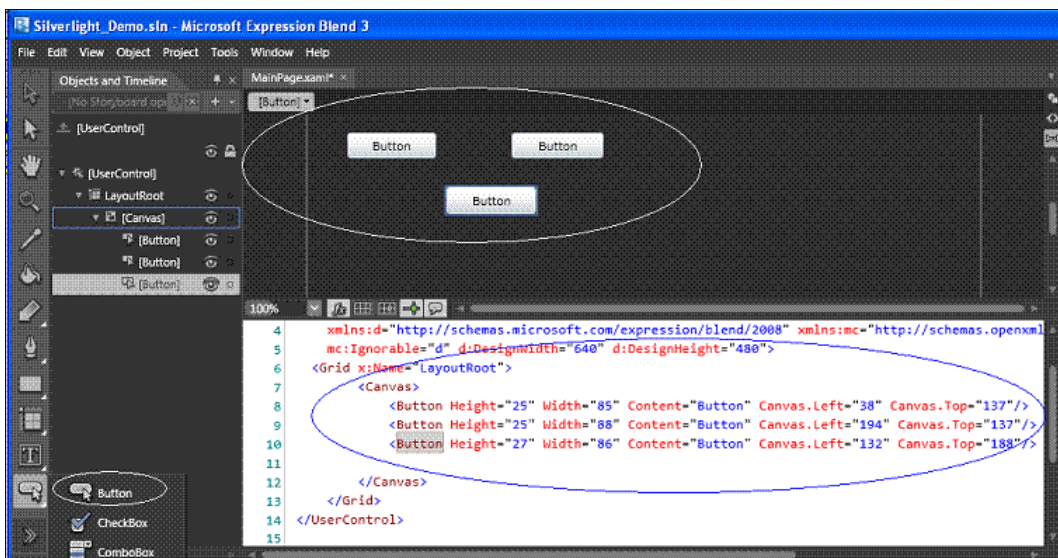
O Canvas, como outros **Panels** do SilverLight (*WPF também*), apresenta uma característica chamada **Attached Properties** que consiste em 'emprestar' propriedades aos controles contidos no seu interior. Veja abaixo que os controles **Button** tomam emprestados do Canvas as propriedades **Left e Top**;

Uma **propriedade anexada** é um conceito definido pela **Extensible Application Markup Language (XAML)**. Uma propriedade anexada destina-se a ser usada como um tipo de propriedade global que é configurável em qualquer objeto. No Windows Presentation Foundation (WPF), propriedades anexadas geralmente são definidas como uma forma especializada de propriedade de dependência que não possui o "invólucro" de propriedade convencional.

Uma das finalidades de uma **propriedade anexada** é permitir que diferentes elementos-filhos especifiquem valores únicos para uma propriedade que na verdade é definida em um elemento-pai. Uma aplicação específica desta situação é fazer com que elementos-filhos informem o elemento-pai sobre como elas devem ser apresentadas na interface do usuário (UI).

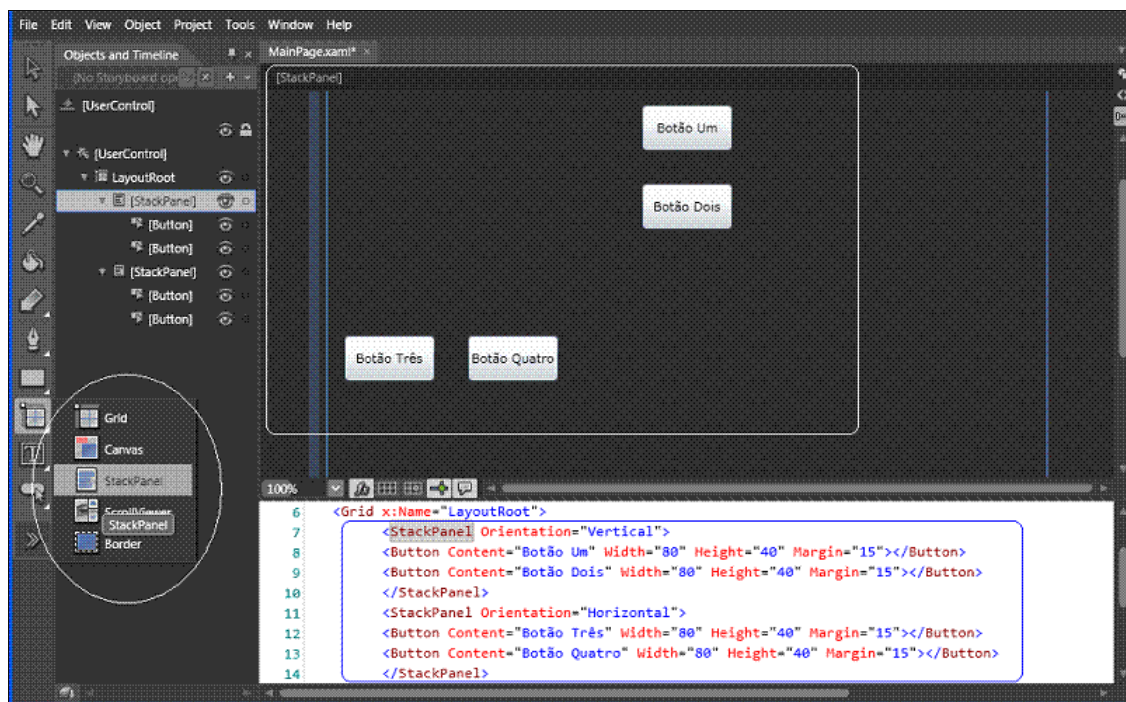
fonte: <http://msdn.microsoft.com/pt-br/library/ms749011.aspx>

O controle **Canvas** não é muito flexível e não se adapta às mudanças de tamanho da página.



O **StackPanel** é um Panel que permite o posicionamento de controles na horizontal ou vertical e que se adapta às mudanças da página;

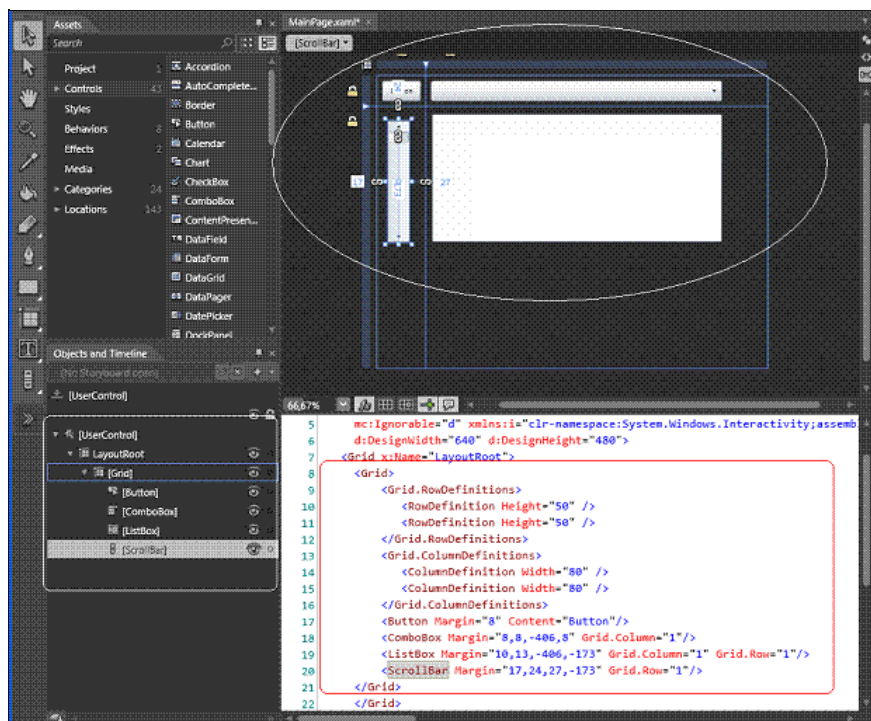
Na figura abaixo vemos dois **StackPannels** cada um contendo dois controles **Button**; Um tem orientação horizontal e o outro Vertical; Observe que estamos usando o atributo **Margin** que fornece um espaço de 15 pixels entre os controles;



O **Grid** é o Panel mais flexível e permite posicionar os controles em linhas e colunas como em uma tabela HTML usando as tags **<Grid.RowDefinitions>** e **<Grid.ColumnDefinitions>**;

O panel **Grid** também apresenta o recurso das **Attached Properties** e dessa forma 'empresta' as propriedades **Grid.Column** e **Grid.Row** para os elementos no seu interior;

No Exemplo abaixo Definimos um Grid com um **RowDefinitions** e um **ColumnDefinitions** e incluímos no seu interior os controles **Button**, **ComboBox**, **ListBox** e **ScrollBar**. Observe como estes elementos 'tomaram emprestados' as propriedades **Row** e **Column** do Grid;



O **Expression Blend** possui dezenas de outros recursos como **Estilos**, **Modelos**, **Efeitos**, **Behaviors**, **Data Binding**, etc. sobre os quais não estou entrando em detalhes devido ao espaço e ao tempo que exigiriam. Recomendo a quem se interessar pelo assunto fazer uma busca no Google ou pelas referências citadas se aprofundar nesta fantástica ferramenta que esta se aperfeiçoando cada vez mais.

Aguarde em breve mais exemplos práticos sobre o **SilverLight** e o **Expression Blend**...

Eu sei é apenas **SilverLight**, mas eu gosto...

Referências:

- <http://www.codeplex.com/Silverlight>

- [Silverlight 3 Toolkit July 2009 Installer](#)
- [Web Platform Installer](#)
- <http://silverlight.net/>
- <http://www.lynda.com/home/DisplayCourse.aspx?lpk2=384>
- http://www.microsoft.com/expression/products/Blend_Overview.aspx
- <http://blogs.silverlight.net/blogs/jesseliberty/default.aspx>
- <http://designwithsilverlight.com/>
- <http://silverzine.com/>
- [Apresentando o WPF - Windows Presentation Foundation](#)
- [WPF - DataBinding](#)
- [WPF - Criando um navegador Web](#)
- [WPF - Usando a vinculação de dados II](#)
- [WPF e LINQ - Modelo de Deployment](#)
- [WPF - Usando a vinculação de dados](#)
- [Introdução a XAML](#)