

Silverlight 4 Data and Services Cookbook

Aplicações de Negócios em Silverlight

Neste capítulo, vamos abordar os seguintes tópicos:

- ✓ Preparando o ambiente para desenvolver aplicações Silverlight.
- ✓ Criando o primeiro projeto silverlight.
- ✓ Utilizando o fluxo de trabalho entre o Visual Studio e o Expression Blend.
- ✓ Implantação de um aplicativo Silverlight no servidor.

Introdução

O Silverlight foi lançado no primeiro semestre de 2007, e desde então, trouxe com ele uma série de novidades. Enquanto o ASP.NET é uma plataforma de desenvolvimento do lado do servidor, com a chegada do Silverlight, o foco deslocou-se para o lado do cliente novamente. Uma aplicação Silverlight é executada no navegador do cliente e em uma versão específica do Common Language Runtime (CLR).

Um grande benefício para os desenvolvedores é que o Silverlight utiliza .NET a partir da versão 2 em diante. Tem uma versão reduzida da Base Class Library (BCL), que é impressionantemente estendida considerando o tamanho do plugin Silverlight (menos de 5 MB).

O Silverlight em si pode ser considerado como uma versão reduzida de sua contraparte desktop, o Windows Presentation Foundation (WPF). Entre Silverlight 4 e WPF 4, ainda existem algumas diferenças. Alguns recursos estão incluídos no WPF 4, mas não estão em Silverlight 4 e vice-versa.

Preparando o ambiente para o desenvolvimento Silverlight

Para começar a desenvolver aplicativos Silverlight, vamos precisar instalar as ferramentas necessárias e SDKs. Realize os seguintes passos, para começar:

1. Precisamos ter instalado o Visual Web Developer Express 2010 ou o Visual Studio 2010 versão para teste, disponível em :
<http://www.microsoft.com/visualstudio/en-us/download>
2. Vá para <http://www.silverlight.net/getstarted/> para baixar e instalar o Silverlight 4 Tools para Visual Studio 2010. (Visual Studio 2010 vem com o Silverlight 3 instalado).
3. Vá para <http://www.microsoft.com/expression/> para baixar e instalar Blend 4.

Como funciona ...

Para desenvolver em Silverlight, o mínimo que precisamos são das ferramentas de desenvolvimento. Ao instalar as ferramentas de desenvolvimento para Silverlight 4, os seguintes componentes são automaticamente baixados e instalados:

- Silverlight 4 developer runtime
- Silverlight 4 software development kit and Visual Studio project support
- WCF RIA services

Nós podemos escrever o código XAML usando o Visual Studio. No entanto, se você leva muito a sério a UI, você pode usar o Blend para fazer o acabamento final da Interface do usuário. Esta ferramenta é principalmente para os designers, para gerar uma interface rica, fácil de usar. Ela também se integra muito bem com o Visual Studio.

Criando nosso primeiro aplicativo Silverlight

Neste primeiro exemplo, nós vamos construir uma aplicação muito simples Silverlight. Nós estaremos usando ligação de dados (data binding), que é uma técnica para ligar facilmente os dados para a interface de usuário (UI) e se conectar a um Windows Communication Foundation (WCF).

No entanto, o principal objetivo é obter um controle sobre os fundamentos do Silverlight, tentando entender como um aplicativo do Silverlight é construído, qual é a estrutura do projeto, quais arquivos são criados, e qual é o seu uso.

Cenário:

Nossa primeira aplicação Silverlight permite ao usuário visualizar os detalhes de um hotel que está selecionado em um controle ComboBox. A informação do hotel é recuperado através de um serviço e é utilizado para preencher o ComboBox e os detalhes apresentados em vários controles TextBlock que são colocados em um Grid.

Para começar a construir qualquer aplicação Silverlight, você vai precisar realizar os seguintes passos:

1. Abra o Visual Studio 2010. Clique em **File | New | Project** na janela que surgir, selecione o nó Silverlight no Visual C # e selecione **Silverlight Application**. Digite para o nome do aplicativo **SilverlightHotelBrowser** e clique no botão OK. Na caixa de diálogo que aparece, selecione **ASP.NET Web Application Project** como o tipo de projeto web que será utilizado para hospedar o aplicativo Silverlight.
2. Dois projetos são criados na solução: o projeto Silverlight e um projeto de aplicação web que é responsável por hospedar o conteúdo Silverlight.
3. Nosso serviço irá retornar as informações de hotel. Um hotel pode ser representado por uma instância da classe Hotel. Esta classe deve ser incluída na aplicação Web "SilverlightHotelBrowser.Web".

Há algumas coisas a notar sobre esta classe:

- Esta classe tem um atributo **DataContract**. Este atributo é necessário para especificar que esta classe pode ser serializada quando enviada sobre a rede para a aplicação cliente.

- Cada propriedade é decorada com o atributo DataMember. Ao adicionar esse atributo para uma propriedade, especificamos que esta propriedade é parte do contrato e que deve ser incluída na resposta serializada que será enviado para o cliente.

O código a seguir define esta classe:

```
[DataContract]

public class Hotel
{
    [DataMember]
    public string Name { get; set; }

    [DataMember]
    public string Location { get; set; }

    [DataMember]
    public string Country { get; set; }

    [DataMember]
    public double Price { get; set; }
}
```

4. Vamos agora adicionar um serviço ao projeto Web. Clique com o botão direito do mouse em SilverlightHotelBrowser.web e selecione **Add | New | Item...** adicione um **Silverlight-enabled WCF Service** por selecionar o nó Silverlight e dê o nome de Hotel Service. Clique no botão **Add**. Dois arquivos serão adicionados ao projeto, HotelService.svc e HotelService.svc.cs.
 5. Nesta classe de serviço, agora podemos adicionar um método que retorna uma lista de hotéis. Remova o método DoWork e substitua-o pelo código a seguir:
-

```
[ServiceContract(Namespace = "")]

[AspNetCompatibilityRequirements(RequirementsMode =
AspNetCompatibilityRequirementsMode.Allowed)]

public class HotelService
{
    [OperationContract]
    public List<Hotel> GetHotels()
```

```
{  
    return new List<Hotel>  
    {  
        new Hotel  
        {  
            Name = "Brussels Hotel",  
            Price = 100,  
            Location = "Brussels",  
            Country = "Belgium"  
        },  
        new Hotel  
        {  
            Name = "London Hotel",  
            Price = 200,  
            Location = "London",  
            Country = "United Kingdom"  
        },  
        new Hotel  
        {  
            Name = "Paris Hotel",  
            Price = 150,  
            Location = "Paris",  
            Country = "France"  
        },  
        new Hotel  
        {  
            Name = "New York Hotel",
```

```
Price = 230,  
Location = "New York",  
Country = "USA"  
}  
};  
}  
}
```

Observe os atributos que foram usados nesta classe. O atributo `ServiceContract` especifica que a classe contém um contrato de serviço que define quais funcionalidades o serviço expõe. O atributo `OperationContract` é adicionado às operações que podem ser invocados pelos clientes no serviço. Isso efetivamente significa que se você adicionar métodos para o serviço sem este atributo, não pode ser chamado de um cliente.

6. Agora construa a solução, ou seja, dê um Build na solução para compilá-la. Se não tiver erro(s), então, vamos fazer com que a aplicação silverlight reconheça o serviço.

Botão direito do mouse sobre a aplicação Silverlight e selecione **Add Service Reference...** Na caixa de diálogo que aparece, como mostrado na imagem abaixo, clique em **Discover** e selecione o seu serviço. Como é na mesma solução, o serviço irá aparecer. Digite `HotelService` no campo `Namespace`.

Clique no botão **OK** para confirmar. Os serviços, agora, podem ser utilizáveis a partir da aplicação Silverlight.

7. A interface do usuário é bastante simples. O código XAML para o Grid chamado **LayoutRoot** dentro do arquivo `MainPage.xaml` é a seguinte:

```
<Grid x:Name="LayoutRoot" Width="400" Height="300" Background="LightGray">  
  <Grid.RowDefinitions>  
    <RowDefinition Height="50"></RowDefinition>  
    <RowDefinition></RowDefinition>  
  </Grid.RowDefinitions>  
  <ComboBox x:Name="HotelComboBox" Width="250"  
    SelectionChanged="HotelComboBox_SelectionChanged"  
    DisplayMemberPath="Name">
```

```

        VerticalAlignment="Center">
</ComboBox>
<Grid x:Name="HotelDetailGrid" Grid.Row="1"
    VerticalAlignment="Top">
    <Grid.RowDefinitions>
        <RowDefinition></RowDefinition>
        <RowDefinition></RowDefinition>
        <RowDefinition></RowDefinition>
        <RowDefinition></RowDefinition>
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition></ColumnDefinition>
        <ColumnDefiknition></ColumnDefinition>
    </Grid.ColumnDefinitions>
    <TextBlock x:Name="NameTextBlock" Grid.Row="0" Grid.Column="0"
        FontWeight="Bold" Text="Name: " HorizontalAlignment="Right">
</TextBlock>
    <TextBlock x:Name="NameValueTextBlock" Grid.Row="0"
        Grid.Column="1" Text="{Binding Name}">
</TextBlock>
    <TextBlock x:Name="LocationTextBlock" Grid.Row="1" Grid.Column="0"
        FontWeight="Bold" Text="Location: " HorizontalAlignment="Right">
</TextBlock>
    <TextBlock x:Name="LocationValueTextBlock" Grid.Row="1" Grid.Column="1"
        Text="{Binding Location}">
</TextBlock>
    <TextBlock x:Name="CountryTextBlock" Grid.Row="2" Grid.Column="0"

```

```

        FontWeight="Bold" Text="Country: " HorizontalAlignment="Right">
</TextBlock>
<TextBlock x:Name="CountryValueTextBlock" Grid.Row="2" Grid.Column="1"
        Text="{Binding Country}">
</TextBlock>
<TextBlock x:Name="PriceTextBlock" Grid.Row="3" Grid.Column="0"
        FontWeight="Bold" Text="Price: " HorizontalAlignment="Right">
</TextBlock>
<TextBlock x:Name="PriceValueTextBlock" Grid.Row="3" Grid.Column="1"
        Text="{Binding Price}">
</TextBlock>
</Grid>
</Grid>

```

8. No código behind (class MainPage.xaml.cs), vamos escrever o seguinte código:

```

public MainPage()
{
    InitializeComponent();
    HotelService.HotelServiceClient proxy = new
        SilverlightHotelBrowser.HotelService.HotelServiceClient();
    proxy.GetHotelsCompleted += new EventHandler
        <SilverlightHotelBrowser.HotelService.GetHotelsCompletedEventArgs>
        (proxy_GetHotelsCompleted);
    proxy.GetHotelsAsync();
}

void proxy_GetHotelsCompleted(object sender, SilverlightHotelBrowser.HotelService.
    GetHotelsCompletedEventArgs e)

```

```
{  
    HotelComboBox.ItemsSource = e.Result;  
}  
  
private void HotelComboBox_SelectionChanged(object sender, SelectionChangedEventArgs e)  
{  
    HotelDetailGrid.DataContext = (sender as ComboBox) .SelectedItem as HotelService.Hotel;  
}
```

9. Compile e rode a aplicação.

Aplicações Silverlight sempre tem que ser executado no contexto do navegador. Essa é a razão pela qual o Visual Studio inicialmente pergunta-nos como queremos hospedar o conteúdo Silverlight.

A opção padrão é ASP.NET Web Application Project. Esta opção nos dá o número máximo de possibilidades para a configuração do projecto no servidor (host). Esta opção é a mais usada quando estamos trabalhando com serviços, pois ela oferece mais opções de configuração. A segunda opção é ASP.NET Web Site e é um site baseado em arquivo conhecidos do ASP.NET 2.0.

Implantação de um aplicativo Silverlight no servidor

Uma vez que temos uma aplicação Silverlight pronta, vamos querer implantá-la.

O Silverlight é uma tecnologia .NET, que não exige .NET para ser instalado no servidor. Lembre-se que é uma tecnologia client-side. O plugin Silverlight no cliente fará o download e executa o aplicativo usando a versão do Common Language Runtime (CLR) incorporados no plug-in Silverlight. Neste laboratório, vamos ver como podemos implantar um aplicativo Silverlight.

Como fazer...

Implantação de um aplicativo Silverlight é fácil, o código do Silverlight é compilado e empacotado em um arquivo *.xap. Ficando este arquivo no lado do cliente e executá-lo de lá é a nossa única preocupação. Os seguintes passos devem ser realizados para a implantação de um aplicativo do Silverlight:

1. Observe que o Visual Studio criou um arquivo *.Xap no diretório ClientBin. Este arquivo, que nada mais é do que um arquivo *.zip, mas com outra

- extensão, contém todos os silverlight compilados de nossa aplicação e, os recursos opcionais, e também o arquivo AppManifest.xaml.
2. Se você dê uma olhada, você vai perceber que no projeto Visual Studio na web, foram criados dois arquivos de teste para rodar nossa aplicação: uma página HTML (HelloWorldTestPage.html) e uma página ASPX (HelloWorldTestPage.aspx):
 3. Ambas as páginas têm inclusa uma tag OBJECT. Um dos parâmetros é nomeado como Fonte (source) e ele tem uma referência para o arquivo *. xap na ClientBin como mostrado o seguinte código. Se quisermos implantar o arquivo *. xap para outro local, nós necessitamos atualizar esta referência. Vamos usar o padrão a partir de agora.
-

```
<object data="data:application/x-silverlight-2,"
        type="application/x-silverlight-2"
        width="100%"
        height="100%">
  <param name="source" value="ClientBin/DeployHelloWorld.xap"/>
  <param name="onError" value="onSilverlightError" />
  <param name="background" value="white" />
  <param name="minRuntimeVersion" value="4.0.41108.0" />
  <param name="autoUpgrade" value="true" />
  <a href="http://go.microsoft.com/fwlink/?LinkID=149156&v=4.0.4
    1108.0"
    style="text-decoration:none">
    
  </a>
</object>
```

Note-se que o valor de minRuntimeVersion podem diferir ligeiramente por causa de diferentes Releases Silverlight versão.

4. Se estiver usando a página HTML, os seguintes arquivos precisam ser copiados:
 - a. HelloWorldTestPage.html
 - b. Silverlight.js

- c. ClientBin/DeployHelloWorld.xap
5. Se estiver usando a página ASPX, precisamos copiar os seguintes arquivos:
 - a. HelloWorldTestPage.aspx
 - b. Silverlight.js
 - c. ClientBin/DeployHelloWorld.xap
 - d. bin directory, if using code-behind for the ASPX page
 - e. web.config
6. Precisamos testar a página em um navegador. Se ele não for carregado, verifique os tipos MIME servido pelo software de servidor web. Deve haver *.xap e *.xaml lá.(Eles são especificados como o tipo de dados na tag OBJECT.)

Uma das melhores coisas sobre o Silverlight é que ele pode rodar em qualquer tipo de servidor. Se estamos usando ASP.NET, PHP, JSP, ou o bom e velho HTML, Silverlight ainda pode ser incorporado. Silverlight é executado no lado do cliente. O plugin tem um CLR incorporado para que ele hospede a nossa aplicação. No lado do servidor, a única coisa que precisamos fazer é servir os arquivos (o mais importante, o arquivo *.xap) que serão transferidos para o lado do cliente, quando solicitado.

Alterações de configuração no servidor

Se o aplicativo Silverlight não está sendo mostrado, pode ser que o software de servidor (IIS ou Apache) não está configurado para servir os tipos de arquivo usado pelo Silverlight (*.xap e *.xaml). O Windows Vista SP1 e Windows Server veem com o Internet Information Services (IIS 7), enquanto o Windows 7 e Windows Server 2008 R2 inclui o IIS 7.5. Sobre estas versões do sistema operacional, o IIS 7 e IIS 7.5 são configurados para servir *.xap e arquivos *.xaml. No Windows Vista sem o SP1, é preciso adicionar os tipos conhecidos MIME. Nós podemos fazer isso abrindo o Internet Information Services (IIS) Manager e selecionando MIME Types. Então, simplesmente clique em Adicionar e adicione os dois seguintes itens:

- .xap na extensão de nome de arquivo: campo e application / x-silverlight-app no tipo MIME: campo
- .xaml na extensão de nome de arquivo: campo e application / xml + xaml no MIMETipo: campo

E se o servidor não permite o uso XAP?

Se o ambiente de servidor não permite adicionar tipos de MIME (um plano de hospedagem compartilhada), não há motivo para pânico. Como um arquivo *.xap é nada mais do que um arquivo *.zip, mas com outra extensão, o Silverlight suporta o arquivo *.xap sendo implantado como um arquivo *.zip.

Para as coisas funcionarem, mude o nome do arquivo *.xap na ClientBin para *.zip. Além disso, substitua a referência para o arquivo *.xap para o novo nome, como mostrado no código a seguir:

```
<object data="data:application/x-silverlight-2,"
```

type="application/x-silverlight-2"

width="100%"

height="100%">>

<param name="source" value="ClientBin/DeployHelloWorld.zip"/>

...

</object>
