

Silverlight 4



فهرست مطالب

فصل ۷ - آشنایی با سیستم Silverlight در Navigation	۱۳۵
مقدمه	۱۳۵
تعویض پویای محتوای قسمتی از صفحه	۱۳۵
تعویض پویای محتوای تمام صفحه	۱۳۸
ایجاد پنجره‌های جدید از نوع Modal	۱۴۰
استفاده از Navigation Framework	۱۴۶
مقابله با خطاهای مدیریت نشده‌ی سیستم Navigation	۱۴۹
استفاده از Silverlight Pages	۱۵۰
نگاشت آدرس‌ها در سیستم Navigation	۱۵۳
نحوه‌ی دسترسی به مقادیر Query Strings	۱۵۵
راهنمای کاربران به صفحاتی در خارج از برنامه	۱۵۵
قالب ایجاد پروژه‌هایی مبتنی بر سیستم Silverlight در Navigation	۱۵۶

چاپ عمومی غیر رایگان این مطالب بدون مجوز کتبی از طرف نویسنده به هر نحوی غیرمجاز است.
انتشار این مطالب بر روی اینترنت و یا استفاده از آن به صورت مستقیم و یا غیر مستقیم در نشریات الکترونیکی با ذکر مأخذ بلا مانع است.

فصل ۷ - آشنایی با سیستم Navigation در Silverlight

مقدمه

در کلیه مثال‌های فصل‌های قبلی کتاب جاری، تنها با یک صفحه سر و کار داشتیم. در عمل یک برنامه‌ی کاربردی از چندین صفحه تشکیل خواهد شد؛ بنابراین نیاز است تا بتوان به سادگی بین صفحات مختلف آن حرکت نمود. در این فصل با روش‌های مختلف راهبری و حرکت بین صفحات، آشنا خواهیم شد.

تعویض پویای محتوای قسمتی از صفحه

در طی یک مثال قصد داریم، محتوای قسمتی از صفحه را با محتوای دو User control جدید اضافه شده به پروژه، تعویض نمائیم. نمایی از فایل‌های این پروژه و همچنین برنامه‌ی در حال اجرا را در شکل ۱ می‌توان مشاهده نمود. برای این منظور یک پوشه‌ی جدید به نام Views به پروژه اضافه شده و سپس دو User control جدید به نام‌های Ellipse.xaml و Rectangle.xaml به این پوشه افزوده شده‌اند. در صفحه‌ی اصلی برنامه (MainPage.xaml)، دو دکمه به همراه یک Border بر روی صفحه قرار گرفته‌اند. قصد داریم با کلیک بر روی هر کدام از دکمه‌ها، محتوای Border را به صورت پویا با محتوای یکی از این User controls تعویض نمائیم. کدهای XAML این فایل‌ها در ادامه ارائه خواهند شد :

Ellipse.xaml

```
<UserControl x:Class="SilverlightApplication38.Views.Ellipse"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="
        http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignHeight="300" d:DesignWidth="400">
    <Grid x:Name="LayoutRoot" Background="White">
        <Ellipse Stroke="Black" Height="79"
            Width="79" >
            <Ellipse.Fill>
                <RadialGradientBrush>
```

```

        <GradientStop Color="Black" Offset="0" />
        <GradientStop Color="#FFE6BFB3" Offset="1" />
    </RadialGradientBrush>
</Ellipse.Fill>
</Ellipse>
</Grid>
</UserControl>

```

Rectangle.xaml

```

<UserControl x:Class="SilverlightApplication38.Views.Rectangle"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="
        http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignHeight="300" d:DesignWidth="400">
    <Grid x:Name="LayoutRoot" Background="White">
        <Rectangle Stroke="Black" Height="53" Width="51">
            <Rectangle.Fill>
                <RadialGradientBrush>
                    <GradientStop Color="Black" Offset="0" />
                    <GradientStop Color="#FFE8D8FF" Offset="1" />
                </RadialGradientBrush>
            </Rectangle.Fill>
        </Rectangle>
    </Grid>
</UserControl>

```

MainPage.xaml

```

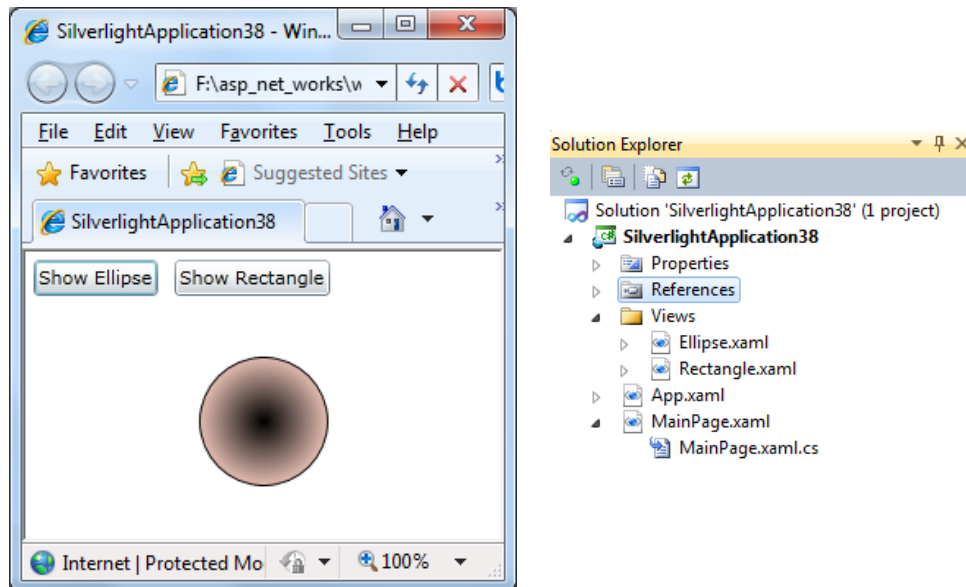
<UserControl x:Class="SilverlightApplication38.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="
        http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignHeight="300" d:DesignWidth="400">
    <Grid x:Name="LayoutRoot" Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="Auto" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>
        <StackPanel Orientation="Horizontal" Grid.Row="0">
            <Button Content="Show Ellipse" Name="btnEllipse"
                Margin="5" HorizontalAlignment="Left" Click="btnEllipse_Click" />
            <Button Content="Show Rectangle" Name="btnRectangle"

```

```

        HorizontalAlignment="Left" Margin="5" Click="btnRectangle_Click" />
    </StackPanel>
    <Border Name="border1" Grid.Row="1" />
</Grid>
</UserControl>

```



شکل ۱- تعویض پویای قسمتی از صفحه

اکنون می‌توان جهت تعویض محتوای Border با محتوای هر کدام از User Controls اضافه شده به برنامه، به صورت زیر عمل کرد:

MainPage.xaml.cs

```

using System.Windows;

namespace SilverlightApplication38
{
    public partial class MainPage
    {
        public MainPage()
        {
            InitializeComponent();

            private void btnEllipse_Click(object sender,
                RoutedEventArgs e)
            {
                border1.Child = new Views.Ellipse();
            }

            private void btnRectangle_Click(object sender,

```

```

        RoutedEventArgs e)
    {
        border1.Child = new Views.Rectangle();
    }
}

```

همانطور که ملاحظه می‌کنید وهله‌ای جدید از کلاس‌های هر کدام از User controls به خاصیت Child شیء Border انتساب داده شده است.

تعویض پویای محتوای تمام صفحه

در مثال قبل، تعویض پویای محتوای قسمتی از صفحه بررسی شد. اکنون در مثالی دیگر قصد داریم محتوای یک صفحه را به صورت کامل در زمان اجرا تعویض نمائیم. برای تعویض کامل محتوای صفحه‌ی آغازین برنامه باید در متد Application_Startup فایل استاندارد App.xaml.cs، وهله‌ای جدید از User control مورد نظر را به خاصیت RootVisual نسبت داد. هر چند این روش برای انتساب آغازین یک صفحه‌ی جدید به خوبی کار خواهد کرد، اما دیگر امکان تغییر مقدار RootVisual را نخواهیم داشت. به همین منظور یک شیء Border را در این فایل تعریف نموده و همانند کدهای ذیل بجای RootVisual از آن استفاده خواهیم کرد:

App.xaml.cs

```

private readonly Border _border = new Border();
private void Application_Startup(object sender, StartupEventArgs e)
{
    // do not assign a UserControl to RootVisual
    // instead, assign a Layout Panel

    _border.BorderBrush = new SolidColorBrush(Colors.Orange);
    _border.BorderThickness = new Thickness(3.0);
    this.RootVisual = _border;

    _border.Child = new MainPage();
}

// add a static method to App class to swap child
public static void ChangePage(UserControl page)
{
    var currentApp = (App)Application.Current;
    currentApp._border.Child = page;
}

```

اکنون سایر صفحات، تنها کافی است متد ChangePage فوق را جهت تعویض پویای کل محتوای صفحه‌ی خود فراخوانی نمایند؛ که مثالی از آن را در ادامه مشاهده خواهیم کرد. در این مثال یک User control جدید به نام NewPage.xaml به برنامه اضافه شده است. ابتدا توسط MainPage.xaml به این صفحه‌ی جدید هدایت خواهیم شد:

MainPage.xaml

```
<UserControl x:Class="SilverlightApplication39.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="
        http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignHeight="300" d:DesignWidth="400">
    <StackPanel x:Name="LayoutRoot"
        Background='LightYellow' Margin='20'>
        <TextBlock FontSize='14' Text='Main Page' />
        <Button Content='Go To Other Page'
            Click='Button_Click' Margin='20' />
    </StackPanel>
</UserControl>
```

کدهای متناظر با MainPage.xaml جهت انتقال به صفحه‌ی جدید به صورت پویا:

MainPage.xaml.cs

```
using System.Windows;

namespace SilverlightApplication39
{
    public partial class MainPage
    {
        public MainPage()
        {
            InitializeComponent();
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            App.ChangePage(new NewPage());
        }
    }
}
```

سپس توسط NewPage.xaml نیز قصد داریم به صفحه‌ی MainPage.xaml بازگشت نماییم:

NewPage.xaml

```
<UserControl x:Class="SilverlightApplication39.NewPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="
        http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignHeight="300" d:DesignWidth="400">
    <StackPanel x:Name="LayoutRoot" Background='LightBlue'>
        <TextBlock FontSize='14' Text='Second Page' Margin='20' />
        <Button Content='Back To Main'
            Click='Button_Click' Margin='20' />
        <TextBlock Text='Lorem Ipsum...' Margin='20' />
    </StackPanel>
</UserControl>
```

کدهای این صفحه جهت تعویض پویای تمام محتوای آن به شرح زیر می باشند :

NewPage.xaml.cs

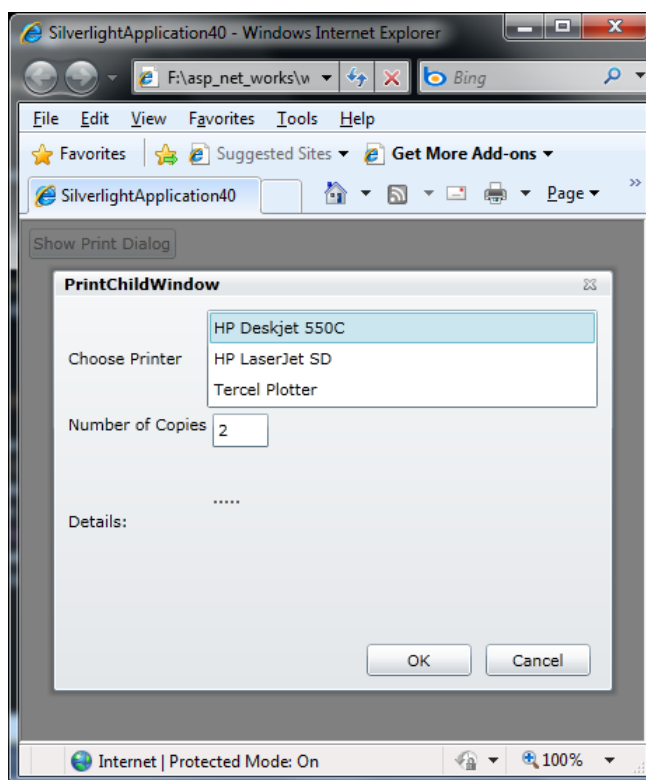
```
using System.Windows;
namespace SilverlightApplication39
{
    public partial class NewPage
    {
        public NewPage()
        {
            InitializeComponent();
        }

        private void Button_Click(object sender, RoutedEventArgs e)
        {
            App.ChangePage(new MainPage());
        }
    }
}
```

ایجاد پنجره‌های جدید از نوع Modal

یکی از انواع فایل‌هایی که می‌توان به یک پروژه‌ی Silverlight افزود، Silverlight Child Windows است که از آن جهت نمایش صفحات جدید Modal استفاده می‌شود (همانند برنامه‌های Desktop). یک پروژه‌ی جدید را آغاز نموده و از طریق منوی پروژه، گزینه‌ی Add new item، یک Silverlight Child Window جدید را به نام PrintChildWindow.xaml به برنامه اضافه نمایید. پس از افزودن آن اگر به کدهای XAML تولید شده

دقت نمائیم، المان ریشه در اینجا `controls:ChildWindow` نام گرفته است و این صفحه دارای دو دکمه‌ی OK و Cancel پیش فرض می‌باشد که کاملاً قابلیت سفارشی شدن را نیز دارا هستند. در ادامه قصد داریم نحوه‌ی نمایش این صفحه‌ی Modal جدید را به همراه نحوه‌ی دریافت اطلاعات وارد شده در آن، در صفحه‌ی اصلی فراخوان برنامه بررسی نمائیم (شکل ۲).



شکل ۲- نمایشی از برنامه‌ی استفاده از یک Silverlight Child Window.

کدهای XAML صفحه جدید برنامه به شرح بعد می‌باشند. در این کدها امکان تعریف یک چاپگر از پیش معین شده و همچنین تعداد صفحات مورد نیاز، مهیا می‌گردد:

PrintChildWindow.xaml

```
<controls:ChildWindow
  x:Class="SilverlightApplication40.PrintChildWindow"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:controls=
    "clr-namespace:System.Windows.Controls;assembly=System.Windows.Controls"
  Width="400" Height="300" Title="PrintChildWindow">
  <Grid x:Name="LayoutRoot" Margin="2">
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="Auto" />
      <ColumnDefinition />
    </Grid.ColumnDefinitions>
```

```

<Grid.RowDefinitions>
    <RowDefinition Height="Auto" />
    <RowDefinition Height="Auto" MinHeight="52" />
    <RowDefinition />
    <RowDefinition Height="Auto" MinHeight="58" />
    <RowDefinition Height="Auto" MinHeight="81" />
    <RowDefinition Height="Auto" />
</Grid.RowDefinitions>
<TextBlock HorizontalAlignment="Left"
VerticalAlignment="Center"
            Text="Choose Printer"
            Grid.Row="0" Margin="0,4" />
<ListBox x:Name='listPrinters'
            Grid.Column="1">
    <TextBlock Text='HP Deskjet 550C' />
    <TextBlock Text='HP LaserJet SD' />
    <TextBlock Text='Tercel Plotter' />
</ListBox>

<TextBlock HorizontalAlignment="Left"
VerticalAlignment="Top"
            Text="Number of Copies"
            Grid.Row="1"
            Margin="0,4,0,0" />
<TextBox x:Name="numberCopiesTextbox"
            Grid.Row="1"
            Grid.Column="1"
            Margin="4,4,4,0"
            Grid.RowSpan="1"
            Text='2'
            MinWidth='40'
            VerticalAlignment="Top"
            HorizontalAlignment="Left" />

<TextBlock HorizontalAlignment="Left"
VerticalAlignment="Center"
            Text="Details:"
            Grid.Row="3"
            Margin="0,4" />
<TextBlock x:Name="selectionTextbox"
            Grid.Row="3"
            Grid.Column="1"
            Margin="4"
            Text="....."
            TextWrapping="Wrap" />

<StackPanel Orientation='Horizontal'

```

```

        Grid.Row="4"

        Grid.Column="1"
            HorizontalAlignment="Right">
        <Button x:Name="OKButton"
            Content="OK"
            Click="OKButton_Click"
            Width="75"    Height="23"
            HorizontalAlignment="Left"
            Margin="5,5,5,0"
        VerticalAlignment="Bottom" />
        <Button x:Name="CancelButton"
            Content="Cancel"
            Click="CancelButton_Click"
            VerticalAlignment="Bottom"
            Width="75"    Height="23"
            Margin="5,5,5,0" />

    </StackPanel>
</Grid>
</controls:ChildWindow>

```

کدهای متناظر با این صفحه در ادامه ذکر شده‌اند. نکته‌ی مهم این کدها تعریف دو خاصیت عمومی جهت دسترسی به مقادیر وارد شده در صفحه، توسط صفحات دیگر برنامه است:

PrintChildWindow.xaml.cs

```

using System.Windows;
using System.Windows.Controls;

namespace SilverlightApplication40
{
    public partial class PrintChildWindow
    {
        public PrintChildWindow()
        {
            InitializeComponent();
        }

        private void OKButton_Click(object sender, RoutedEventArgs e)
        {
            this.DialogResult = true;
        }

        private void CancelButton_Click(object sender, RoutedEventArgs e)
        {
            this.DialogResult = false;
        }

        // create properties to expose values to other pages
        public int NumberOfCopies

```

```

    {
        get
        {
            return int.Parse(numberCopiesTextbox.Text);
        }
    }
    public string PrinterName
    {
        get
        {
            var textBlock = listPrinters.SelectedItem as TextBlock;
            return textBlock != null ? textBlock.Text : string.Empty;
        }
    }
}
}

```

کدهای XAML صفحه‌ی اصلی برنامه که کار نمایش صفحه‌ی جدید را به عهده خواهد داشت به شرح زیر می‌باشند. در این کدها یک دکمه جهت نمایش صفحه و یک برچسب جهت نمایش مقادیر دریافت شده از صفحه‌ی جدید، در نظر گرفته شده اند:

MainPage.xaml

```

<UserControl x:Class="SilverlightApplication40.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="
        http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignHeight="300" d:DesignWidth="400">
    <Grid x:Name="LayoutRoot" Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="auto" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>
        <Button Content="Show Print Dialog" HorizontalAlignment="Left"
            Grid.Row="0"
            Margin="5" Name="btnShow" Click="btnShow_Click" />
        <TextBlock Grid.Row="1" Name="txtResult" />
    </Grid>
</UserControl>

```

سپس کدهای MainPage.xaml.cs را ملاحظه می‌نمائید. ابتدا یک وهله‌ی جدید از شیء PrintChildWindow ایجاد شده است. سپس مایل هستیم تا در زمان بسته شدن صفحه، در صورتیکه بر روی دکمه‌ی OK کلیک شده

بود، مقادیر دریافتی را در یک برچسب نمایش دهیم. به همین جهت رخداد Closed این شیء جدید مورد بررسی قرار می‌گیرد. در نهایت برای نمایش این صفحه، متد Show باید فراخوانی گردد:

MainPage.xaml.cs

```
using System;
using System.Windows;

namespace SilverlightApplication40
{
    public partial class MainPage
    {
        public MainPage()
        {
            InitializeComponent();

            private void btnShow_Click(object sender, RoutedEventArgs e)
            {
                var dialog = new PrintChildWindow();
                dialog.Closed += dialog_Closed;
                dialog.Show();
            }

            void dialog_Closed(object sender, EventArgs e)
            {
                var dialog = sender as PrintChildWindow;
                if (dialog == null) return;
                if (dialog.DialogResult == true) //if OK ...
                    txtResult.Text = string.Format(
                        "NumberOfCopies: {0}, PrinterName: {1}",
                        dialog.NumberOfCopies,
                        dialog.PrinterName);
            }
        }
    }
}
```

نحوه‌ی دریافت و پردازش اطلاعات صفحه‌ی جدید را در متد dialog_Closed می‌توان ملاحظه نمود. Sender در اینجا همان شیء PrintChildWindow است. پس از انجام تبدیلات لازم، از طریق همین شیء می‌توان به خواص DialogResult و یا سایر خواص سفارشی تعریف شده، دسترسی داشت.

استفاده از Navigation Framework

هرچند برنامه‌های Silverlight در مرورگر وب اجرا می‌شوند اما همانند صفحات وب امکان به خاطر سپاری آدرس صفحه و موقعیت جاری در سایت وجود ندارد. برای رفع این مشکلات، Navigation Framework معرفی شده است که توسط اسمبلی System.Windows.Controls.Navigation.dll در اختیار برنامه‌ی شما قرار می‌گیرد. دو مؤلفه‌ی مهم آن به شرح زیر هستند:

- **Frame**: حرکت به سمت جلو و عقب را در مرورگر پشتیبانی می‌کند (ثبت شدن در Browser History journal). همچنین امکان استفاده از URI را جهت بارگذاری صفحات مختلف برنامه می‌سازد. هر Frame جهت دربرگیری اشیاء Page ارائه شده است.
- **Page**: جهت استفاده‌ی بهینه از این سیستم، بهتر است از Silverlight Pages بجای User controls استفاده گردد. زیرا صفحات نیز دسترسی به سیستم Navigation و همچنین تاریخچه‌ی فعالیت‌های مرورگر را دارا می‌باشند. بعلاوه امکان کار با پارامترهای دریافت شده از یک URI را نیز دارند. یکی از نکات جالب آن، خاصیت Title صفحات است که به صورت عنوان صفحه‌ی جاری در مرورگر اعمال می‌گردد.

در ادامه در طی یک مثال کاربردی، با جزئیات توانایی‌های Navigation Framework بیشتر آشنا خواهیم شد. یک پروژه‌ی جدید را آغاز نمائید. سپس پوشه‌ای را به نام UserControls به آن افزوده و User control جدیدی را به نام BlueUserControl.xaml به آن اضافه نمائید. توسط المان Frame قصد داریم با دو روش متفاوت این User control را در صفحه‌ی اصلی برنامه نمایش دهیم. محتوای BlueUserControl.xaml نکته‌ی خاصی نداشته و صرفاً جهت ارائه‌ی یک محتوای ساده، تعریف شده است:

BlueUserControl.xaml

```
<UserControl
  x:Class="SilverlightApplication41.UserControls.BlueUserControl"
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
  xmlns:mc="
    http://schemas.openxmlformats.org/markup-compatibility/2006"
  mc:Ignorable="d" d:DesignHeight="300" d:DesignWidth="400">
  <StackPanel x:Name="LayoutRoot"
    Background="#FF9FD6D8">
    <TextBlock Text='Blue User Control'
      HorizontalAlignment='Center'
      Margin='20' />
    <TextBlock
      Text='Same page, loaded with two different mechanisms'
```

```

        HorizontalAlignment='Center'
        Margin='20' />

        <TextBlock Text='' x:Name='resultsTextBlock'
            HorizontalAlignment='Center'
            Margin='20' />
    </StackPanel>
</UserControl>

```

در صفحه‌ی اصلی برنامه که کدهای XAML آنرا در ادامه ملاحظه می‌نمائید، دو دکمه جهت نمایش محتویات BlueUserControl.xaml در Frame قرار گرفته بر روی فرم، اضافه شده‌اند (کنترل Frame را از نوار ابزار کشیده و بر روی صفحه رها نمائید تا VS.NET ارجاعات لازم و فضای نام مربوطه را به صورت خودکار اضافه نماید):

MainPage.xaml

```

<UserControl x:Class="SilverlightApplication41.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="
        http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignHeight="300" d:DesignWidth="400"
    xmlns:sdk="http://schemas.microsoft.com/winfx/2006/xaml/presentation/sdk">
    <Grid x:Name="LayoutRoot" Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="auto" />
            <RowDefinition Height="*" />
        </Grid.RowDefinitions>
        <sdk:Frame Height="auto" Width="auto"
            HorizontalAlignment="Left" Margin="5"
            Name="frame1"
            VerticalAlignment="Top" Grid.Row="1" />
        <StackPanel Orientation="Horizontal" Grid.Row="0">
            <Button Content="Blue (Loaded as Content)"
                HorizontalAlignment="Left" Margin="5"
                Name="btnContent" Click="btnContent_Click" />
            <Button Content="Blue (Use Navigation)"
                HorizontalAlignment="Left" Margin="5"
                Name="btnNavigation" Click="btnNavigation_Click" />
        </StackPanel>
    </Grid>
</UserControl>

```

کدهای نمایش این User control با دو روش متفاوت به شرح بعد هستند :

MainPage.xaml.cs

```

using System;
using System.Windows;
using SilverlightApplication41.UserControls;

namespace SilverlightApplication41
{
    public partial class MainPage
    {
        public MainPage()
        {
            InitializeComponent();

            private void btnContent_Click(object sender, RoutedEventArgs e)
            {
                frame1.Content = new BlueUserControl();
            }

            private void btnNavigation_Click(object sender,
                RoutedEventArgs e)
            {
                frame1.Navigate(
                    new Uri("/UserControls/BlueUserControl.xaml",
                        UriKind.Relative));
            }
        }
    }
}

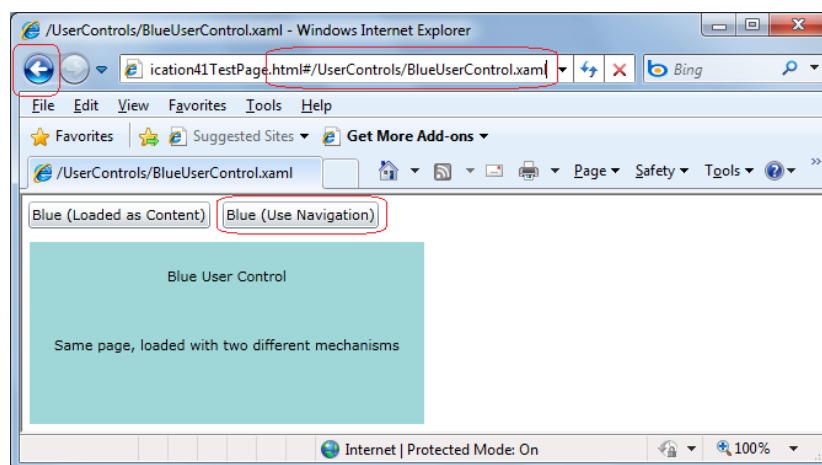
```

یا می‌توان وهله‌ای از شیء BlueUserControl را به خاصیت Content شیء Frame انتساب داد که در این صورت از خواص سیستم Navigation محروم خواهیم شد، یا می‌توان از متد Navigate جهت رهنمون شدن به User control مورد نظر سود جست (شکل ۳).

اگر به شکل ۳ دقت نمائید، دو نکته را در حین استفاده از سیستم Navigation می‌توان دریافت:

۱. فعال شدن دکمه‌های Back و Forward در مرورگر، همانند زمانیکه با صفحات HTML کار می‌کنیم (Navigation history).

۲. آدرس User control مورد استفاده در انتهای URL صفحه اضافه شده است. اگر صفحه را بسته و مجدداً این آدرس را در مرورگر وارد نمائیم دقیقاً به همین صفحه‌ی خاص، به همراه محتوای User control مورد نظر هدایت خواهیم شد.



شکل ۳- نمایشی از نمایش یک User control به دو روش در Silverlight.

نکته‌ی مهمی را که حین کار با سیستم Navigation در Silverlight باید در نظر داشت این است که به ازای هر بار فراخوانی یک صفحه در این حالت، یک وهله‌ی جدید از آن ساخته خواهد شد و وهله‌ی قبلی دیگر در دسترس نخواهد بود.

مقابله با خطاهای مدیریت نشده‌ی سیستم Navigation

در مثال قبل اگر ابتدا بر روی دکمه‌ی بارگذاری محتوا و سپس بر روی دکمه‌ی استفاده از سیستم Navigation کلیک نمائیم، دکمه‌ی بازگشت به قبل مرورگر فعال می‌گردد. اگر در این حالت بر روی این دکمه کلیک شود، با یک خطا و پایان کار برنامه مواجه می‌شویم؛ زیرا صفحه‌ی اولیه مورد استفاده از روش سیستم Navigation برای بارگذاری و نمایش استفاده نکرده است. برای رفع این مشکل باید از رخداد NavigationFailed کمک گرفت. روش معرفی کردن روال رویدادگران مرتبط در فایل XAML برنامه به صورت زیر است:

MainPage.xaml

```
<sdk:Frame Height="auto" Width="auto"
    HorizontalAlignment="Left" Margin="5"
    Name="frame1"
    VerticalAlignment="Top" Grid.Row="1"
    NavigationFailed="frame1_NavigationFailed" />
```

سپس برای صرفنظر کردن از این خطای رخ داده، روال رویدادگردان آن به شرح بعد خواهد بود:

MainPage.xaml.cs

```
private void frame1_NavigationFailed(object sender,
    System.Windows.Navigation.NavigationFailedEventArgs e)
{ e.Handled = true; }
```

استفاده از Silverlight Pages

برای تکمیل برنامه و استفاده از قابلیت‌های صفحات در Silverlight، یک پوشه‌ی جدید را به نام Pages به پروژه‌ی قبلی اضافه نمائید و سپس دو صفحه‌ی جدید از نوع Silverlight Page را به نام‌های Page1.xaml و Page2.xaml به این پوشه اضافه کنید. کدهای XAML این دو صفحه و همچنین نحوه‌ی استفاده از سیستم Navigation برای حرکت بین این صفحات در ادامه ذکر خواهند گردید. ابتدا به فایل MainPage.xaml مراجعه کرده و خاصیت Source شیء Frame را به صورت زیر مقدار دهی نمائید. به این ترتیب صفحه‌ی آغازین Frame مشخص خواهد شد :

MainPage.xaml

```
<sdk:Frame Height="auto" Width="auto"
    HorizontalAlignment="Left" Margin="5"
    Name="frame1"
    VerticalAlignment="Top" Grid.Row="1"
    Source="/Pages/page1.xaml"
    BorderThickness="4"
    BorderBrush="#FF48A41F"
    NavigationFailed="frame1_NavigationFailed" />
```

سپس کدهای XAML صفحات برنامه به شرح زیر می‌باشند:

Page1.xaml

```
<navigation:Page
x:Class="SilverlightApplication41.Pages.Page1"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="
        http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignWidth="640" d:DesignHeight="480"
    xmlns:navigation=
        "clr-namespace:System.Windows.Controls;assembly=System.Windows.Controls.Navigation"
    Title="Page1">
    <Grid x:Name="LayoutRoot">
        <StackPanel Background="White">
            <TextBlock FontSize="18.667"
                Text="Page1" TextWrapping="Wrap" />
            <Button Margin="5"
                Content="Go to Page2"
                x:Name='btnGotoPage2'
                Click="btnGotoPage2_Click" />
        </StackPanel>
    </Grid>
</navigation:Page>
```

نحوه‌ی استفاده از NavigationService مهیا، به صورت زیر است:

Page1.xaml.cs

```
using System;
using System.Windows;
using System.Windows.Navigation;

namespace SilverlightApplication41.Pages
{
    public partial class Page1
    {
        public Page1()
        {
            InitializeComponent();

            // Executes when the user navigates to this page.
            protected override void OnNavigatedTo(NavigationEventArgs e)
            {
            }

            private void btnGotoPage2_Click(object sender,
                RoutedEventArgs e)
            {
                this.NavigationService.Navigate(
                    new Uri("/Pages/Page2.xaml", UriKind.Relative)
                );
            }
        }
    }
}
```

کدهای XAML صفحه‌ی دو به شرح بعد می‌باشند:

Page2.xaml

```
<navigation:Page
    x:Class="SilverlightApplication41.Pages.Page2"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="
        http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d" d:DesignWidth="640" d:DesignHeight="480"
    xmlns:navigation=
        "clr-namespace:System.Windows.Controls;assembly=System.Windows.Controls.Navigation"
    Title="Page2">
    <Grid x:Name="LayoutRoot">
        <StackPanel Background="White">
```

```
<TextBlock FontSize="18.667"
    Text="Page2" TextWrapping="Wrap" />
<Button Margin="5"
    Content="Go to Page1"
    x:Name='btnGotoPage1'
    Click="btnGotoPage1_Click" />
</StackPanel>
</Grid>
</navigation:Page>
```

و از کدهای زیر برای هدایت به صفحه‌ی یک می‌توان کمک گرفت:

Page2.xaml.cs

```
using System;
using System.Windows;
using System.Windows.Navigation;

namespace SilverlightApplication41.Pages
{
    public partial class Page2
    {
        public Page2()
        {
            InitializeComponent();
        }

        // Executes when the user navigates to this page.
        protected override void OnNavigatedTo(NavigationEventArgs e)
        {
        }

        private void btnGotoPage1_Click(object sender,
            RoutedEventArgs e)
        {
            this.NavigationService.Navigate(
                new Uri("/Pages/Page1.xaml", UriKind.Relative)
            );
        }
    }
}
```

نگاشت آدرس‌ها در سیستم Navigation

اگر مثال دوم سیستم Navigation را اجرا نمائید، به آدرس‌هایی همانند آدرس ذیل برخورد خواهید خورد:
SilverlightApplication41TestPage.html#/Pages/Page1.xaml

برای کوتاه سازی این آدرس‌ها و همچنین ارائه‌ی زیباتر آن‌ها به کاربران، امکان نگاشت این آدرس‌ها به عباراتی کوتاه‌تر نیز وجود دارد. جهت تعریف این نگاشت‌ها ابتدا قسمت منابع user control اصلی که شیء Frame در آن قرار گرفته است را ایجاد کرده و سپس تعاریف UriMapper و UriMapping متناظر را به آن خواهیم افزود:

MainPage.xaml

```
<UserControl.Resources>
  <sdk:UriMapper x:Key="uriMapper1">
    <sdk:UriMapping Uri="/Page1"
      MappedUri="/Pages/Page1.xaml" />
    <sdk:UriMapping Uri="/Page2"
      MappedUri="/Pages/Page2.xaml" />
  </sdk:UriMapper>
</UserControl.Resources>
```

در اینجا می‌توان به ازای هر آدرس، یک نگاشت سفارشی تعریف نمود. اکنون برای معرفی این منبع تعریف شده باید به صورت زیر عمل نمود :

MainPage.xaml

```
<sdk:Frame Height="auto" Width="auto"
  HorizontalAlignment="Left" Margin="5"
  Name="frame1"
  VerticalAlignment="Top" Grid.Row="1"
  Source="/Page1"
  BorderThickness="4"
  UriMapper="{StaticResource uriMapper1}"
  BorderBrush="#FF48A41F"
  NavigationFailed="frame1_NavigationFailed" />
```

در خاصیت UriMapper، منبع ثابت تعریف شده معرفی گردیده است. سپس خاصیت Source نیز بر اساس این نگاشت‌ها تغییر کرده است. زیرا اکنون سیستم Navigation دقیقاً می‌داند که عبارت وارد شده معادل کدام صفحه در سیستم است.

سپس می‌توان کدهای برنامه را نیز جهت استفاده‌ی بهینه از نگاشت‌های تعریف شده به صورت بعد در دو فایل کدهای صفحات برنامه تغییر داد:

Page1.xaml.cs

```
private void btnGotoPage2_Click(object sender, RoutedEventArgs e)
{
    this.NavigationService.Navigate(
        new Uri("/Page2", UriKind.Relative)
    );
}
```

Page2.xaml.cs

```
private void btnGotoPage1_Click(object sender, RoutedEventArgs e)
{
    this.NavigationService.Navigate(
        new Uri("/Page1", UriKind.Relative)
    );
}
```

مثال‌های بیشتری در مورد سیستم نگاشت آدرس‌ها و توانایی‌های آن را در جداول زیر می‌توان ملاحظه نمود.
مثالی ابتدایی در مورد نحوه‌ی تعریف و استفاده از سیستم نگاشت‌ها:

تعریف نگاشت	Uri = "/Home" MappedUri = "/Views/Home.xaml"
نمونه آدرس نهایی نمایش داده شده	/Home
نگاشت نهایی در برنامه	/Views/Home.xaml

در این سیستم امکان تعریف متغیرها هم با استفاده از {} وجود دارد:

تعریف نگاشت	Uri =("/{page}" MappedUri = "/Views/{page}Page.xaml"
نمونه آدرس نهایی نمایش داده شده	/About
نگاشت نهایی در برنامه	/Views/AboutPage.xaml

همانند برنامه‌های ASP.NET، امکان تعریف Query String نیز مهیا است:

تعریف نگاشت	Uri = "/Product/{category}" MappedUri = "/ContosoShop/Product.xaml?category={category}"
نمونه آدرس نهایی نمایش داده شده	/Product/bikes
نگاشت نهایی در برنامه	/ContosoShop/Product.xaml?category=bikes

در مثال پیچیده‌ی ذیل، صفحات و همچنین Query Strings با کمک {} به شکل متغیر تعریف شده‌اند:

تعریف نگاشت	Uri = "{reporttype}/{month}/{format}" (in XAML) MappedUri = "/Views/Reports/{reporttype}.xaml?time={month}&show={format}" (in Visual Basic or C#) MappedUri = "/Views/Reports/{reporttype}.xaml?time={month}&show={format}"
نمونه آدرس نهایی نمایش داده شده	/Sales/June/Short
نگاشت نهایی در برنامه	/Views/Reports/Sales.xaml?time=June&show=Short

نحوه‌ی دسترسی به مقادیر Query Strings

برای دسترسی به Query String تعریف شده‌ای به نام ProductId در کدهای برنامه می‌توان به شکل زیر عمل کرد:

C#

```
if (this.NavigationContext.QueryString.ContainsKey("ProductId"))
{
    productID = this.NavigationContext.QueryString["ProductId"];
}
```

راهنمای کاربران به صفحاتی در خارج از برنامه

امکان ارائه‌ی پیوندهایی به صفحات وب خارج از برنامه نیز در Silverlight وجود دارد. برای مثال طراحی منوی یک سایت با استفاده از Silverlight و سپس تعریف پیوندهایی به قسمت‌های مختلف آن. برای این منظور می‌توان از کنترل HyperlinkButton استفاده کرد:

XAML

```
<HyperlinkButton NavigateUri="http://www.microsoft.com"
    Content="Go to Microsoft" TargetName="_new" />
```

اگر به هر علتی در برنامه‌ی خود قصد داشتید که پیوندهای به سایت‌های خارجی را غیرفعال کنید، تنظیم بعد را به جایی که افزونه‌ی Silverlight را تعریف کرده‌اید، اضافه نمائید:

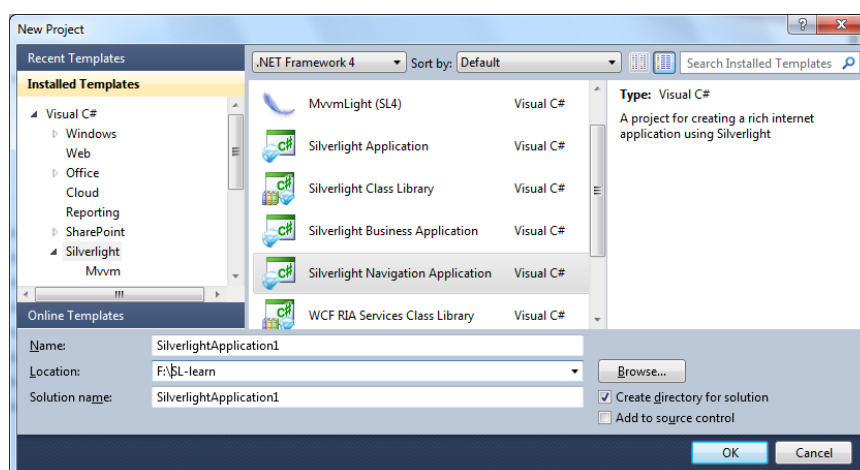
HTML

```
<object ...>
  <param name="enableNavigation" value="none"/>
  ...
</object>
```

به این صورت تنها هدایت کاربران به صفحات داخلی برنامه مجاز بوده و کلیه‌ی راهبری‌های تعریف شده به صفحاتی در خارج از برنامه غیرفعال می‌گردند. باید در نظر داشت که در این حالت، تاریخچه‌ی بازدید صفحات مرورگر نیز غیرفعال خواهد شد.

قالب ایجاد پروژه‌هایی مبتنی بر سیستم Silverlight در Navigation

اگر علاقمند باشید که موارد ذکر شده جهت معرفی سیستم Navigation Framework به شکل ساده‌تری در اختیار شما باشند، یکی از قالب‌های آغاز پروژه‌های Silverlight از همین دست می‌باشد (شکل ۴).



شکل ۴- قالب ایجاد پروژه‌هایی مبتنی بر سیستم Silverlight در Navigation

پس از آغاز پروژه‌ایی از نوع Silverlight Navigation Application، یک Frame در صفحه‌ی اصلی به همراه قالب تهیه‌ی نگاشت‌های آدرس‌ها به صورت خودکار ایجاد خواهد گردید. در اینجا همچنین پوشه‌های Views و Assets نیز جهت تعاریف ساده‌تر صفحات و قالب‌های آن‌ها به پروژه اضافه شده‌اند. علاوه بر آن فضا‌های نام لازم به همراه ارجاعات ضروری به یک سری از اسمبلی‌های این سیستم نیز به صورت خودکار الحاق گردیده‌اند.