



31 Days of Windows 8

# Windows 8 开发 31 日

## 第 13 日

## 页面导航

译者：BeyondVincent(破船)

时间：2013.4.25

版本： 2.0

## 关于破船

程序猿砌墙于云南昆明!

长期扎根移动软件开发!

爱跑步爱打篮球爱运动!

命中无大富大贵之面相!

愿健康与平淡相随一生!

你可以发邮件与破船取得联系: [BeyondVincent@gmail.com](mailto:BeyondVincent@gmail.com)

还可以关注破船的微博: [腾讯微博](#)和[新浪微博](#)。

这里是破船的个人博客, 欢迎光临: [破船之家](#)



## 关于 Windows 8 开发 31 日翻译



Windows 8 开发 31 日是由 Jeff Blankenburg 和 Clark Sell 原创的。

官方站点：<http://31daysofwindows8.com/>

涉及到两个版本：

XAML/C# ( 由 Jeff Blankenburg 撰写 )

HTML5/JS ( 由 Clark Sell 撰写 )

其中涉及到的资源和相关代码请到这里下载：

<https://github.com/csell5/31DaysOfWindows8>

在这里，由于破船对 HTML5/JS 不熟悉，所以只翻译 XAML/C# 相关主题。

建议大家前往看原创内容，如果看不明白，再来这里看我翻译的相关内容。

如果翻译不正确的地方，可以通过上面的联系方式告诉破船。

破船祝你阅读愉快！



目录

关于破船 ..... 2

关于 Windows 8 开发 31 日翻译 ..... 3

目录 4

第 13 日页面导航 ..... 5

1.0. 介绍 .....5

1.1. Windows 8 中 XAML 页面间导航...5

1.2. 在 Windows 8 中的 XAML 页面间传递数据 9

1.3. 缓存页面——提高点击返回按钮时的一些效率 11

1.4. 总结 ..... 11



## 第 13 日页面导航



### 1.0. 介绍

我已经花了几天时间介绍关于更新系统信息，让用户知道什么正在发生！实际上，在本系列文章的第 12 日中，我已经稍微介绍了在 Windows 8 开发中我们会遇到的一个最常见的问题之一：XAML 页面间的导航。

今天，我将详细介绍这部分内容。本文会涉及到 3 部分内容：

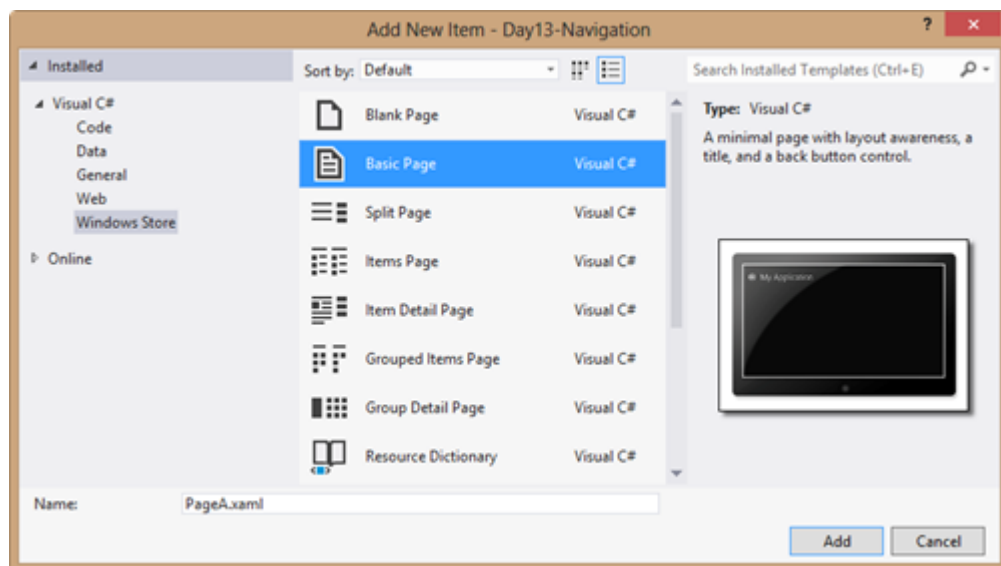
- 简单的从页面 A 导航至页面 B。导航的时候会发生什么，以及对于开发者来说什么信息可用？
- 将数据从一个页面传递到另外一个页面。不仅可以传递字符串和整型，还可以在页面间传递整个对象。
- 缓存页面。当用户点击返回按钮时，他们并不希望已经输入的数据丢失了。稍后将讨论。

### 1.1. Windows 8 中 XAML 页面间导航

当我们在一个新的平台上开发时，好像都会遇到这样的问题。在 XAML 程序

中，同样有新的方法在页面间进行导航。Windows 8 在这方面没有什么不同的。通过创建一些页面，可以很容易的解释如何导航。下面开始吧。

创建一个新的 Blank App Template ,然后添加两个新的 Basic Page items 到工程中。在本示例中，页面分别为：PageA.xaml 和 PageB.xaml。

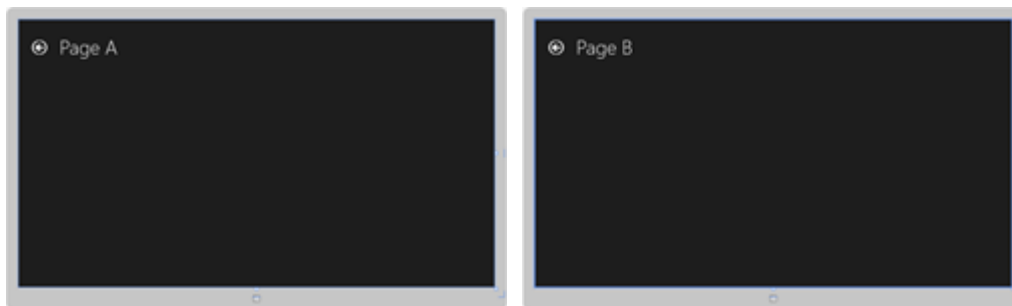


在每个页面中，这里都有一行 XAML 代码来设置 AppName。

```
<Page.Resources>
<x:String x:Key="AppName">Page A</x:String>
</Page.Resources>
```

一般情况，我会将上面的内容移到我的 App.xaml 文件中，让其在程序中的所有页面都可以使用。而今天，我在每个页面写一个，是为了当加载页面时，区分页面中的不同名字。我将 PageA.xaml 中的 AppName 设置为“Page A”，而 PageB.xaml 中的 AppName 则设置为“Page B”，页面看起来是这样的：





为了让示例简单起见，我同样修改了一下 App.xaml.cs 文件。在 OnLaunched() 方法的最后面，代码看起来是这样的：

```
if (rootFrame.Content == null)
{
    if (!rootFrame.Navigate(typeof(PageA), args.Arguments))
    {
        throw new Exception("Failed to create initial page");
    }
}
```

看上面的代码，是将 typeof(MainPage) 修改为 typeof(PageA)。

这样当程序启动的时候，会默认加载 PageAnt.xaml。如果你现在运行程序的话，你将首先看到 [Splash Screen](#)，然后是 PageA。可能你已经注意到了在 Visual Studio 2012 的设计器中，在页面标题旁边有一个将头图标，它被绑定到了 Frame.CanGoBack 属性——页面是否能够返回。稍后会再次介绍。

现在，我们开始导航吧。在 App.xaml.cs 文件中，你可以看到相关导航的技术，不过如果你是从一个页面导航到另外一个页面，有一点点不同。我在 PageA.xaml 文件的<VisualStateManager>节点之前，添加了一个<Button>控件，如下代码：

```
<Button x:Name="PageBButton"
        Grid.Row="1"
        Content="To Page B"
        Width="150"
        Height="150"/>
```



```
Click="PageBButton_Click" />
```

在<Button>的 event handler 中，我写了第一个导航代码片段：

```
private void PageBButton_Click(object sender, RoutedEventArgs e)
{
    Frame.Navigate(typeof(PageB));
}
```

如上所示，处理过程很简单。下面我们让整个程序更加完整点吧。在 PageB 中添加一个导航按钮。这好像只是一个重复性的任务，但是你会发现很快就会遇到麻烦。（最好运行程序，然后实际的导航到 Page B 中）。同样在 PageB 中添加一个<Button>和相关的 event handler，在 event handler 中，导航到 PageA。现在，每个页面都有一个按钮，并分别导航到另外一个页面。如果你只用这两个按钮进行导航，那么最终将会创建一个巨大的回退导航栈(back navigation stack)，这与返回按钮的概念一点都没有关系。我们想要完成返回功能，则需要实现 LayoutAwarePage 对象中的 GoHome event handler。为了有效的演示这个程序，在 PageB.xaml 中添加 2 个按钮。在相关的事件处理中，不再使用导航至某个页面，而是创建一个 GoHome 事件，然后调用它。如下：

【译者注：原文作者可能表达有误：明明说了要调用自己 override 的 GoHome 方法，结果是去调用 base 里面的 GoHome 方法。】

```
protected override void GoHome(object sender, RoutedEventArgs e)
{
    if (this.Frame != null)
    {
        while (this.Frame.CanGoBack) this.Frame.GoBack();
    }
}
```





```
privatevoid HomeButton_Click(object sender, RoutedEventArgs e)
{
    GoHome(sender, e);
}
```

如上所示，我写的 GoHome 方法是一个循环的尝试去调用“GoBack”，直到不能再 GoBack 了。在这里，最终会返回到导航栈的顶部，也就是初始页面。

### 1.2. 在 Windows 8 中的 XAML 页面间传递数据

下面，我将介绍如何在页面间传递数据。可以是一个对象，一个字符串，只要是数据，都可以搞定。事实证明，Frame.Navigate()方法的重载方法中是可以接收一个数据当做参数的。在这个示例中，我在 PageA 中，添加了一个 TextBox 和一个 Button，我假设你已经知道如何添加了。在这里的重点是我给按钮添加的事件，该事件从 TextBox 中获取 text 值，然后将其当做 Navigate()方法的一个参数，如下：

```
privatevoid GoButton_Click(object sender, RoutedEventArgs e)
{
    Frame.Navigate(typeof(PageB), TextBoxValue.Text);
}
```

在 PageB 页面中，我们需要获取到这个值，然后将其显示出来。同样，我添加了一个 TextBlock 控件（你可以在文章的最后，下载到该示例），这次，重点是 OnNavigatedToevent handler，在这个方法中获取到 PageA 中传递过来的数据：

```
protectedoverridevoid OnNavigatedTo(NavigationEventArgs e)
{
    base.OnNavigatedTo(e);

    string text = e.Parameter asstring;
```



```

if (!string.IsNullOrEmpty(text))
{
    TextBlockValue.Text = text;
}
else
{
    TextBlockValue.Text = "You need to pass a string. Press back and enter a value.";
}

//Element gallium = e.Parameter as Element;
//TextBlockValue.Text = gallium.Category;
}

```

如上代码，我从 [NavigationEventArgs.Parameter](#) 中获取传递过来的参数，并将其转换为 string 值。你同样需要将传递过来的数据转换为适当的类型。

另外，之前我提到过用同样的机制可以传递完整的一个对象。还记得第 04 日中 Element 吗？下面，我将把自定义的 Element 对象从 PageA 中传递到 PageB 中。下面是 PageA 中的代码：

```

privatevoid Gallium_Click(object sender, RoutedEventArgs e)
{
    Frame.Navigate(typeof(PageB), newElement { AtomicNumber = 31,
                                                AtomicWeight = 69.72,
                                                Category = "Other Metals",
                                                Name = "Gallium",
                                                Symbol = "Ga",
                                                State = "Solid" });
}

```

可以清楚的看到，我传递了一个新的 Element 对象到 PageB 中，下面是 PageB 中获取数据的代码：

```

protectedoverridevoid OnNavigatedTo(NavigationEventArgs e)
{
    base.OnNavigatedTo(e);

    Element gallium = e.Parameter asElement;
    TextBlockValue.Text = gallium.Category;
}

```



### 1.3. 缓存页面——提高点击返回按钮时的一些效率

最后，我们来看看与导航相关的最后一个主题：页面缓存。这是最后一个示例，我将使用上面的工程，其中的状态：我传递一个字符串值——前进和后退。你可能已经注意到，如果你创建一个自己的工程，然后在 PageA 中的 TextBox 输入一些内容，然后导航到 PageB，当你点击返回按钮，返回到 PageA 时，PageA 并没有记住你之前输入的内容。这是非常糟糕的用户体验。

当然，我们可以很容易的避免这个问题，只需要在我们想要缓存的页面的构造行数中添加一行代码即可：

```
public PageA()
{
    this.InitializeComponent();
    NavigationCacheMode = NavigationCacheMode.Enabled;
}
```

设置了 NavigationCacheMode 之后，你会发现，当你从 PageB 返回 PageA 时，之前输入的 TextBox 值（页面中任何其它内容）将被缓存在回退栈（back stack）中。试试吧...这将会是很好的体验，当用户返回页面是，并不需要我们填充和重新填充内容。

### 1.4. 总结

这里，我们已经广泛的学习了 Windows 8 中 XAML 程序的导航。包括了页面导航，页面间数据传递，以及当用户点击返回按钮时缓存相关页面。



点击下图，可以下载本文的全部示例代码。



明天，我将开始介绍如何使用 GeoLocation 确定用户当前的位置，包括相关规则，以及使用到的技术。到时候见！



感谢你的阅读！

如果对这篇文章有什么想法想法，可以与破船联系，破船的联系方式在文章开头。

破船

