# WPF疑难解惑（1）

## 课程介绍

1. 课程回顾：《**Microsoft Blend 2019**》，40节
2. 课程回顾：《**WPF经典编程实例**》，100节
3. 课程来源：学员在交流群中提出疑难问题。
4. 课程难度：中高
5. 课程基础：熟练运用C#、熟练运用WPF、熟练运用Visual Studio 2019
6. 课程源码：<https://github.com/zmrbak/WPFqa1>
7. 课程进度：来源于学员的提问和建议，初步估计100集
8. 课程产生：学员提问\建议->老师研究->为学员解答->录制视频。
9. **注意事项：本课程基于Visual Studio 2019/2022进行演示和讲解，请使用课程中所示VS版本练习**。

## 如何让TextBlock控件中的文本垂直对齐

课程来源：学员咨询关于TextBlock**文本垂直居中**的问题

老师解答：换用Label控件

## 如何动态地复制和粘贴控件

课程来源：学员咨询“**如何把XAML控件保存下来**”

老师解答：XamlReader、XamlWriter

## 如何为粘贴的控件添加事件

课程来源：上节课的延伸

老师解答：VisualTreeHelper

((Button)(((StackPanel)uIElement).Children[2])).Click += Button\_Click\_1;

## 如何取出被动态添加的控件的属性

课程来源：上节课的延伸

老师解答：

foreach (var item in sp1.Children)

if (item is StackPanel stackPanel)

TextBox textBox = stackPanel.Children[1] as TextBox;

textBox.Text

## XAML绑定跟踪调试与虚拟值转换器

课程来源：学员咨询“在XAML中绑定的数据为什么不生效”

老师解答：

xmlns:dbg="clr-namespace:System.Diagnostics;assembly=WindowsBase"

dbg :PresentationTraceSources.TraceLevel=High

Converter={StaticResource DebugConverter}

public class DebugConverter : IValueConverter

Debugger.Break();

return value;

## 如何在WPF中播放Flash动画

课程来源：学员提问“在wpf中能播放flash动画吗？”

老师解答：

AxInterop.ShockwaveFlashObjects

Interop.ShockwaveFlashObjects

WindowsFormsIntegration

System.Windows.Forms

xmlns:f1="clr-namespace:AxShockwaveFlashObjects;assembly=AxInterop.ShockwaveFlashObjects"

<WindowsFormsHost >

<f1:AxShockwaveFlash x:Name="flashShow" />

</WindowsFormsHost>

flashShow.Movie = Environment.CurrentDirectory + @"\flashtime-0028.swf";

## 如何绑定到用户控件中某控件的属性

**课程来源：**学员提问“我想把一个窗口里的元素的IsEnabled属性，绑定到UserControl里面的一个元素的IsEnabled属性，如何实现呢？”

**老师解答：**在UserControl中添加依赖项属性

## 如何在XAML代码中添加C#代码

**课程来源：**学员提问“我想把C#代码写到XAML中，如何实现呢？”

**老师解答：<x:Code>…代码…</x:Code>**

## 如何引用子文件夹中的dll

**课程来源：**学员提问“怎么引用子文件夹中的dll?”

**老师解答：<runtime>**

## 在页面（Page）切换的时候，如何保证原页面的数据不变？

**课程来源：**学员提问：如何保证页面切换回来的时候，原来打开的页面内容不变？

**老师解答：**Application.Current.Properties["DataB"] = tbB.Text;

## 如何为TextBox添加虚拟提示信息

**课程来源：**学员提问：我想给TextBox添加一个背景信息，用户没有添加信息的时候显示提示信息，用户输入的时候，让提示信息消失。在WPF中怎么做？

**老师解答：**Style.Triggers

## 如何解决Mutex单实例失效的问题

**课程来源：**学员提问：按照《WPF经典编程实例》中，第7课《如何创建一个单实例的WPF应用程序》，自己写的程序不能实现单实例运行，请问如何解决？

**老师解答：**使用静态变量

## 如何为其它程序套个WPF的外衣？

**课程来源：**学员提问：我看到有一个PC微信管理软件，启动之后把PC微信的界面嵌在它里面，同时在PC微信外面编写了很多操作微信的功能按钮，在WPF中如何实现？

**老师解答：Win32Api**

## 如何扑捉退格键旁边的那两个键（-，=）的键值呢？

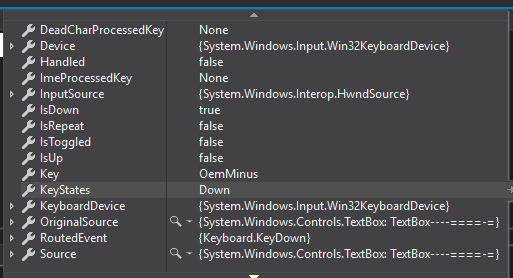
**课程来源：**学员提问：键盘Key的枚举值没有等号，按下键盘上的“=”键的时候，如何判断“=”呢？

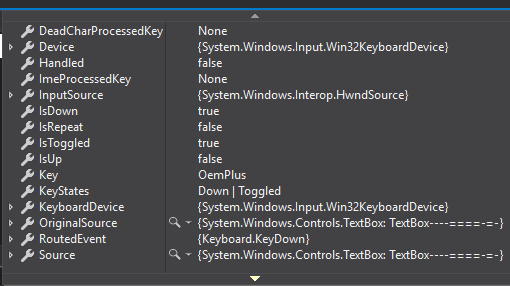
**老师解答：**PreviewKeyDown

TextBox\_KeyDown

中文输入法：不接收-=

英文输入法：可接收

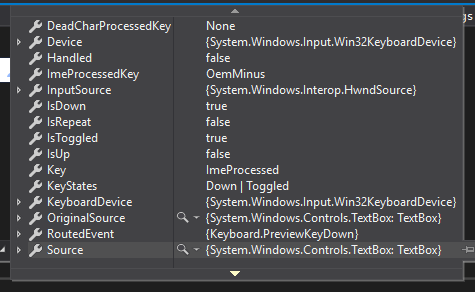


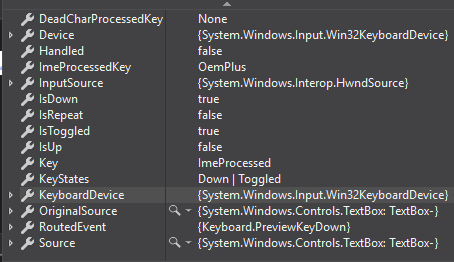


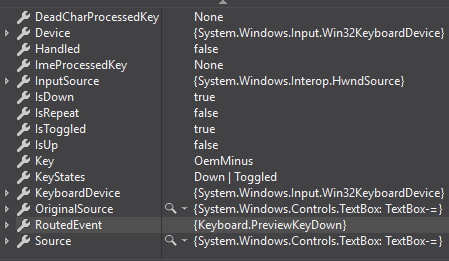
TextBox\_PreviewKeyDown

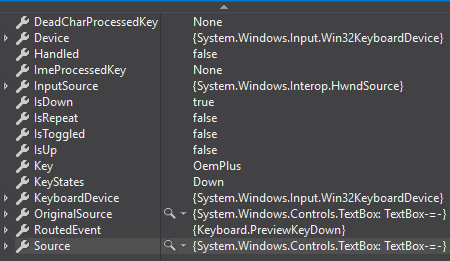
中文输入法：可接收-=

英文输入法：可接收-









## 在自定义的控件中如何用VisualState切换控件属性？

**课程来源：**学员提问：请问我自定义的控件继承自textbox，用visualstate来切换状态时，怎么设置自身的背景颜色属性？

**老师解答：**请参考示例

## 如何为TreeView绑定多层数据？

**课程来源：**学员提问：TreeView绑定的数据应该怎样做，我的只能绑定第一层？

**老师解答**：HierarchicalDataTemplate

## 在WPF中如何使用第三方字体？

**课程来源：**学员提问：我从网上下载了一个字体，如何用到我的WPF中？

**老师解答**：将字体拖入项目中，为控件选择该字体。

## 如何解决拖动无边框窗口菜单项被干扰的问题？

**课程来源：**学员提问：我给窗口设置了拖动事件，但菜单却会闪烁，怎么不要干扰到菜单呢？

**老师解答：**在拖动窗口的时候，设置菜单的IsHitTestVisible属性。

## 如何用纯C#代码向TextBlock中添加一个Grid？

**课程来源：**学员提问：用后台代码如何实现向TextBlock中添加Grid？

**老师解答：**textBlock.Inlines.Add()

## 如何让WPF中的WindowsFormsHost控件不在最顶层？



WPF经典编程实例：95，如何在WPF应用程序中驻留WinForm控件

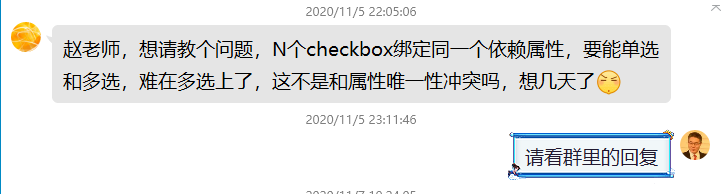
<https://www.xpnas.com/posts/3889909467.html>

## 如何有效使用WPF中虚拟化技术？



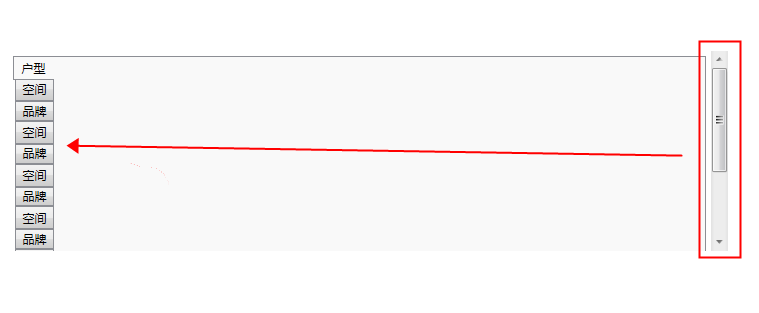
<https://github.com/zmrbak/WeChatContractPurifier>

## 如何让多个CheckBox共同绑定一个属性，而且能够多选？



## .NET序列化与反序列化

## 如何为TabControl中的TabItem添加滚动条



## 如何在.Net Framework中使用最新的C#10

C# 10.0 向 C# 语言添加了以下功能和增强功能（从 .NET 6 预览 7 开始）：

* global using 指令

可将 global 修饰符添加到任何 using 指令，以指示编译器该指令适用于编译中的所有源文件。 这通常是项目中的所有源文件。

* 文件范围的命名空间声明
* 扩展属性模式
* 可使用 const 内插字符串
* 记录类型可密封 ToString()
* 在同一析构中可同时进行赋值和声明
* 可在方法上使用 AsyncMethodBuilder 属性

使用 .NET 6.0 预览 5 时，此功能要求将 **.csproj** 文件中的 <LangVersion> 元素设置为 **preview**。

## 如何为PasswordBox控件绑定密码（1）

## 在WPF中绑定的坑

## 如何为PasswordBox控件绑定密码（2）

## 如何在后台代码中修改控件模板中属性值

派生Button类，添加一个依赖项属性

public class MyButton: Button

{

public ImageSource ImageFile

{

get { return (ImageSource)GetValue(ImageFileProperty); }

set { SetValue(ImageFileProperty, value); }

}

// Using a DependencyProperty as the backing store for ImageFile. This enables animation, styling, binding, etc...

public static readonly DependencyProperty ImageFileProperty =

DependencyProperty.Register("ImageFile", typeof(ImageSource), typeof(MyButton), new PropertyMetadata());

}

<ControlTemplate x:Key="ButtonTemplate1" TargetType="{x:Type local:MyButton}">

btn1.ImageFile = new BitmapImage(new Uri("34521.ico",UriKind.RelativeOrAbsolute));

依赖项属性

local:ExImage.ImageFile="3452.png"

public class ExImage

{

public static ImageSource GetImageFile(DependencyObject obj)

{

return (ImageSource)obj.GetValue(ImageFileProperty);

}

public static void SetImageFile(DependencyObject obj, ImageSource value)

{

obj.SetValue(ImageFileProperty, value);

}

// Using a DependencyProperty as the backing store for ImageFile. This enables animation, styling, binding, etc...

public static readonly DependencyProperty ImageFileProperty =

DependencyProperty.RegisterAttached("ImageFile", typeof(ImageSource), typeof(ExImage), new PropertyMetadata());

}

ExImage.SetImageFile(btn1, new BitmapImage(new Uri("34521.ico", UriKind.RelativeOrAbsolute)));

## 控件模板中属性类型绑定的问题

<TextBlock Text="{TemplateBinding MyInt, **Converter**={StaticResource IntStringConverter}}" />

## XAML名称空间中神秘的网址

xmlns:MyNamespace="clr-namespace:WpfCustomControlLibrary1;assembly=WpfCustomControlLibrary1"

xmlns:MyNamespace="clr-namespace:WpfCustomControlLibrary1;assembly=WpfCustomControlLibrary1"

xmlns:local="clr-namespace:Wpfq32"

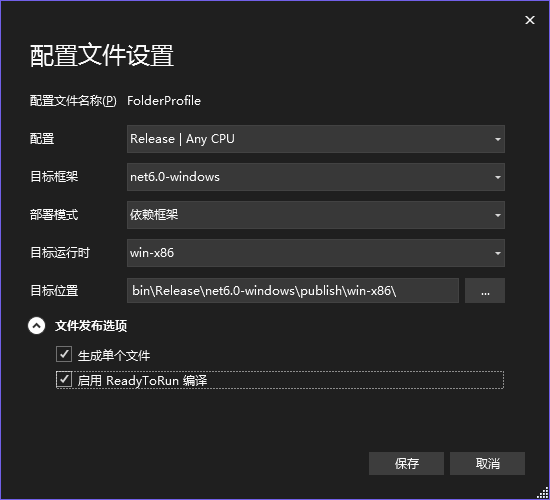
<https://docs.microsoft.com/zh-cn/dotnet/api/system.windows.markup.xmlnsprefixattribute?view=netframework-4.8>

一个uri可以对应多个名称空间

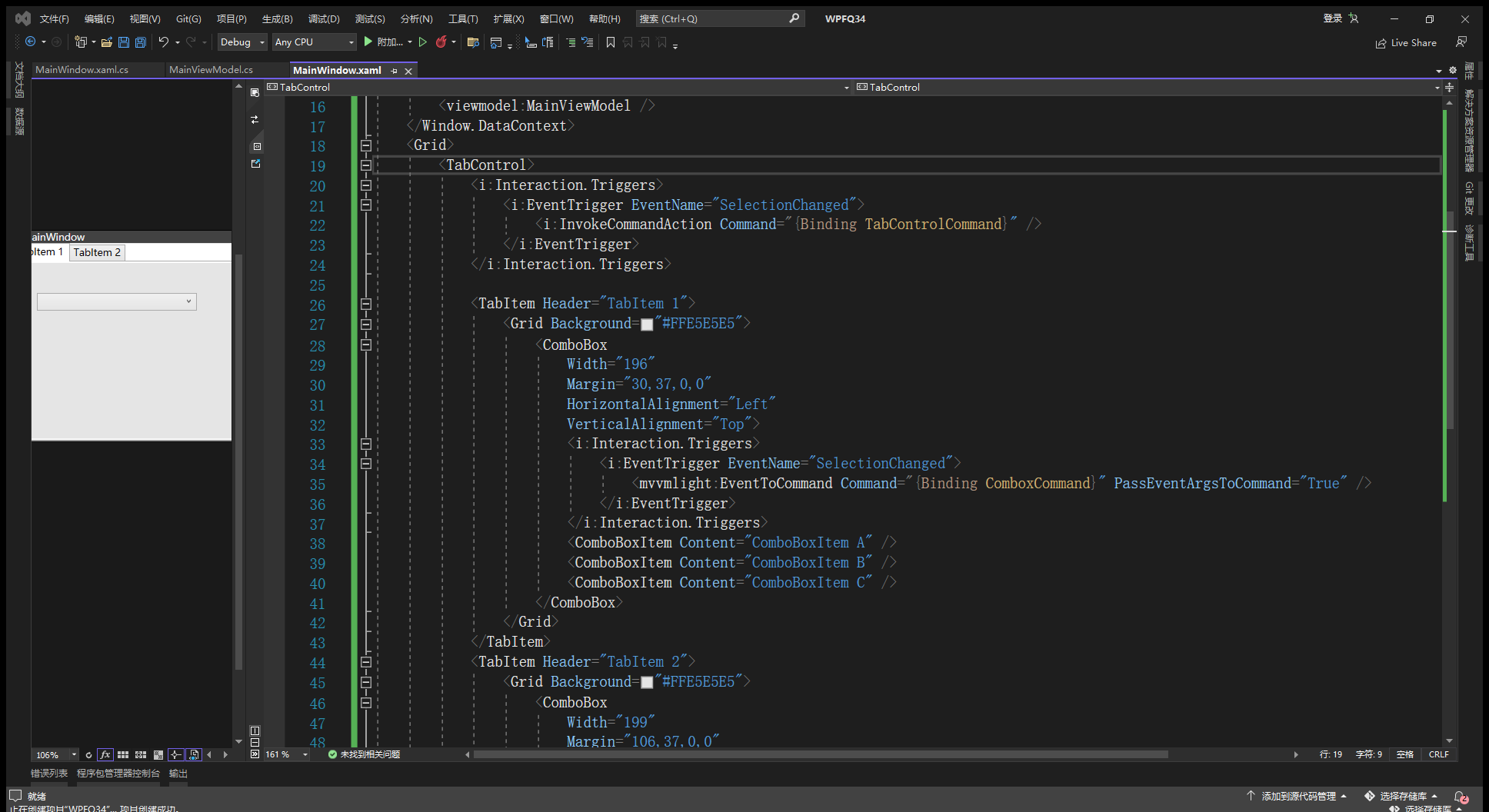
在项目中，不可以使用自身的uri名称空间

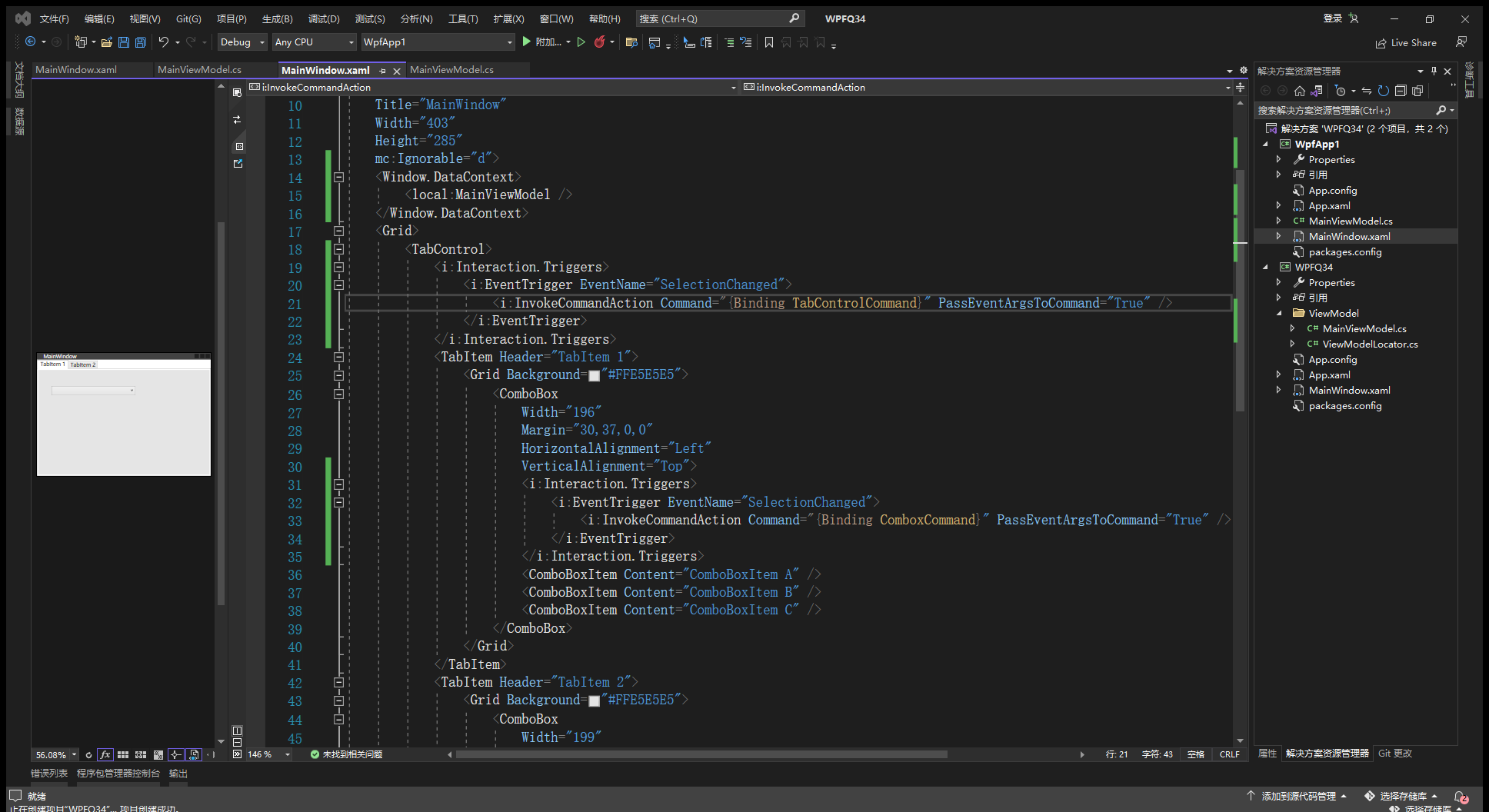
可以将自身项目名称空间附加到知名的uri名称空间，但不推荐。

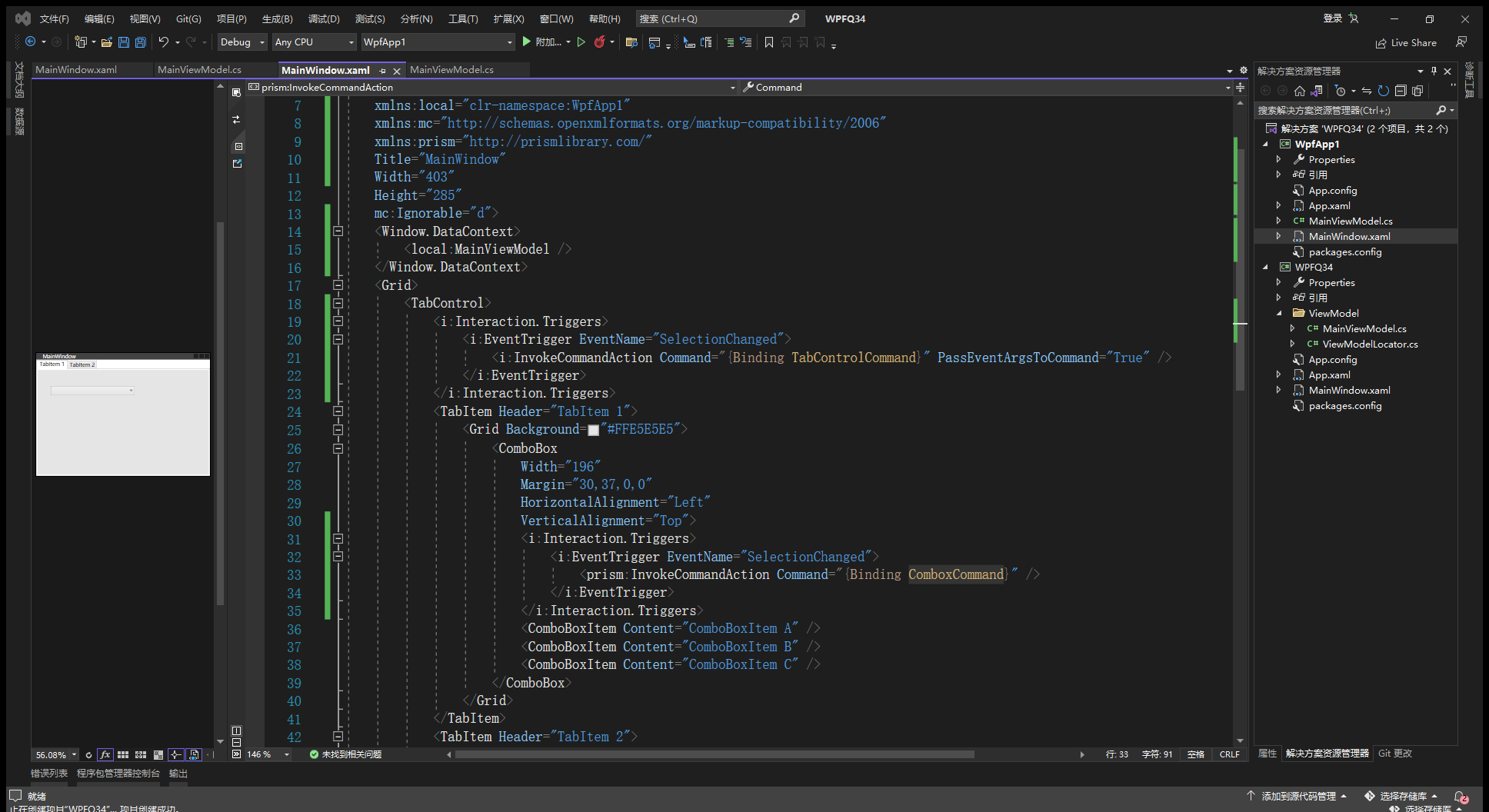
## 如何将.NET Core程序编译成一个不带DLL的可执行程序



## 如何在MVVM中阻止冒泡传播







## System.Windows.Interactivity与Microsoft.Xaml.Behaviors的联系和区别

支持：PassEventArgsToCommand

xmlns:i="http://schemas.microsoft.com/xaml/behaviors"

System.Windows.Interactivity

不支持PassEventArgsToCommand

xmlns:i=<http://schemas.microsoft.com/expression/2010/interactivity>

Microsoft.Xaml.Behaviors

System.Configuration

**System.Windows.Interactivity**

C:\Users\libit\source\repos\WPFQ34\packages\MvvmLightLibs.5.4.1.1\lib\net45\System.Windows.Interactivity.dll

[assembly: **XmlnsPrefix**("http://schemas.microsoft.com/expression/2010/interactivity", "**i**")]

[assembly: XmlnsDefinition("http://schemas.microsoft.com/expression/2010/interactivity", "System.Windows.Interactivity")]

**Microsoft.Xaml.Behaviors**

C:\Users\libit\source\repos\WPFQ34\packages\Microsoft.Xaml.Behaviors.Wpf.1.1.31\lib\net45\Microsoft.Xaml.Behaviors.dll

[assembly: **XmlnsPrefix**("http://schemas.microsoft.com/xaml/behaviors", "**b**")]

[assembly: XmlnsDefinition("http://schemas.microsoft.com/xaml/behaviors", "Microsoft.Xaml.Behaviors")]

[assembly: XmlnsDefinition("http://schemas.microsoft.com/xaml/behaviors", "Microsoft.Xaml.Behaviors.Core")]

[assembly: XmlnsDefinition("http://schemas.microsoft.com/xaml/behaviors", "Microsoft.Xaml.Behaviors.Input")]

[assembly: XmlnsDefinition("http://schemas.microsoft.com/xaml/behaviors", "Microsoft.Xaml.Behaviors.Layout")]

[assembly: XmlnsDefinition("http://schemas.microsoft.com/xaml/behaviors", "Microsoft.Xaml.Behaviors.Media")]

**System.Configuration**

C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Configuration.dll

## WPF中控件属性的联动

第一个控件某个属性改变后，第二个控件的某个属性跟着改变。

|  |
| --- |
| <StackPanel>  <TextBlock Text="使用后台代码" />  <Separator />  <TextBlock x:Name="tb1" />  <CheckBox  Click="CheckBox\_Click"  Content="修改内容"  Loaded="CheckBox\_Click" />  </StackPanel> |

|  |
| --- |
| private void CheckBox\_Click(object sender, RoutedEventArgs e)  {  if(sender is CheckBox checkBox)  {  if(checkBox.IsChecked==true)  {  this.tb1.Text = "111";  }  else  {  this.tb1.Text = "222";  }  }  } |

|  |
| --- |
| <StackPanel>  <TextBlock Text="使用绑定 Tag" />  <Separator />  <TextBlock x:Name="tb2" Text="{Binding Tag, ElementName=cb2}" />  <CheckBox x:Name="cb2" Content="修改内容">  <CheckBox.Style>  <Style TargetType="CheckBox">  <Style.Triggers>  <Trigger Property="IsChecked" Value="False">  <Setter Property="Tag" Value="222" />  </Trigger>  <Trigger Property="IsChecked" Value="True">  <Setter Property="Tag" Value="111" />  </Trigger>  </Style.Triggers>  </Style>  </CheckBox.Style>  </CheckBox>  </StackPanel> |

|  |
| --- |
| <StackPanel>  <TextBlock Text="使用绑定 Tag" />  <Separator />  <TextBlock x:Name="tb2" Text="{Binding Tag, ElementName=cb2}" />  <CheckBox x:Name="cb2" Content="修改内容">  <CheckBox.Style>  <Style TargetType="CheckBox">  <Style.Triggers>  <Trigger Property="IsChecked" Value="False">  <Setter Property="Tag" Value="{Binding TextBlockA}" />  </Trigger>  <Trigger Property="IsChecked" Value="True">  <Setter Property="Tag" Value="{Binding TextBlockB}" />  </Trigger>  </Style.Triggers>  </Style>  </CheckBox.Style>  </CheckBox>  </StackPanel> |

|  |
| --- |
| public partial class MainWindowViewModel : ObservableObject  {  [ObservableProperty]  private string textBlockA = "222";  [ObservableProperty]  private string textBlockB = "111";  } |

|  |
| --- |
| <StackPanel>  <TextBlock Text="数据触发器 + Tag 绑定" />  <Separator />  <TextBlock x:Name="tb3" Text="{Binding Tag, ElementName=cb3}" />  <CheckBox x:Name="cb3" Content="修改内容">  <CheckBox.Style>  <Style TargetType="CheckBox">  <Style.Triggers>  <DataTrigger Binding="{Binding IsChecked, ElementName=cb3}" Value="False">  <Setter Property="Tag" Value="{Binding TextBlockA}" />  </DataTrigger>  <DataTrigger Binding="{Binding IsChecked, ElementName=cb3}" Value="True">  <Setter Property="Tag" Value="{Binding TextBlockB}" />  </DataTrigger>  </Style.Triggers>  </Style>  </CheckBox.Style>  </CheckBox>  </StackPanel> |

|  |
| --- |
| <StackPanel>  <TextBlock Text="数据触发器" />  <Separator />  <TextBlock x:Name="tb4">  <TextBlock.Style>  <Style TargetType="TextBlock">  <Style.Triggers>  <DataTrigger Binding="{Binding IsChecked, ElementName=cb4}" Value="False">  <Setter Property="Text" Value="{Binding TextBlockA}" />  </DataTrigger>  <DataTrigger Binding="{Binding IsChecked, ElementName=cb4}" Value="True">  <Setter Property="Text" Value="{Binding TextBlockB}" />  </DataTrigger>  </Style.Triggers>  </Style>  </TextBlock.Style>  </TextBlock>  <CheckBox x:Name="cb4" Content="修改内容" />  </StackPanel> |

|  |
| --- |
| <StackPanel>  <StackPanel.Resources>  <Storyboard  x:Key="tb5\_sb1"  Storyboard.TargetName="tb5"  Storyboard.TargetProperty="Text">  <StringAnimationUsingKeyFrames>  <DiscreteStringKeyFrame KeyTime="0" Value="{Binding TextBlockB}" />  </StringAnimationUsingKeyFrames>  </Storyboard>  <Storyboard  x:Key="tb5\_sb2"  Storyboard.TargetName="tb5"  Storyboard.TargetProperty="Text">  <StringAnimationUsingKeyFrames>  <DiscreteStringKeyFrame KeyTime="0" Value="{Binding TextBlockA}" />  </StringAnimationUsingKeyFrames>  </Storyboard>  </StackPanel.Resources>  <TextBlock Text="使用故事版 + EventTrigger" />  <Separator />  <TextBlock x:Name="tb5" />  <CheckBox x:Name="cb5" Content="修改内容">  <CheckBox.Triggers>  <EventTrigger RoutedEvent="CheckBox.Checked">  <BeginStoryboard Storyboard="{StaticResource tb5\_sb1}" />  </EventTrigger>  <EventTrigger RoutedEvent="CheckBox.Unchecked">  <BeginStoryboard Storyboard="{StaticResource tb5\_sb2}" />  </EventTrigger>  <EventTrigger RoutedEvent="CheckBox.Loaded">  <BeginStoryboard Storyboard="{StaticResource tb5\_sb2}" />  </EventTrigger>  </CheckBox.Triggers>  </CheckBox>  </StackPanel> |

|  |
| --- |
| <StackPanel>  <TextBlock Text="使用ViewModel中间值 + Command" />  <Separator />  <TextBlock x:Name="tb6" Text="{Binding TextBlockC}" />  <CheckBox  x:Name="cb6"  Command="{Binding CBChangedCommand}"  CommandParameter="{Binding IsChecked, ElementName=cb6}"  Content="修改内容" />  </StackPanel> |

|  |
| --- |
| public partial class MainWindowViewModel : ObservableObject  {  [ObservableProperty]  private string textBlockA = "222";  [ObservableProperty]  private string textBlockB = "111";  **[ObservableProperty]**  **private string textBlockC = "222";**  [RelayCommand]  private void CBChanged(bool arg)  {  TextBlockC = arg ? TextBlockB : TextBlockA;  }  } |

|  |
| --- |
| <StackPanel>  <TextBlock Text="使用ViewModel中间值 " />  <Separator />  <TextBlock x:Name="tb7" Text="{Binding TextBlockC}" />  <CheckBox  x:Name="cb7"  Content="修改内容"  IsChecked="{Binding IsCheckBoxChecked}" />  </StackPanel> |

|  |
| --- |
| [ObservableProperty]  [NotifyPropertyChangedFor(nameof(TextBlockC))]  private bool isCheckBoxChecked;  partial void OnIsCheckBoxCheckedChanged(bool value)  {  TextBlockC = value ? TextBlockB : TextBlockA;  } |

|  |
| --- |
| xmlns:i="http://schemas.microsoft.com/xaml/behaviors" |

|  |
| --- |
| <StackPanel>  <StackPanel.Resources>  <Storyboard  x:Key="tb8\_sb1"  Storyboard.TargetName="tb8"  Storyboard.TargetProperty="Text">  <StringAnimationUsingKeyFrames>  <DiscreteStringKeyFrame KeyTime="0" Value="{Binding TextBlockB}" />  </StringAnimationUsingKeyFrames>  </Storyboard>  <Storyboard  x:Key="tb8\_sb2"  Storyboard.TargetName="tb8"  Storyboard.TargetProperty="Text">  <StringAnimationUsingKeyFrames>  <DiscreteStringKeyFrame KeyTime="0" Value="{Binding TextBlockA}" />  </StringAnimationUsingKeyFrames>  </Storyboard>  </StackPanel.Resources>  <TextBlock Text="使用 behavior 中的事件触发器 、故事版" />  <Separator />  <TextBlock x:Name="tb8" />  <CheckBox x:Name="cb8" Content="修改内容">  <i:Interaction.Triggers>  <i:EventTrigger EventName="Checked">  <i:ControlStoryboardAction ControlStoryboardOption="Play" Storyboard="{StaticResource tb8\_sb1}" />  </i:EventTrigger>  <i:EventTrigger EventName="Unchecked">  <i:ControlStoryboardAction ControlStoryboardOption="Play" Storyboard="{StaticResource tb8\_sb2}" />  </i:EventTrigger>  <i:EventTrigger EventName="Loaded">  <i:ControlStoryboardAction ControlStoryboardOption="Play" Storyboard="{StaticResource tb8\_sb2}" />  </i:EventTrigger>  </i:Interaction.Triggers>  </CheckBox>  </StackPanel> |

|  |
| --- |
| <StackPanel>  <TextBlock Text="使用 behavior 中的事件触发器 ChangePropertyAction" />  <Separator />  <TextBlock x:Name="tb9" />  <CheckBox x:Name="cb9" Content="修改内容">  <i:Interaction.Triggers>  <i:EventTrigger EventName="Checked">  <i:ChangePropertyAction  PropertyName="Text"  TargetObject="{Binding ElementName=tb9}"  Value="{Binding TextBlockB}" />  </i:EventTrigger>  <i:EventTrigger EventName="Unchecked">  <i:ChangePropertyAction  PropertyName="Text"  TargetObject="{Binding ElementName=tb9}"  Value="{Binding TextBlockA}" />  </i:EventTrigger>  <i:EventTrigger EventName="Loaded">  <i:ChangePropertyAction  PropertyName="Text"  TargetObject="{Binding ElementName=tb9}"  Value="{Binding TextBlockA}" />  </i:EventTrigger>  </i:Interaction.Triggers>  </CheckBox>  </StackPanel> |

|  |
| --- |
| <StackPanel>  <StackPanel.Resources>  <ControlTemplate x:Key="controlTemplate10" TargetType="TextBox">  <TextBlock x:Name="textBlock10">  <VisualStateManager.VisualStateGroups>  <VisualStateGroup>  <VisualState x:Name="visualState1">  <Storyboard>  <StringAnimationUsingKeyFrames Storyboard.TargetName="textBlock10" Storyboard.TargetProperty="Text">  <DiscreteStringKeyFrame KeyTime="0" **Value="111"** />  </StringAnimationUsingKeyFrames>  </Storyboard>  </VisualState>  <VisualState x:Name="visualState2">  <Storyboard>  <StringAnimationUsingKeyFrames Storyboard.TargetName="textBlock10" Storyboard.TargetProperty="Text">  <DiscreteStringKeyFrame KeyTime="0" **Value="222"** />  </StringAnimationUsingKeyFrames>  </Storyboard>  </VisualState>  </VisualStateGroup>  </VisualStateManager.VisualStateGroups>  </TextBlock>  </ControlTemplate>  </StackPanel.Resources>  <TextBlock Text="使用 behavior 中的数据触发器 GoToStateAction" />  <Separator />  <TextBox x:Name="tb10" Template="{StaticResource controlTemplate10}" />  <CheckBox x:Name="cb10" Content="修改内容">  <i:Interaction.Triggers>  <i:DataTrigger Binding="{Binding IsChecked, ElementName=cb10}" Value="True">  <i:GoToStateAction StateName="visualState1" TargetObject="{Binding ElementName=tb10}" />  </i:DataTrigger>  <i:DataTrigger Binding="{Binding IsChecked, ElementName=cb10}" Value="False">  <i:GoToStateAction StateName="visualState2" TargetObject="{Binding ElementName=tb10}" />  </i:DataTrigger>  </i:Interaction.Triggers>  </CheckBox>  </StackPanel> |

|  |
| --- |
| <StackPanel>  <StackPanel.Resources>  <ControlTemplate x:Key="controlTemplate11" TargetType="TextBox">  <TextBlock x:Name="textBlock11">  <i:Interaction.Behaviors>  <i:DataStateBehavior  Binding="{Binding IsChecked, ElementName=cb11}"  FalseState="visualState1"  TrueState="visualState2"  Value="False" />  </i:Interaction.Behaviors>  <VisualStateManager.VisualStateGroups>  <VisualStateGroup>  <VisualState x:Name="visualState1">  <Storyboard>  <StringAnimationUsingKeyFrames Storyboard.TargetName="textBlock11" Storyboard.TargetProperty="Text">  <DiscreteStringKeyFrame KeyTime="0" Value="111" />  </StringAnimationUsingKeyFrames>  </Storyboard>  </VisualState>  <VisualState x:Name="visualState2">  <Storyboard>  <StringAnimationUsingKeyFrames Storyboard.TargetName="textBlock11" Storyboard.TargetProperty="Text">  <DiscreteStringKeyFrame KeyTime="0" Value="222" />  </StringAnimationUsingKeyFrames>  </Storyboard>  </VisualState>  </VisualStateGroup>  </VisualStateManager.VisualStateGroups>  </TextBlock>  </ControlTemplate>  </StackPanel.Resources>  <TextBlock Text="使用 behavior DataStateBehavior" />  <Separator />  <TextBox x:Name="tb11" Template="{StaticResource controlTemplate11}" />  <CheckBox x:Name="cb11" Content="修改内容">  <i:Interaction.Triggers>  <i:EventTrigger EventName="Loaded">  <i:GoToStateAction StateName="visualState2" TargetObject="{Binding ElementName=tb11}" />  </i:EventTrigger>  </i:Interaction.Triggers>  </CheckBox>  </StackPanel> |