WPF Slide Library "SlideView" – Code Overview

Table of Contents

Introduction

This document provides a detailed, human-readable walkthrough of three elements of a WPF project:

1. **ChildItem.xaml** – A user control that displays an individual "slide" (image, icon, and text) with animations.
2. **SlideControl.xaml** – A user control container that holds and animates multiple ChildItem controls in a sliding/carousel-like manner.
3. **A .resx resource file** – A standard .NET resource file with no custom data in this snippet.

ChildItem.xaml Overview

ChildItem Key Elements

1. Root Element & Namespaces

xml

Copy

```
<UserControl x:Class="WPFSlideLibrary.ChildItem"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:y="clr-namespace:WPFSlideLibrary"
        Tag="{Binding ID}">
  ...
</UserControl>
```

- Declares the ChildItem control in the WPFSlideLibrary namespace.
- Sets Tag="{Binding ID}" so the control's Tag property is bound to a data context property ID.

2. Resources (Converters, Styles, Storyboards)

xml

Copy

```
<FrameworkElement.Resources>
 <y:ImageVisibilityConverter x:Key="imageVisConverter"/>
 <y:XamlVisibilityConverter x:Key="xamlVisConverter"/>
 <y:BitmapConverter x:Key="bitmapConverter"/>

 <Style x:Key="MainFont" TargetType="{x:Type TextBlock}">
  <Setter Property="TextBlock.FontFamily" Value="segoeuil.ttf# Segoe UI Light"/>
  <Setter Property="TextBlock.FontWeight" Value="Light"/>
  <Setter Property="TextBlock.Foreground" Value="White"/>
  <Setter Property="TextBlock.FontSize" Value="12"/>
```

```
 </Style>


 <!-- Animations -->
 <Storyboard x:Key="sbShowHighlight">
  <!-- Animates the Opacity of a highlight rectangle -->
 </Storyboard>


 <Storyboard x:Key="sbHideHighlight">
  <!-- Reverses the highlight animation -->
 </Storyboard>


 <Storyboard x:Key="sbActiveItem">
  <!-- Slightly scales the item up for an “active” effect -->
 </Storyboard>
</FrameworkElement.Resources>
```

- **Converters**: Used for controlling visibility and image conversions.
- **MainFont Style**: A simple style that changes the font family, weight, color, and size of TextBlock elements.
- **Storyboards**:
    - sbShowHighlight / sbHideHighlight: Animate a rectangle’s opacity to show or hide a highlight.
    - sbActiveItem: Scales the control when it becomes active (e.g., click or selection).

3. Render Transforms

xml

Copy

```
<UIElement.RenderTransform>
```

```
 <TransformGroup>
  <ScaleTransform x:Name="mainScale" ScaleX="1" ScaleY="1"/>
 </TransformGroup>
</UIElement.RenderTransform>
```

- A ScaleTransform at the root can be used to zoom the entire user control.

4. Main Grid

xml

Copy

```
<Grid MouseEnter="Image_MouseEnter" MouseLeave="Image_MouseLeave"
    x:Name="mainGrid" Width="350" Height="300" RenderTransformOrigin="0.5,0.5">
 <UIElement.RenderTransform>
  <TransformGroup>
   <ScaleTransform x:Name="scale" ScaleX="1" ScaleY="1"/>
  </TransformGroup>
 </UIElement.RenderTransform>
 ...
</Grid>
```

- The main layout container with a defined size (350x300).
- Handles mouse-over events to trigger highlight animations or other behaviors.

5. Background Borders

xml

Copy

```
<Border x:Name="backgroundOn" Margin="0" Background="#cc5e9923"
CornerRadius="10,10,10,10">
 <Image ... Source="Images/fade.jpg" ... />
</Border>
```

```
<Border x:Name="backgroundOff" Margin="0" Background="#CC849176"
CornerRadius="10,10,10,10">
 <Image ... Source="Images/fade.png" ... />
</Border>
```

- Two overlapping Borders with different background images or colors, possibly toggled to show “active vs. inactive” states.

6. Content StackPanel

xml

Copy

```
<StackPanel Margin="10,10,10,10">
 <Grid Height="215">
  <Image x:Name="Image" Stretch="UniformToFill" Source="{Binding Screenshot}" />
 </Grid>
 <StackPanel x:Name="pnlDescription" Orientation="Horizontal" ... Height="58">
  <Image x:Name="ChildIcon" Width="48"
      Source="{Binding Icon, Converter={StaticResource bitmapConverter}}"/>
  <TextBlock FontSize="15" TextWrapping="Wrap" Width="250"
       Style="{StaticResource MainFont}" Foreground="#FF222222"
       Text="{Binding Name}"/>
 </StackPanel>
</StackPanel>
```

- The Image likely displays a screenshot/preview.
- A smaller icon next to a text description ({Binding Name}).

7. Highlight Border

xml

Copy

<Border x:Name="rectangle" Margin="5" IsHitTestVisible="false" Opacity=".5" CornerRadius="10">

<Border.Background>

<RadialGradientBrush GradientOrigin="0.694,0.792">

<!-- Gradient stops for a “halo” highlight effect -->

</RadialGradientBrush>

</Border.Background>

</Border>

- A semi-transparent radial gradient used for highlighting, animated via the storyboards above.

ChildItem Overall Purpose

ChildItem is a self-contained control that displays:

- A preview image, icon, and text label for a “slide” or item.
- Hover, highlight, and active selection animations.
- Custom converters for image and visibility logic.

SlideControl.xaml Overview

SlideControl Key Elements

1. Root Element

xml

Copy

<UserControl x:Class="WPFSlideLibrary.SlideControl"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:slide="clr-namespace:WPFSlideLibrary"

Height="300">

...

</UserControl>

- Declares SlideControl with a fixed height of 300.

2. Resources (Storyboards)

xml

Copy

```
<FrameworkElement.Resources>
 <Storyboard x:Key="sbListPosition">
  <!-- Animates X property of a TranslateTransform for sliding -->
 </Storyboard>

 <Storyboard x:Key="sbListDragPosition">
  <!-- Similar sliding animation, possibly used while dragging -->
 </Storyboard>

 <Storyboard x:Key="sbTierDown" Completed="LeaveTierAnimation_Completed">
  <!-- Scales the control up to 5.0 and fades opacity to 0 -->
 </Storyboard>

 <Storyboard x:Key="sbTierUp" Completed="LeaveTierAnimation_Completed">
  <!-- Scales control down to 0.01, also used for transitions -->
 </Storyboard>

 <Storyboard x:Key="sbTierResetDown" Completed="EnterTierAnimation_Completed">
  <!-- Resets scale from 0.01 to 1.0 -->
 </Storyboard>
```

```
 <Storyboard x:Key="sbTierResetUp" Completed="EnterTierAnimation_Completed">
  <!-- Resets scale from 5.0 back to 1.0 -->
 </Storyboard>
</FrameworkElement.Resources>
```

- **sbListPosition** & **sbListDragPosition**: For horizontal translation of a list panel.
- **Tier Storyboards**: For zooming/scaling the entire control in or out (e.g., transitioning to another "tier" of the UI).

3. Main Grid & ScaleTransform

xml

Copy

```
<Grid x:Name="grid" RenderTransformOrigin="0.5,0.5">
 <UIElement.RenderTransform>
  <ScaleTransform x:Name="scale" ScaleX="1" ScaleY="1"/>
 </UIElement.RenderTransform>
 ...
</Grid>
```

- Similar transform approach to ChildItem, letting the entire control scale up or down.

4. Interaction Grid

xml

Copy

```
<Grid MouseLeftButtonDown="Control_MouseLeftButtonDown"
    MouseLeftButtonUp="Control_MouseLeftButtonUp"
    MouseLeave="Control_MouseLeave"
    MouseMove="Control_MouseMove"
    Margin="0,0,0,-100"
    Background="#00000000"
```

Cursor="Hand"/>

- Captures mouse events to handle dragging or selecting slides.
- The negative top margin pulls the area of interaction upward slightly.

5. ItemsControl for Slides

xml

Copy

```
<ItemsControl x:Name="list">
 <UIElement.RenderTransform>
  <TranslateTransform x:Name="translate" X="0" Y="0"/>
 </UIElement.RenderTransform>
 <ItemsControl.ItemsPanel>
  <ItemsPanelTemplate>
   <slide:SlideStackPanel Loaded="SlideStackPanel_Loaded"/>
  </ItemsPanelTemplate>
 </ItemsControl.ItemsPanel>
 <ItemsControl.ItemTemplate>
  <DataTemplate>
   <Grid Tag="0">
    <slide:ChildItem MouseLeftButtonUp="ChildItem_Click"
                     Width="350"
                     Cursor="Hand"
                     Tag="{Binding ID}"/>
   </Grid>
  </DataTemplate>
 </ItemsControl.ItemTemplate>
</ItemsControl>
```

- **ItemsControl**: Displays a collection of items as ChildItems.
- **SlideStackPanel**: A custom panel class that arranges items horizontally/“sliding” style.
- **TranslateTransform**: Moves the entire set of items left or right, creating a “carousel” effect.

SlideControl Overall Purpose

SlideControl provides a container for multiple ChildItems, giving:

- Horizontal sliding/dragging functionality via mouse events.
- Large-scale transitions (zoom out/in) through storyboards.
- A “carousel” or “slider” user experience for navigating items.