

# **Zeus ImageControls Ver. 1.0**

## **Manual**

**Copyright © 2020 Χρήστος Μουρατίδης**

## Πίνακας περιεχομένων

<b>ZEUS IMAGECONTROLS VERSION 1.0</b> .....	<b>1</b>
<b>Ο ΟΡΙΣΜΟΣ ΤΗΣ ΚΛΑΣΗΣ</b> .....	<b>2</b>
<b>ΔΙΑΝΟΜΗ</b> .....	<b>3</b>
<b>ΕΠΙΚΟΙΝΩΝΙΑ</b> .....	<b>4</b>
<b>SLIDESHOW</b> .....	<b>5</b>
ΑΠΑΡΙΘΜΗΣΕΙΣ .....	18
<i>SlidingEffectEnum</i> .....	19
ΙΔΙΟΤΗΤΕΣ .....	21
<i>Caption</i> .....	24
<i>CaptionBorderStyle</i> .....	28
<i>CaptionStyle</i> .....	30
<i>CurrentImageIndex</i> .....	32
<i>ImageInfoList</i> .....	34
<i>Looping</i> .....	36
<i>NavigationAreaBorderStyle</i> .....	37
<i>NavigationText</i> .....	39
<i>NavigationTextStyle</i> .....	41
<i>NextButtonInSpaceStyle</i> .....	43
<i>NextButtonSmallStyle</i> .....	46
<i>PreviousButtonInSpaceStyle</i> .....	49
<i>PreviousButtonSmallStyle</i> .....	52
<i>ShowButtonsInSpace</i> .....	55
<i>ShowButtonsSmall</i> .....	57
<i>ShowCaption</i> .....	59
<i>ShowNavigationText</i> .....	61
<i>ShowToolTipsForPreviousNextButtons</i> .....	63
<i>SlideDuration</i> .....	64
<i>SlidingEffect</i> .....	65
<i>SlidingEffectDuration</i> .....	66
<i>SlidingMoveInEffectOffset</i> .....	67
<i>ToolTipTextForNextButton</i> .....	69
<i>ToolTipTextForPreviousButton</i> .....	71
<i>UseSameSlidingEffectForAllImages</i> .....	73
ΣΥΜΒΑΝΤΑ .....	75
<i>ImageChanged</i> .....	76
<i>SlideShowPaused</i> .....	78
<i>SlideShowResumed</i> .....	80
<i>SlideShowStarted</i> .....	82
ΜΕΘΟΔΟΙ .....	86
<i>GotoSlide(nbrSlide)</i> .....	87
<i>GotoSlide(slideCaption)</i> .....	88
<i>PauseSlideShow</i> .....	89
<i>ResumeSlideShow</i> .....	90
<i>StartSlideShow</i> .....	91
<i>StopSlideShow</i> .....	92
STYLES ΚΑΙ TEMPLATES .....	93
<i>Parts και States</i> .....	94

<i>To default Style και ControlTemplate .....</i>	<i>95</i>
<i>Ένα προσαρμοσμένο Style και ControlTemplate .....</i>	<i>107</i>
ΠΑΡΑΡΤΗΜΑ - ΚΛΑΣΕΙΣ .....	119
<i>ImageInfo.....</i>	<i>120</i>

## Zeus ImageControls version 1.0



**Κλάσεις:** SlideShow

**Inherits:** System.Windows.Controls

**Namespace:** Zeus.WPF.Controls.ImageControls


**Assembly:** ZeusImageControls (in ZeusImageControls.dll)

**Dependencies:** -

### Περιγραφή

Η βιβλιοθήκη **Zeus ImageControls** περιλαμβάνει, προς το παρόν, το **SlideShow** control που **επιτρέπει να εμφανίζεται διαδοχικά μία ακολουθία εικόνων**. Η αλληλουχία αυτή μπορεί να είναι αυτόματη (auto-play slideshow) ή/και να ελέγχεται από τον χρήστη με τα αντίστοιχα buttons μπρος-πίσω. Να σημειώσουμε ότι, κάθε εικόνα μπορεί να συνοδεύεται από το δικό της caption και εφέ εισόδου. Μπορούμε να ορίσουμε να χρησιμοποιηθεί ένα γενικό εφέ εισόδου που θα χρησιμοποιηθεί σε όλες τις εικόνες ή να χρησιμοποιηθεί το εφέ κάθε εικόνας.

Παρακάτω, παρατίθενται τα **ImageControls**.

	<u>SlideShow</u>	Εμφανίζει διαδοχικά (αυτόματα ή/και χειροκίνητα μπρος-πίσω) μία σειρά εικόνων με caption και εφέ εισόδου.
---	------------------	---

## Ο ορισμός της κλάσης

Η κλάση έχει οριστεί ως εξής:

---

- Για την κλάση **SlideShow**:

**Σύνταξη:**

**VB:**

```
<TemplateVisualState(Name="Normal", GroupName="CommonStates"),  
TemplateVisualState(Name="MouseOver", GroupName="CommonStates"),  
TemplateVisualState(Name="Disabled", GroupName="CommonStates"),  
TemplateVisualState(Name="Focused", GroupName="FocusStates"),  
TemplateVisualState(Name="Unfocused", GroupName="FocusStates"),  
TemplatePart(Name="PART_Previous_InSpace", Type:=GetType(Button)),  
TemplatePart(Name="PART_Next_InSpace", Type:=GetType(Button)),  
TemplatePart(Name="PART_Previous_Small", Type:=GetType(Button)),  
TemplatePart(Name="PART_Next_Small", Type:=GetType(Button)),  
TemplatePart(Name="PART_Caption", Type:=GetType(TextBlock)),  
TemplatePart(Name="PART_NavigationText", Type:=GetType(TextBlock)),  
TemplatePart(Name="PART_Image", Type:=GetType(Image))>  
Public Class SlideShow  
    Inherits Control
```

**XAML Object Element Usage:**

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.ImageControls;assembly=ZeusImageControl  
s"
```

Χρήση:

```
<zeus:SlideShow ... />
```

---

## Διανομή

Κατά τη διανομή, στο φάκελο της εφαρμογής σας πρέπει να αντιγράψετε το **assembly αρχείο ZeusImageControls.dll**.

## Επικοινωνία

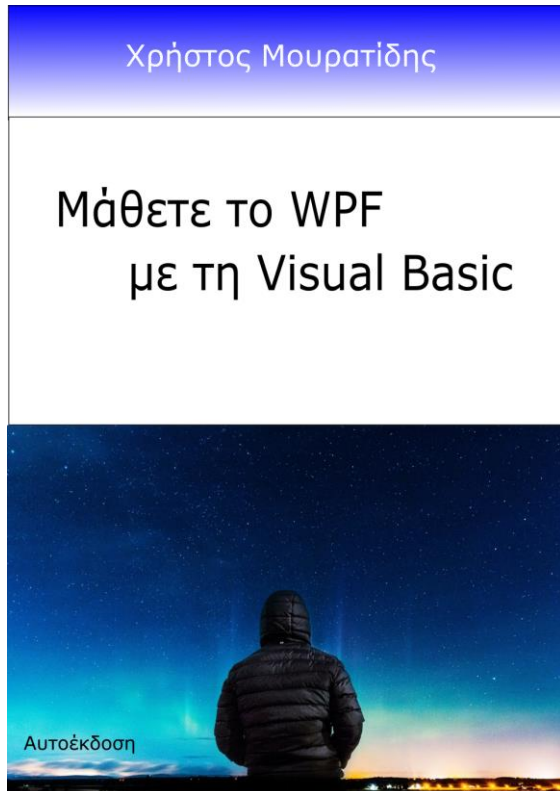
Για οποιαδήποτε πληροφορία ή διευκρίνιση παρακαλώ επικοινωνήστε στο :

[mouratx@yahoo.com](mailto:mouratx@yahoo.com) ή [mouratx@hotmail.com](mailto:mouratx@hotmail.com)



Χρήστος Μουρατίδης,  
Πειραιάς, Ιανουάριος 2020

Υ.Γ. Μπορείτε να επικοινωνήσετε μαζί μου για να προμηθευτείτε το **βιβλίο** μου  
"Μάθετε το WPF με τη Visual Basic" (1.333 σελίδες, Αυτοέκδοση 2018).





## SlideShow

Ένα **Control** που εμφανίζει διαδοχικά (αυτόματα ή/και χειροκίνητα μπρος-πίσω) μία σειρά εικόνων με **caption** και **εφέ εισόδου**.



### Σύνταξη:

#### VB:

```
<TemplateVisualState(Name:="Normal", GroupName:="CommonStates"),
TemplateVisualState(Name:="MouseOver", GroupName:="CommonStates"),
TemplateVisualState(Name:="Disabled", GroupName:="CommonStates"),
TemplateVisualState(Name:="Focused", GroupName:="FocusStates"),
TemplateVisualState(Name:="Unfocused", GroupName:="FocusStates"),
TemplatePart(Name:="PART_Previous_InSpace", Type:=GetType(Button)),
TemplatePart(Name:="PART_Next_InSpace", Type:=GetType(Button)),
TemplatePart(Name:="PART_Previous_Small", Type:=GetType(Button)),
TemplatePart(Name:="PART_Next_Small", Type:=GetType(Button)),
TemplatePart(Name:="PART_Caption", Type:=GetType(TextBlock)),
TemplatePart(Name:="PART_NavigationText", Type:=GetType(TextBlock)),
TemplatePart(Name:="PART_Image", Type:=GetType(Image))>
Public Class SlideShow
    Inherits Control
```

#### XAML Object Element Usage:

Εισαγωγή namespace:

```
xmlns:zeus="clr-namespace:Zeus.WPF.Controls.ImageControls;assembly=ZeusImageControls"
```



Χρήση:

```
<zeus:SlideShow ... />
```

## Παράδειγμα:

Στο παράδειγμα που ακολουθεί, τοποθετούμε το SlidesShow control στο πάνω μέρος του παραθύρου. Πιο κάτω, έχουμε βάλει μέσα σε ένα GroupBox ρυθμίσεις για το auto-play και πιο κάτω ένα άλλο GroupBox με ρυθμίσεις για το UI του SlideShow control.

Να σημειώσουμε ότι στα Window.Resources έχουμε ορίσει ένα Style για το caption (λεζάντα). Στον υποφάκελο Images της εφαρμογής υπάρχουν τα αρχεία εικόνων. Η ρουτίνα CreateListOfImagesWithSpecificEffectForEveryImage δημιουργεί τη λίστα των εικόνων με πληροφορίες για το απόλυτο path, το caption και το εφέ της κάθε εικόνας.

## XAML:

```
<Window x:Class="MainWindow"
  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"
  xmlns:zeus="clr-namespace:Zeus.WPF.Controls.ImageControls;assembly=ZeusImageControls"
  mc:Ignorable="d"
  Title="Image Controls Sample Project" Height="500.52" Width="425.78"
  WindowStartupLocation="CenterScreen" Loaded="Window_Loaded" >

  <Window.Resources>

    <Style x:Key="SlideShowCaptionStyle" TargetType="{x:Type TextBlock}">
      <Setter Property="FontFamily" Value="Calibri"/>
      <Setter Property="HorizontalAlignment" Value="Center" />
      <Setter Property="Foreground" Value="Aqua" />
    </Style>

    <!-- Στυλ για τα auto-play buttons -->
    <Style x:Key="AutoPlayButtonStyle" TargetType="{x:Type Button}">
      <Setter Property="FontFamily" Value="Calibri"/>
      <Setter Property="Padding" Value="5"/>
      <Style.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
          <Setter Property="Foreground" Value="Blue"/>
        </Trigger>
      </Style.Triggers>
    </Style>

  </Window.Resources>

  <Grid>

    <Grid.RowDefinitions >
      <RowDefinition Height="Auto"/>
      <RowDefinition />
      <RowDefinition Height="Auto" />
    </Grid.RowDefinitions>
```

```
<!-- Header -->
<TextBlock Text="SlideShow" FontFamily="Monotype Corsiva"
           FontSize="26" FontWeight="Bold" Foreground="Blue" Margin="5,10,5,5"/>

<!-- SlideShow content-->
<zeus:SlideShow Name="slideShow" Grid.Row="1" Background="#78909C"
                SlidingEffect="FadeIn"
                CaptionStyle="{StaticResource SlideShowCaptionStyle }"
                ImageChanged="slideShow_ImageChanged" />

<!-- Buttons -->
<StackPanel Grid.Row="2" Margin="0,10,0,0">

    <GroupBox Header="Auto play" Padding="8" >

        <StackPanel Orientation="Horizontal" ButtonBase.Click="btnAutoPlay_Click">
            <Button Name="btnStartSlideShow" Content="Start"
                  Style="{StaticResource AutoPlayButtonStyle }"
                  ToolTip="Ξεκινάει από την αρχή"/>

            <Button Name="btnStopSlideShow" Content="Stop/Pause" Margin="5,0,0,0"
                  Style="{StaticResource AutoPlayButtonStyle }"
                  ToolTip="Σταματάει/Διακόπτει προσωρινά το slideshow"/>

            <Button Name="btnResumeSlideShow" Content="Resume" Margin="5,0,0,0"
                  Style="{StaticResource AutoPlayButtonStyle }"
                  ToolTip="Συνεχίζει το slideshow"/>

            <StackPanel Orientation="Horizontal" Margin="15,0,0,0"
                      VerticalAlignment="Center" >
                <TextBlock Text="Slide duration in secs: "/>
                <TextBox Name="txtImageDuration" Text="5"
                       HorizontalContentAlignment="Right" Width="30"
                       Language="el"
                       TextChanged="txtImageDuration_TextChanged"/>
            </StackPanel>
        </StackPanel>

    </GroupBox>

    <GroupBox Header="UI Settings" Padding="8"
            ToggleButton.Checked="GroupBoxUISettings_Checked"
            ToggleButton.Unchecked ="GroupBoxUISettings_Unchecked">

        <Grid>
            <Grid.RowDefinitions >
                <RowDefinition Height="20"/>
                <RowDefinition Height="20"/>
                <RowDefinition Height="20"/>
                <RowDefinition Height="20"/>
            </Grid.RowDefinitions>

            <Grid.ColumnDefinitions >
                <ColumnDefinition />
                <ColumnDefinition />
            </Grid.ColumnDefinitions>

            <CheckBox Name="chkInSpaceButtons"
                    Content="In-space navigation buttons" IsChecked="True"/>

            <CheckBox Name="chkSmallButtons"
                    Content="Small navigation buttons" IsChecked="True"/>
        </Grid>
    </GroupBox>
</StackPanel>
```

```

        Content="Small navigation buttons" IsChecked="True"
        Grid.Column="1"/>

<CheckBox Name="chkCaption" Content="Caption" IsChecked="True"
        Grid.Row="1"/>

<CheckBox Name="chkNavigationText"
        Content="Navigation text" IsChecked="True"
        Grid.Row="1" Grid.Column="1"/>

<CheckBox Name="chkLooping" Content="Looping" IsChecked="True"
        Grid.Row="2" />

<CheckBox Name="chkSameEffectForAllImages"
        Content="Same effect for all images" IsChecked="True"
        Grid.Row="3" />

<StackPanel Orientation="Horizontal" Grid.Row="3" Grid.Column="1" >

    <TextBlock Text="Effect duration in secs: "/>

    <TextBox Name="txtEffectDuration" Text="0,3"
        HorizontalContentAlignment="Right" Width="30"
        Language="el"
        TextChanged="txtEffectDuration_TextChanged"/>

</StackPanel>

</Grid>

</GroupBox>

</StackPanel>

</Grid>
</Window>

```

**VB:**

```
Imports Zeus.WPF.Controls.ImageControls
```

```
Class MainWindow
```

```
'ο τρέχον φάκελος της εφαρμογής.
```

```
Dim AppPath As String = System.AppDomain.CurrentDomain.BaseDirectory
```

```
'Σημαία, αν έχει σταματήσει το auto-play slideshow.
```

```
Dim isPaused As Boolean = False
```

```
#Region "Event handlers"
```

```
Private Sub Window_Loaded(sender As Object, e As RoutedEventArgs)
```

```
'Γεμίζουμε τη λίστα των εικόνων.
```

```
slideShow.ImageInfoList = _
    CreateListofImagesWithSpecificEffectForEveryImage()
```

```
End Sub
```

'Εδώ ενημερώνουμε το navigation text που εμφανίζεται στο κάτω δεξί μέρος.

```
Private Sub slideShow_ImageChanged(sender As Object, e As RoutedEventArgs)

    Dim slideShow As SlideShow = TryCast(sender, SlideShow)

    If slideShow IsNot Nothing Then

        With slideShow
            .NavigationText = String.Concat(If(.ImageInfoList.Count > 0, _
                .CurrentImageIndex + 1, 0), "/", .ImageInfoList.Count)
        End With

    End If

End Sub
```

'Start, Stop/Pause, Resume.

```
Private Sub btnAutoPlay_Click(sender As Object, e As RoutedEventArgs)

    Dim btn As Button = TryCast(e.OriginalSource, Button)

    If btn IsNot Nothing Then

        If btn Is btnStartSlideShow Then
            slideShow.StartSlideShow() : isPaused = False
        ElseIf btn Is btnStopSlideShow Then
            slideShow.PauseSlideShow() : isPaused = True
        ElseIf btn Is btnResumeSlideShow Then
            If isPaused Then
                slideShow.ResumeSlideShow() : isPaused = False
            End If
        End If

    End If

End Sub
```

'UI settings Checked for CheckButtons.

```
Private Sub GroupBoxUISettings_Checked(sender As Object, e As RoutedEventArgs)

    Dim chk As CheckBox = TryCast(e.OriginalSource, CheckBox)

    If chk IsNot Nothing Then

        If chk Is chkInSpaceButtons Then
            slideShow.ShowButtonsInSpace = True
        ElseIf chk Is chkSmallButtons Then
            slideShow.ShowButtonsSmall = True
        ElseIf chk Is chkCaption Then
            slideShow.ShowCaption = True
        ElseIf chk Is chkNavigationText Then
            slideShow.ShowNavigationText = True
        ElseIf chk Is chkLooping Then
            slideShow.Looping = True
        End If

    End If

End Sub
```

```
        ElseIf chk Is chkSameEffectForAllImages Then
            slideShow.UseSameSlidingEffectForAllImages = True
        End If

    End If

End Sub

'UI settings Unchecked for CheckButtons.
Private Sub GroupBoxUISettings_Unchecked(sender As Object, e As RoutedEventArgs)

    Dim chk As CheckBox = TryCast(e.OriginalSource, CheckBox)

    If chk IsNot Nothing Then

        If chk Is chkInSpaceButtons Then
            slideShow.ShowButtonsInSpace = False

        ElseIf chk Is chkSmallButtons Then
            slideShow.ShowButtonsSmall = False

        ElseIf chk Is chkCaption Then
            slideShow.ShowCaption = False

        ElseIf chk Is chkNavigationText Then
            slideShow.ShowNavigationText = False

        ElseIf chk Is chkLooping Then
            slideShow.Looping = False

        ElseIf chk Is chkSameEffectForAllImages Then
            slideShow.UseSameSlidingEffectForAllImages = False
        End If

    End If

End Sub

'Αλλαγή slide duration in secs.
Private Sub txtImageDuration_TextChanged(sender As Object, _
                                         e As TextChangedEventArgs)

    Dim txtImageDuration As TextBox = TryCast(sender, TextBox)

    If txtImageDuration IsNot Nothing Then

        Try
            Dim secs As Single = CSng(txtImageDuration.Text)

            If secs > 0 Then
                slideShow.SlideDuration = TimeSpan.FromSeconds(secs)
            End If
        Catch
        End Try

    End If

End Sub

'Αλλαγή effect duration in secs.
Private Sub txtEffectDuration_TextChanged(sender As Object, _
                                         e As TextChangedEventArgs)
```

```
Dim txtEffectDuration As TextBox = TryCast(sender, TextBox)

If txtEffectDuration IsNot Nothing Then

    Try
        Dim secs As Single = CSng(txtEffectDuration.Text)

        If secs > 0 Then
            slideShow.SlidingEffectDuration = TimeSpan.FromSeconds(secs)
        End If
    Catch
    End Try

End If

End Sub

#End Region

#Region "Private methods"

'Δημιουργεί τη λίστα των slides. Κάθε slide περιέχει το απόλυτο path της εικόνας,
'το caption και το εφέ εισόδου.
Private Function CreateListOfImagesWithSpecificeffectForEveryImage()
    As List(Of ImageInfo)

    Dim imgList As New List(Of ImageInfo)

    With imgList

        .Add(New ImageInfo(AppPath & "Images\Photo-01.jpg", "Photo-01", _
            SlideShow.SlidingEffectEnum.FadeIn))

        .Add(New ImageInfo(AppPath & "Images\Photo-02.jpg", "Photo-02", _
            SlideShow.SlidingEffectEnum.MoveInFromLeft))

        .Add(New ImageInfo(AppPath & "Images\Αγριοβοτανο-1.jpg", "Αγριοβοτανο-1", _
            SlideShow.SlidingEffectEnum.MoveInFromTopLeft))

        .Add(New ImageInfo(AppPath & "Images\Αγριοβοτανο-2.jpg", "Αγριοβοτανο-2", _
            SlideShow.SlidingEffectEnum.ZoomInFromMiddle))

        .Add(New ImageInfo(AppPath & "Images\Άνδρος.jpg", "Άνδρος", _
            SlideShow.SlidingEffectEnum.ZoomInFromTopLeft))

        .Add(New ImageInfo(AppPath & "Images\Ελληνικά.jpg", "Ελληνικά", _
            SlideShow.SlidingEffectEnum.ZoomInFromBottomRight))

    End With

    Return imgList


End Function

#End Region

End Class
```

Στο παράρτημα μπορούμε να δούμε τον ορισμό της κλάσης ImageInfo.

Στις παρακάτω εικόνες βλέπουμε διάφορα στιγμιότυπα, δείγμα της ευελιξίας του control όπου μπορούμε να ρυθμίσουμε διάφορα elements του UI.



SlideShow

Photo-02 2/6

Auto play

Start Stop/Pause Resume Slide duration in secs: 5

UI Settings

In-space navigation buttons  Small navigation buttons

Caption  Navigation text

Looping

Same effect for all images Effect duration in secs: 0,3

ShowButtonsInSpace = True  
ShowCaption = True  
ShowButtonsSmall = True  
ShowNavigationText = True

Όλα τα elements είναι ενεργοποιημένα.

Image Controls Sample Project

*SlideShow*

Photo-02

Auto play

Start Stop/Pause Resume Slide duration in secs: 5

UI Settings

In-space navigation buttons  Small navigation buttons

Caption  Navigation text

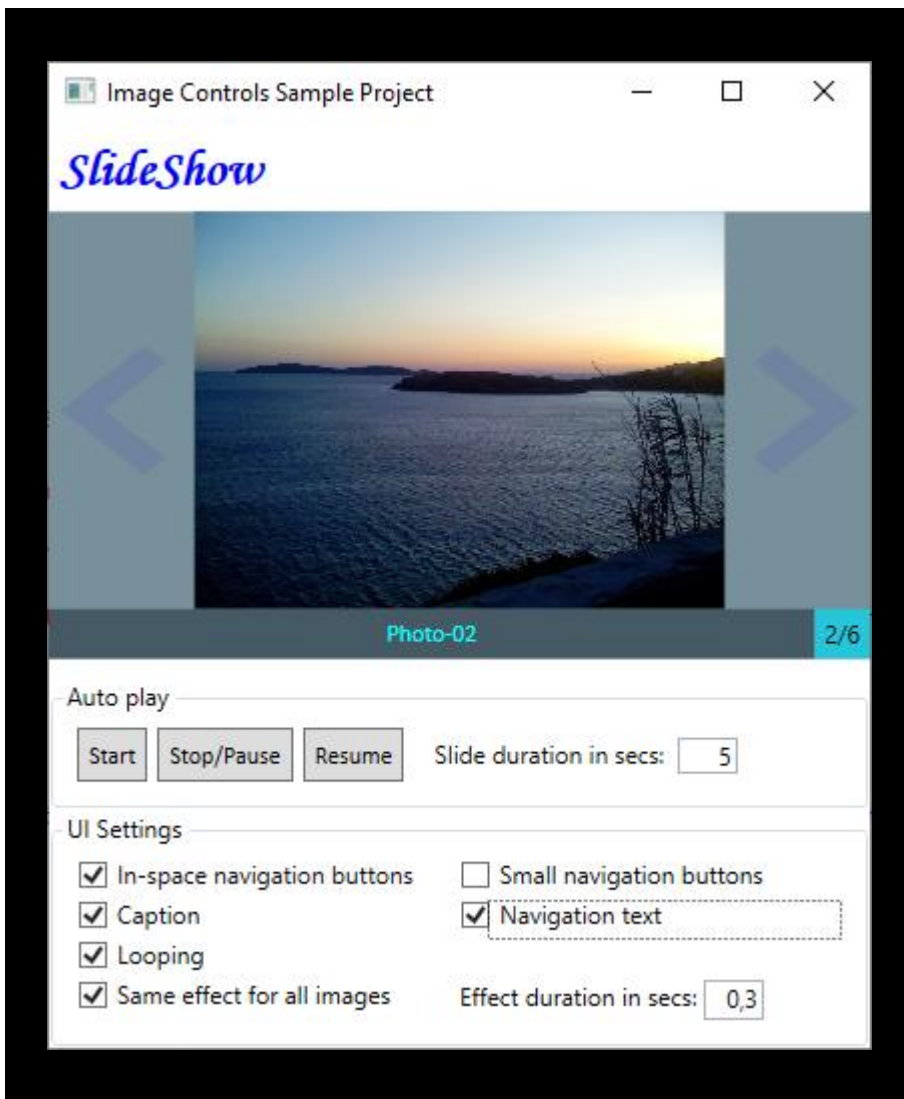
Looping

Same effect for all images Effect duration in secs: 0,3

ShowButtonsInSpace = True  
 ShowCaption = True  
 ShowButtonsSmall = False  
 ShowNavigationText = False

Η navigation area στο  
 κάτω-δεξι μέρος έχει  
 αποκρυβεί.





ShowButtonsInSpace = True  
 ShowCaption = True  
 ShowButtonsSmall = False  
 ShowNavigationText = True

Στη navigation area στο κάτω-δεξί μέρος εμφανίζεται μόνο το navigation text (ενημερωτικό κείμενο πλοήγησης).




Image Controls Sample Project

*SlideShow*

Auto play

Start Stop/Pause Resume Slide duration in secs: 5

UI Settings

In-space navigation buttons  Small navigation buttons

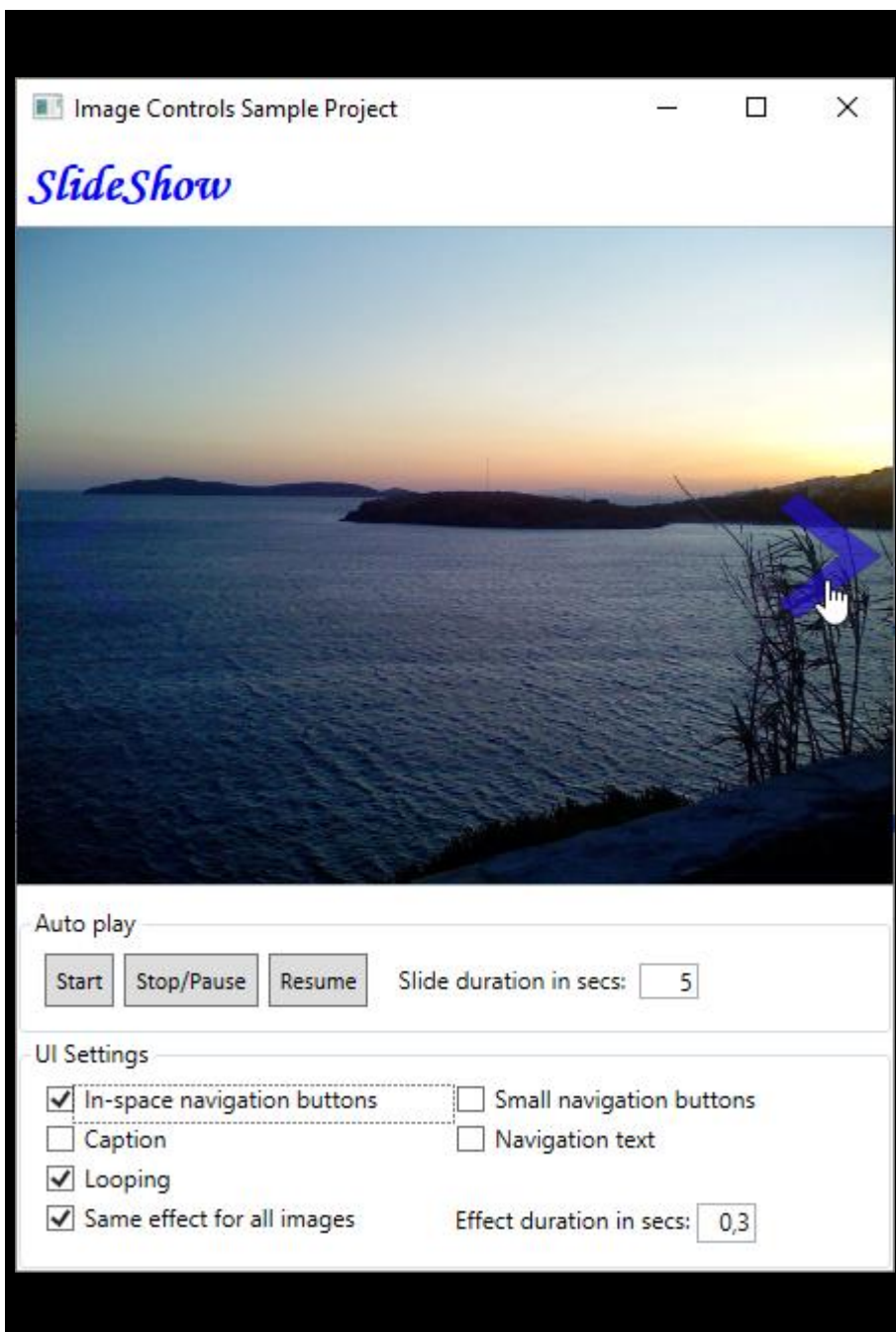
Caption  Navigation text

Looping

Same effect for all images Effect duration in secs: 0,3

ShowButtonsInSpace = True  
ShowCaption = False  
ShowButtonsSmall = False  
ShowNavigationText = False

Η navigation area στο  
κάτω-δεξί μέρος καθώς και η  
λεζάντα (caption) έχει  
αποκρυβεί.



```
ShowButtonsInSpace = True  
ShowCaption = False  
ShowButtonsSmall = False  
ShowNavigationText = False
```

Προσαρμόζοντας το παράθυρο. Όταν πάμε το ποντίκι πάνω στο in-space button (εδώ στο "Next") αυτό εμφανίζεται πιο έντονο και ο δείκτης γίνεται "χεράκι".

Αυτή η ρύθμιση είναι κατάλληλη και για auto-play slideshow.

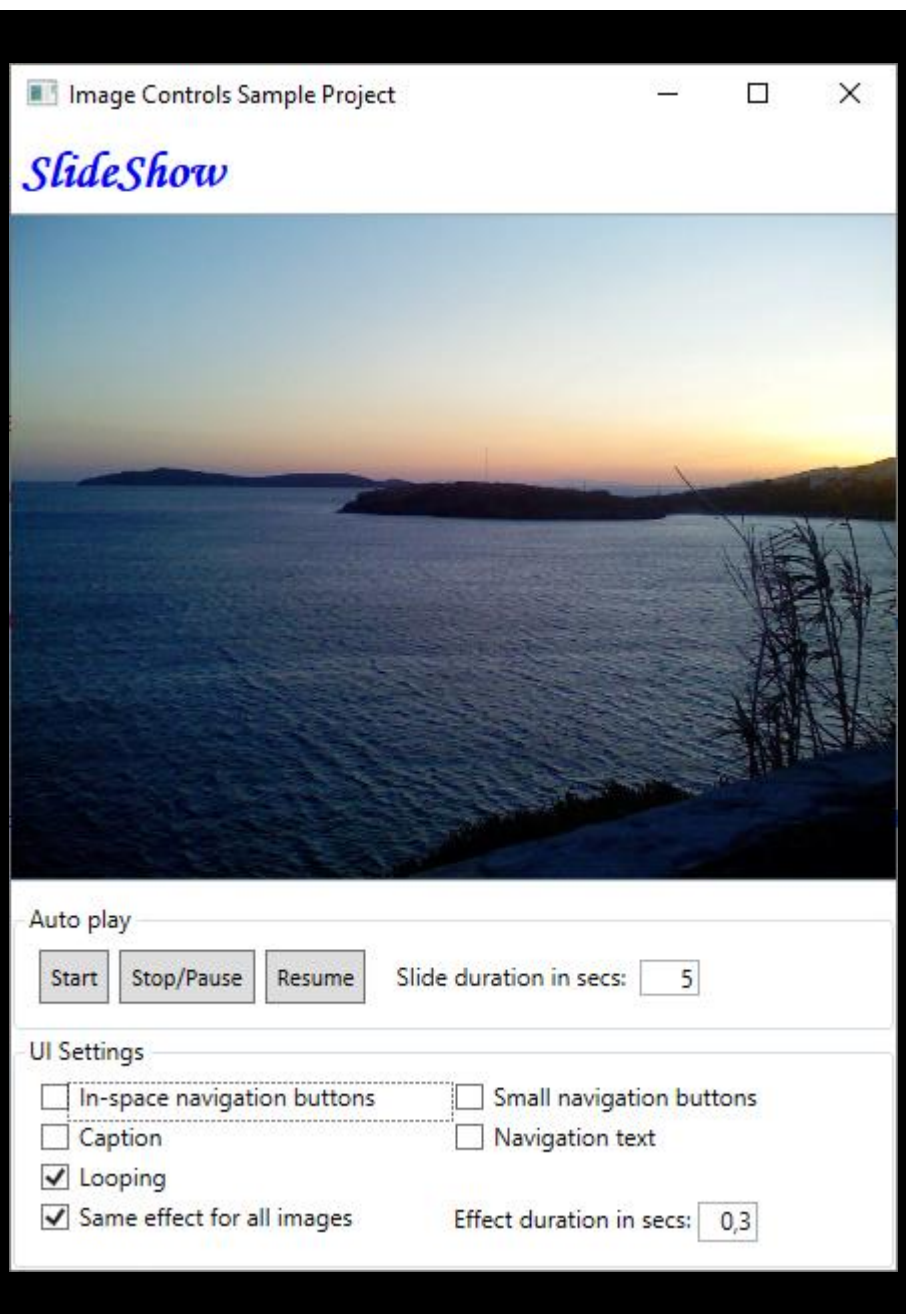


Image Controls Sample Project

*SlideShow*

Auto play

Start Stop/Pause Resume Slide duration in secs: 5

UI Settings

In-space navigation buttons  Small navigation buttons

Caption  Navigation text

Looping

Same effect for all images Effect duration in secs: 0,3

ShowButtonsInSpace = False  
ShowCaption = False  
ShowButtonsSmall = False  
ShowNavigationText = False

Όλα τα elements είναι απενεργοποιημένα.

Το παράθυρο έχει προσαρμοστεί κατάλληλα.

Αυτή η ρύθμιση είναι κατάλληλη ειδικά για auto-play slideshow καθώς δεν υπάρχει δυνατότητα από τον χρήστη να πλοηγηθεί χειροκίνητα.

### Παρατήρηση:

Η μετάβαση στην **προηγούμενη/επόμενη εικόνα** γίνεται αντίστοιχα με τα **shortcut keys CTRL + left arrow / CTRL + right arrow**, εφόσον το control έχει την εστίαση (focus).

## Απαριθμήσεις

Όνομα	Περιγραφή
<a href="#">SlidingEffectEnum</a>	Περιλαμβάνει τις τιμές για τα εφέ εικόνων (sliding effects).

## SlidingEffectEnum

Περιλαμβάνει τις τιμές για τα εφέ εικόνων (sliding effects).

**Σύνταξη:**

**VB:**

```
Public Enum SlidingEffectEnum
```

**Μέλη:**

Τιμή	Περιγραφή
None	Κανένα εφέ.
FadeIn	Σταδιακή εμφάνιση.
MoveInFromLeft	Είσοδος από αριστερά.
MoveInFromTopLeft	Είσοδος από πάνω-αριστερά.
MoveInFromTop	Είσοδος από πάνω.
MoveInFromTopRight	Είσοδος από πάνω-δεξιά.
MoveInFromRight	Είσοδος από δεξιά.
MoveInFromBottomRight	Είσοδος από κάτω-δεξιά.
MoveInFromBottom	Είσοδος από κάτω.
MoveInFromBottomLeft	Είσοδος από κάτω-αριστερά.

---

ZoomInFromMiddle	Μεγέθυνση από το μέσον.
ZoomInFromLeft	Μεγέθυνση από αριστερά.
ZoomInFromTopLeft	Μεγέθυνση από πάνω-αριστερά.
ZoomInFromTop	Μεγέθυνση από πάνω.
ZoomInFromTopRight	Μεγέθυνση από πάνω-δεξιά.
ZoomInFromRight	Μεγέθυνση από δεξιά.
ZoomInFromBottomRight	Μεγέθυνση από κάτω-δεξιά.
ZoomInFromBottom	Μεγέθυνση από κάτω.
ZoomInFromBottomLeft	Μεγέθυνση από κάτω-αριστερά.

**Παρατηρήσεις:**

Να σημειώσουμε ότι, στα MoveInXXX εφέ η default offset αρχική τοποθέτηση είναι στα 100 pixels. Αυτό μπορεί να αλλάξει μέσω της ιδιότητας SlidingMoveInEffectOffset. Επίσης, η default διάρκεια των εφέ είναι 0,3 secs αλλά μπορεί να αλλάξει μέσω της ιδιότητας SlidingEffectDuration.

## Ιδιότητες

Όνομα	Περιγραφή
<a href="#">Caption</a>	Καθορίζει τη λεζάντα της εικόνας, τύπου <b>String</b> .
<a href="#">CaptionBorderStyle</a>	Καθορίζει το <b>στυλ</b> για το περίγραμμα της λεζάντας, τύπου <b>Style</b> .
<a href="#">CaptionStyle</a>	Καθορίζει το <b>στυλ</b> του κειμένου της λεζάντας, τύπου <b>Style</b> .
<a href="#">CurrentImageIndex</a>	Επιστρέφει το <b>index</b> της τρέχουσας εικόνας στη λίστα των <b>εικόνων</b> . Είναι τύπου <b>Integer</b> .  Το index ξεκινάει από το 0.
<a href="#">ImageInfoList</a>	Καθορίζει τη <b>λίστα</b> των εικόνων, τύπου <b>ImageInfo</b> .
<a href="#">Looping</a>	Καθορίζει <b>αν θα επαναλαμβάνεται το slideshow από την αρχή</b> (είτε στο auto-play είτε χειροκίνητα μέσω των buttons previous/next). Είναι τύπου <b>Boolean</b> .
<a href="#">NavigationAreaBorderStyle</a>	Καθορίζει το <b>στυλ</b> για το περίγραμμα της περιοχής πλοήγησης (στο κάτω δεξί μέρος), τύπου <b>Style</b> .
<a href="#">NavigationText</a>	Καθορίζει το <b>κείμενο</b> πλοήγησης (π.χ. 1 από 10 ή 1/10), τύπου <b>String</b> .  Συνήθως, εδώ τοποθετούμε ενημερωτικό κείμενο για το σημείο στο οποίο βρίσκεται η πλοήγηση στις εικόνες του slideshow. Η τοποθέτηση γίνεται εκμεταλλευόμενοι το συμβάν ImageChanged.
<a href="#">NavigationTextStyle</a>	Καθορίζει το <b>στυλ</b> για το κείμενο της πλοήγησης, τύπου <b>Style</b> .
<a href="#">NextButtonInSpaceStyle</a>	Καθορίζει το <b>στυλ</b> για το <b>button "Next"</b> που βρίσκεται στον χώρο της εικόνας (in-space). Είναι τύπου <b>Style</b> .



<a href="#">NextButtonSmallStyle</a>	Καθορίζει το <b>στυλ</b> για το μικρό <b>button "Next"</b> που βρίσκεται στον χώρο της πλοήγησης. Είναι τύπου <b>Style</b> .
<a href="#">PreviousButtonInSpaceStyle</a>	Καθορίζει το <b>στυλ</b> για το <b>button "Previous"</b> που βρίσκεται στον χώρο της εικόνας ( <b>in-space</b> ). Είναι τύπου <b>Style</b> .
<a href="#">PreviousButtonSmallStyle</a>	Καθορίζει το <b>στυλ</b> για το μικρό <b>button "Previous"</b> που βρίσκεται στον χώρο της πλοήγησης. Είναι τύπου <b>Style</b> .
<a href="#">ShowButtonsInSpace</a>	Καθορίζει <b>αν θα εμφανίζονται τα buttons "Previous/Next"</b> στον χώρο της εικόνας ( <b>in-space</b> ). Είναι τύπου <b>Boolean</b> .
<a href="#">ShowButtonsSmall</a>	Καθορίζει <b>αν θα εμφανίζονται τα μικρά buttons "Previous/Next"</b> στον χώρο της πλοήγησης. Είναι τύπου <b>Boolean</b> .
<a href="#">ShowCaption</a>	Καθορίζει <b>αν θα εμφανίζεται η λεζάντα της εικόνας</b> . Είναι τύπου <b>Boolean</b> .
<a href="#">ShowNavigationText</a>	Καθορίζει <b>αν θα εμφανίζεται το κείμενο της πλοήγησης (π.χ. 1/10)</b> στον χώρο της πλοήγησης. Είναι τύπου <b>Boolean</b> .
<a href="#">ShowToolTipsForPreviousNextButtons</a>	Καθορίζει <b>αν θα εμφανίζονται τα tooltips για τα buttons "Previous/Next"</b> . Είναι τύπου <b>Boolean</b> .
<a href="#">SlideDuration</a>	Καθορίζει τη <b>διάρκεια παραμονής της κάθε εικόνας στην οθόνη, σε ένα auto-play slideshow</b> . Είναι τύπου <b>Duration</b> .
<a href="#">SlidingEffect</a>	Καθορίζει το <b>γενικό εφέ εισόδου για όλες τις εικόνες της λίστας</b> . Είναι τύπου <b>SlidingEffectEnum</b> .  Το εφέ αυτό θα χρησιμοποιηθεί μόνο όταν η ιδιότητα <b>UseSameSlidingEffectForAllImages</b> είναι <b>True</b> . Αλλιώς, θα χρησιμοποιηθεί το κάθε εφέ της εικόνας, όπως το έχουμε προσδιορίσει κατά την εισαγωγή της στη λίστα των εικόνων.
<a href="#">SlidingEffectDuration</a>	Καθορίζει τη <b>διάρκεια του εφέ εισόδου</b> . Είναι τύπου <b>Duration</b> .

<a href="#">SlidingMoveInEffectOffset</a>	<p>Καθορίζει την αρχική offset μετακίνηση της εικόνας σε σχέση με την αρχική της θέση.</p> <p>Π.χ. μία τιμή 100 σημαίνει 100 pixels offset κατά X ή/και Y, ανάλογα με το εφέ μετακίνησης που έχουμε επιλέξει. Είναι τύπου <b>Double</b>.</p> <p>Η τιμή της ιδιότητας αυτής επιδρά πάνω σε όλα τα <b>MoveInXXX</b> εφέ.</p>
<a href="#">ToolTipTextForNextButton</a>	<p>Καθορίζει το κείμενο του tooltip για τα buttons "Next". Είναι τύπου <b>String</b>.</p>
<a href="#">ToolTipTextForPreviousButton</a>	<p>Καθορίζει το κείμενο του tooltip για τα buttons "Previous". Είναι τύπου <b>String</b>.</p>
<a href="#">UseSameSlidingEffectForAllImages</a>	<p>Καθορίζει αν θα χρησιμοποιηθεί το ίδιο εφέ, για όλες τις εικόνες, όπως έχει προσδιοριστεί στην ιδιότητα <b>SlidingEffect</b>. Είναι τύπου <b>Boolean</b>.</p>

## Caption

Καθορίζει το κείμενο για τη λεζάντα της τρέχουσας εικόνας που εμφανίζεται, τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property Caption As String
```

#### Τύπος: **System.String**

Προσδιορίζουμε μία τιμή String που θα αποτελεί τη λεζάντα στην τρέχουσα εικόνα που εμφανίζεται. Η default τιμή είναι "Caption".

### Dependency Property Information:

**Identifier field:** CaptionProperty

### Παρατηρήσεις:

Το caption της κάθε εικόνας καταχωρείται κατά την εισαγωγή της εικόνας στη λίστα ImageInfoList. Όταν έρθει η ώρα να εμφανιστεί η εικόνα τότε αυτόματα το caption εμφανίζεται στο τμήμα Caption του control. Στο 1ο παράδειγμα παρακάτω, δείχνουμε τον τρόπο.

Μπορούμε, όμως, να ορίσουμε το caption διαφορετικά. Συγκεκριμένα, να εκμεταλλευτούμε το συμβάν ImageChanged και να θέτουμε εκεί είτε ένα διαφορετικό, κάθε φορά, κείμενο είτε ένα σταθερό κείμενο. Στο 2ο παράδειγμα παρακάτω, δείχνουμε τον τρόπο.

### Παράδειγμα 1:

Στο επόμενο παράδειγμα, καθορίζουμε το caption κάθε εικόνας κατά την εισαγωγή της στη λίστα των εικόνων, που πραγματοποιείται στην private μέθοδο CreateListOfImagesWithSpecificeffectForEveryImage.

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
CaptionStyle="{StaticResource SlideShowCaptionStyle }"
ImageChanged="slideShow_ImageChanged" />
```

#### VB:

```
Imports Zeus.WPF.Controls.ImageControls
```

```

Private Sub Window_Loaded(sender As Object, e As RoutedEventArgs)

    'Γεμίζουμε τη λίστα των εικόνων.
    slideShow.ImageInfoList = CreateListOfImagesWithSpecificeffectForEveryImage()

End Sub

'Δημιουργεί τη λίστα των slides. Κάθε slide περιέχει το απόλυτο path της εικόνας,
'το caption και το εφέ εισόδου.
Private Function CreateListOfImagesWithSpecificeffectForEveryImage()
    As List(Of ImageInfo)

    Dim imgList As New List(Of ImageInfo)

    With imgList

        .Add(New ImageInfo(AppPath & "Images\Photo-01.jpg", "Photo-01", _
            SlideShow.SlidingEffectEnum.FadeIn))

        .Add(New ImageInfo(AppPath & "Images\Photo-02.jpg", "Photo-02", _
            SlideShow.SlidingEffectEnum.MoveInFromLeft))

        .Add(New ImageInfo(AppPath & "Images\Αγριοβοτανο-1.jpg", "Αγριοβοτανο-1", _
            SlideShow.SlidingEffectEnum.MoveInFromTopLeft))

        .Add(New ImageInfo(AppPath & "Images\Αγριοβοτανο-2.jpg", "Αγριοβοτανο-2", _
            SlideShow.SlidingEffectEnum.ZoomInFromMiddle))

        .Add(New ImageInfo(AppPath & "Images\Ανδρος.jpg", "Ανδρος", _
            SlideShow.SlidingEffectEnum.ZoomInFromTopLeft))

        .Add(New ImageInfo(AppPath & "Images\Ελληνικά.jpg", "Ελληνικά", _
            SlideShow.SlidingEffectEnum.ZoomInFromBottomRight))

    End With

    Return imgList

End Function

```

Παρακάτω, εμφανίζεται η πρώτη εικόνα που έχει caption "Photo-01".



## Παράδειγμα 2:

Στο επόμενο παράδειγμα, καθορίζουμε **σταθερό caption** για κάθε εικόνα, ανεξάρτητα τί έχουμε καταχωρήσει κατά την εισαγωγή της στη λίστα των εικόνων. Για να το πετύχουμε, εκμεταλλευόμαστε το συμβάν **ImageChanged**:

### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
    CaptionStyle="{StaticResource SlideShowCaptionStyle }"
    ImageChanged="slideShow_ImageChanged" />
```

### VB:

```
Imports Zeus.WPF.Controls.ImageControls

Private Sub Window_Loaded(sender As Object, e As RoutedEventArgs)

    'Γεμίζουμε τη λίστα των εικόνων.
    slideShow.ImageInfoList = CreateListOfImagesWithSpecificEffectForEveryImage()

End Sub

'Εδώ ενημερώνουμε το navigation text που εμφανίζεται στο κάτω δεξί μέρος.
'Επίσης, θέτουμε ένα σταθερό caption για κάθε εικόνα.
Private Sub slideShow_ImageChanged(sender As Object, e As RoutedEventArgs)

    Dim slideShow As SlideShow = TryCast(sender, SlideShow)

    If slideShow IsNot Nothing Then

        With slideShow

            .NavigationText = String.Concat(If(.ImageInfoList.Count > 0, _
                .CurrentImageIndex + 1, 0), "/", .ImageInfoList.Count)
            .Caption = "Εικόνες από τις εκδρομές μου"

        End With

    End If

End Sub

'Δημιουργεί τη λίστα των slides. Κάθε slide περιέχει το απόλυτο path της εικόνας,
'το caption και το εφέ εισόδου.
Private Function CreateListOfImagesWithSpecificEffectForEveryImage()
    As List(Of ImageInfo)

    Dim imgList As New List(Of ImageInfo)

    With imgList

        .Add(New ImageInfo(AppPath & "Images\Photo-01.jpg", "Photo-01", _
```

```
        SlideShow.SlidingEffectEnum.FadeIn))
    .Add(New ImageInfo(AppPath & "Images\Photo-02.jpg", "Photo-02", _
        SlideShow.SlidingEffectEnum.MoveInFromLeft))
    .Add(New ImageInfo(AppPath & "Images\Αγριοβοτανο-1.jpg", "Αγριοβοτανο-1", _
        SlideShow.SlidingEffectEnum.MoveInFromTopLeft))
    .Add(New ImageInfo(AppPath & "Images\Αγριοβοτανο-2.jpg", "Αγριοβοτανο-2", _
        SlideShow.SlidingEffectEnum.ZoomInFromMiddle))
    .Add(New ImageInfo(AppPath & "Images\Ανδρος.jpg", "Ανδρος", _
        SlideShow.SlidingEffectEnum.ZoomInFromTopLeft))
    .Add(New ImageInfo(AppPath & "Images\Ελληνικά.jpg", "Ελληνικά", _
        SlideShow.SlidingEffectEnum.ZoomInFromBottomRight))
```

End With

Return imgList

End Function

Παρακάτω, εμφανίζεται η πρώτη εικόνα. Το caption "Εικόνες από τις εκδρομές μου" θα είναι το ίδιο για όλες τις εικόνες.



## CaptionBorderStyle

Καθορίζει το **στυλ για το περίγραμμα της λεζάντας, τύπου Style**.

### Σύνταξη:

**VB:**

```
Public Property CaptionBorderStyle As Style
```

**Τύπος: System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο περίγραμμα του Caption. Το default style περιλαμβάνει σκούρο φόντο και padding 5 pixels.

### Dependency Property Information:

**Identifier field:** CaptionBorderStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ένα **resource Style** με όνομα **SlideShowCaptionBorderStyle** που καθορίζει μερικά χαρακτηριστικά για το περίγραμμα (border) του caption: BlueViolet χρώμα περιγράμματος, μεγέθους 1 pixel με λίγο στρογγυλεμένες γωνίες και padding σε 5 pixels. Στη συνέχεια, το θέτουμε στην **ιδιότητα CaptionBorderStyle** του SlideShow control με μία StaticResource έκφραση.

**XAML:**

```
<Window.Resources>

    <Style x:Key="SlideShowCaptionBorderStyle" TargetType="{x:Type Border}">
        <Setter Property="BorderBrush" Value="BlueViolet"/>
        <Setter Property="BorderThickness" Value="1" />
        <Setter Property="CornerRadius" Value="5"/>
        <Setter Property="Padding" Value="5" />
    </Style>

    ...

</Window.Resources>

...

<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
    ShowButtonsSmall="False" ShowNavigationText="False"
    CaptionBorderStyle="{StaticResource SlideShowCaptionBorderStyle}"
```

```
CaptionStyle="{StaticResource SlideShowCaptionStyle }"  
ImageChanged="slideShow_ImageChanged" />
```

**VB:**

```
Dim slideShowCaptionBorderStyle As New Style( GetType(Border))  
  
With slideShowCaptionBorderStyle  
  
    .Setters.Add(New Setter With {.[Property] = BorderBrushProperty, _  
                                  .Value = Brushes.BlueViolet})  
  
    .Setters.Add(New Setter With {.[Property] = BorderThicknessProperty, _  
                                  .Value = New Thickness(1)})  
  
    .Setters.Add(New Setter With {.[Property] = Border.CornerRadiusProperty, _  
                                  .Value = New CornerRadius(5)})  
  
    .Setters.Add(New Setter With {.[Property] = Border.PaddingProperty, _  
                                  .Value = New Thickness(5)})  
  
End With  
  
slideShow.CaptionBorderStyle = slideShowCaptionBorderStyle
```





## CaptionStyle

### Σύνταξη:

**VB:**

```
Public Property CaptionStyle As Style
```

**Τύπος:** System.Windows.Style

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο Caption. Το default style περιλαμβάνει διαφανές φόντο, χρώμα κειμένου άσπρο, μέγεθος γραμμάτων 12, οριζόντια στοίχιση αριστερά και text wrapping σε Wrap.

### Dependency Property Information:

**Identifier field:** CaptionStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ένα **resource Style** με όνομα **SlideShowCaptionStyle** που καθορίζει μερικά χαρακτηριστικά για το caption: Γραμματοσειρά "Calibri", στοίχιση στο κέντρο και χρώμα κειμένου "Aqua". Στη συνέχεια, το θέτουμε στην ιδιότητα **CaptionStyle** του SlideShow control με μία StaticResource έκφραση.

**XAML:**

```
<Window.Resources>

    <Style x:Key="SlideShowCaptionStyle" TargetType="{x:Type TextBlock}">

        <Setter Property="FontFamily" Value="Calibri"/>
        <Setter Property="HorizontalAlignment" Value="Center" />
        <Setter Property="Foreground" Value="Aqua"/>

    </Style>

...

</Window.Resources>

...

<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
    CaptionStyle="{StaticResource SlideShowCaptionStyle }"
    ImageChanged="slideShow_ImageChanged" />
```

**VB:**

```
Dim slideShowCaptionStyle As New Style(GetType(TextBlock))
```

```
With slideShowCaptionStyle
```

```
    .Setters.Add(New Setter With {.[Property] = FontFamilyProperty, _  
                                  .Value = New FontFamily("Calibri")})
```

```
    .Setters.Add(New Setter With {.[Property] = HorizontalAlignmentProperty, _  
                                  .Value = HorizontalAlignment.Center})
```

```
    .Setters.Add(New Setter With {.[Property] = ForegroundProperty, _  
                                  .Value = Brushes.Aqua})
```

```
End With
```

```
slideShow.CaptionStyle = slideShowCaptionStyle
```



## CurrentImageIndex

Επιστρέφει το **index** της τρέχουσας εικόνας στη λίστα των εικόνων. Είναι τύπου **Integer**.

### Σύνταξη:

#### VB:

```
Public Property CurrentImageIndex As Integer
```

#### Τύπος: System.Integer

Η τιμή Integer καθορίζει την θέση της τρέχουσας που εμφανίζεται. Η πρώτη θέση έχει index 0.

Η default τιμή είναι 0.

### Dependency Property Information:

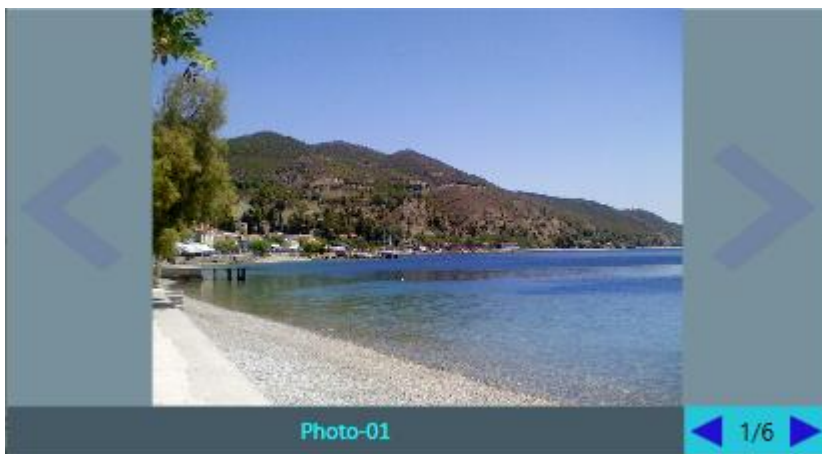
**Identifier field:** CurrentImageIndexProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το navigation text στη μορφή "x/n", με βάση το **CurrentImageIndex** (για το x) και του πλήθους των εικόνων της λίστας (ImageInfoList.Count για το n) μέσα στον **ImageChanged** event handler:

#### VB:

```
Private Sub slideShow_ImageChanged(sender As Object, e As RoutedEventArgs)
    Dim slideShow As SlideShow = TryCast(sender, SlideShow)
    If slideShow IsNot Nothing Then
        With slideShow
            .NavigationText = String.Concat(If(.ImageInfoList.Count > 0, _
                .CurrentImageIndex + 1, 0), "/", .ImageInfoList.Count)
        End With
    End If
End Sub
```



## ImageInfoList

Καθορίζει τη **λίστα των εικόνων**, τύπου **ImageInfo**.

### Σύνταξη:

#### VB:

```
Public Property ImageInfoList As List(Of ImageInfo)
```

**Τύπος:** **List(Of Zeus.WPF.Controls.ImageControls.ImageInfo)**

Προσδιορίζουμε μία λίστα εικόνων, από αντικείμενα ImageInfo, για το slideshow. Αρχικά, η λίστα είναι κενή.

### Dependency Property Information:

**Identifier field:** ImageInfoListProperty

### Παρατηρήσεις:

Κάθε αντικείμενο ImageInfo περιέχει πληροφορίες για: το απόλυτο path του αρχείου εικόνας, το caption και το εφέ εισόδου. Περισσότερα, μπορούμε να δούμε στο παράρτημα\_στο τέλος του manual.

### Παράδειγμα:

Στο επόμενο παράδειγμα, γεμίζουμε τη λίστα των εικόνων με τα αντικείμενα ImageInfo όταν φορτωθεί το παράθυρο. Η όλη διαδικασία επιτελείται καλώντας τη συνάρτηση CreateListOfImagesWithSpecificeffectForEveryImage:

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn" ... />
```

#### VB:

```
Imports Zeus.WPF.Controls.ImageControls
```

```
Private Sub Window_Loaded(sender As Object, e As RoutedEventArgs)
```

```
    'Γεμίζουμε τη λίστα των εικόνων.
```

```
    slideShow.ImageInfoList = CreateListOfImagesWithSpecificeffectForEveryImage()
```

```
End Sub
```

'Δημιουργεί τη λίστα των slides. Κάθε slide περιέχει το απόλυτο path της εικόνας, το caption και το εφέ εισόδου.

```
Private Function CreateListOfImagesWithSpecificEffectForEveryImage()
    As List(Of ImageInfo)

    Dim imgList As New List(Of ImageInfo)

    With imgList

        .Add(New ImageInfo(AppPath & "Images\Photo-01.jpg", "Photo-01", _
            SlideShow.SlidingEffectEnum.FadeIn))

        .Add(New ImageInfo(AppPath & "Images\Photo-02.jpg", "Photo-02", _
            SlideShow.SlidingEffectEnum.MoveInFromLeft))

        .Add(New ImageInfo(AppPath & "Images\Αγριοβοτανο-1.jpg", "Αγριοβοτανο-1", _
            SlideShow.SlidingEffectEnum.MoveInFromTopLeft))

        .Add(New ImageInfo(AppPath & "Images\Αγριοβοτανο-2.jpg", "Αγριοβοτανο-2", _
            SlideShow.SlidingEffectEnum.ZoomInFromMiddle))

        .Add(New ImageInfo(AppPath & "Images\Ανδρος.jpg", "Ανδρος", _
            SlideShow.SlidingEffectEnum.ZoomInFromTopLeft))

        .Add(New ImageInfo(AppPath & "Images\Ελληνικά.jpg", "Ελληνικά", _
            SlideShow.SlidingEffectEnum.ZoomInFromBottomRight))

    End With

    Return imgList

End Function
```

Παρακάτω, εμφανίζεται η πρώτη εικόνα που έχει caption "Photo-01".



## Looping

Καθορίζει αν θα επαναλαμβάνεται το slideshow από την αρχή (είτε στο auto-play είτε χειροκίνητα μέσω των buttons previous/next). Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property Looping As Boolean
```

**Τύπος:** **System.Boolean**

Προσδιορίζουμε μία τιμή Boolean για το αν το slideshow θα επαναλαμβάνεται από την αρχή. Η default τιμή είναι False.

### Dependency Property Information:

**Identifier field:** LoopingProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, **καθορίζουμε ότι το slideshow θα επαναλαμβάνεται**, θέτοντας την **ιδιότητα Looping** σε **True**. Έτσι, όταν ο χρήστης πατήσει το button "Next" στην τελευταία εικόνα τότε θα επιστρέψει στην πρώτη και αντίθετα όταν πατήσει το button "Previous" στην πρώτη εικόνα τότε θα πάει στην τελευταία. Στο auto-play, το slideshow ξεκινάει από την πρώτη εικόνα και πάει προς την τελευταία και κατόπιν από την αρχή.

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C"  
    SlidingEffect="FadeIn"  
    Looping="True" ... />
```

#### VB:

```
slideShow.Loopig = True
```

## NavigationAreaBorderStyle

Καθορίζει το **στυλ για το περίγραμμα της περιοχής πλοήγησης (στο κάτω δεξί μέρος), τύπου Style.**

### Σύνταξη:

#### VB:

```
Public Property NavigationAreaBorderStyle As Style
```

**Τύπος:** System.Windows.Style

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο περίγραμμα της περιοχής πλοήγησης (στο κάτω δεξί μέρος).

Το default style περιλαμβάνει φόντο γαλάζιας απόχρωσης.

### Dependency Property Information:

**Identifier field:** NavigationAreaBorderStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ένα **resource Style** με όνομα **SlideShowNavigationAreaBorderStyle** που καθορίζει μερικά χαρακτηριστικά για το περίγραμμα (border) της περιοχής πλοήγησης (navigation area): πρασινωπό φόντο, μπλε χρώμα περιγράμματος, μεγέθους 1 pixel με λίγο στρογγυλεμένες γωνίες και padding σε 5 pixels. Στη συνέχεια, το θέτουμε στην **ιδιότητα NavigationAreaBorderStyle** του SlideShow control με μία StaticResource έκφραση.

#### XAML:

```
<Window.Resources>

...

<!-- custom NavigationAreaBorderStyle -->
<Style x:Key="SlideShowNavigationAreaBorderStyle" TargetType="{x:Type Border}">

    <Setter Property="Background" Value="#FF4BB21B" />
    <Setter Property="BorderBrush" Value="#FF2B65E2"/>
    <Setter Property="BorderThickness" Value="1" />
    <Setter Property="CornerRadius" Value="5"/>
    <Setter Property="Padding" Value="5" />

</Style>

...
```



```
</Window.Resources>
```

```
...
```

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
CaptionStyle="{StaticResource SlideShowCaptionStyle}"
NavigationAreaBorderStyle="{StaticResource
SlideShowNavigationAreaBorderStyle}"
ImageChanged="slideShow_ImageChanged" />
```

**VB:**

```
Dim slideShowNavigationAreaBorderStyle As New Style(GetType(Border))
```

```
With slideShowNavigationAreaBorderStyle
```

```
.Setters.Add(New Setter With {.Property = BackgroundProperty, _
.Value = New SolidColorBrush(ColorConverter.ConvertFromString("#FF4BB21B"))})
```

```
.Setters.Add(New Setter With {.Property = BorderBrushProperty, _
.Value = New SolidColorBrush(ColorConverter.ConvertFromString("#FF2B65E2"))})
```

```
.Setters.Add(New Setter With {.Property = BorderThicknessProperty, _
.Value = New Thickness(1)})
```

```
.Setters.Add(New Setter With {.Property = BorderThicknessProperty, _
.Value = New Thickness(1)})
```

```
.Setters.Add(New Setter With {.Property = Border.CornerRadiusProperty, _
.Value = New CornerRadius(5)})
```

```
.Setters.Add(New Setter With {.Property = Border.PaddingProperty, _
.Value = New Thickness(5)})
```

```
End With
```

```
slideShow.NavigationAreaBorderStyle = slideShowNavigationAreaBorderStyle
```



## NavigationText

Καθορίζει το κείμενο πλοήγησης (π.χ. 1 από 10 ή 1/10), τύπου `String`.

### Σύνταξη:

#### VB:

```
Public Property NavigationText As String
```

**Τύπος:** `System.String`

Προσδιορίζουμε μία τιμή `String` που θα αποτελεί το κείμενο της πλοήγησης. Η default τιμή είναι "0/0".

### Dependency Property Information:

**Identifier field:** `NavigationTextProperty`

### Παρατηρήσεις:

Τυπικά, εκμεταλλευόμαστε το συμβάν `ImageChanged` για να τοποθετήσουμε εκεί, κάθε φορά που αλλάζει η εικόνα, πληροφορίες για την τρέχουσα θέση στο σύνολο των εικόνων.

### Παράδειγμα:

Στο επόμενο παράδειγμα, τοποθετούμε στον `ImageChanged` event handler πληροφορίες για την τρέχουσα θέση στο σύνολο των εικόνων στην ιδιότητα `NavigationText`:

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
CaptionStyle="{StaticResource SlideShowCaptionStyle }"
ImageChanged="slideShow_ImageChanged" />
```

#### VB:

```
Private Sub slideShow_ImageChanged(sender As Object, e As RoutedEventArgs)
    Dim slideShow As SlideShow = TryCast(sender, SlideShow)
    If slideShow IsNot Nothing Then
        With slideShow
            .NavigationText = String.Concat(If(.ImageInfoList.Count > 0, _
```

```
.CurrentImageIndex + 1, 0), "/", .ImageInfoList.Count)
```

```
End With
```

```
End If
```

```
End Sub
```



## NavigationTextStyle

Καθορίζει το **στυλ για το κείμενο της πλοήγησης, τύπου Style**.

### Σύνταξη:

#### VB:

```
Public Property NavigationTextStyle As Style
```

**Τύπος:** System.Windows.Style

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο κείμενο της πλοήγησης (στο κάτω δεξί μέρος).

Το default style περιλαμβάνει: διαφανές φόντο, χρώμα κειμένου μαύρο, μέγεθος γραμμάτων 12, οριζόντια και κατακόρυφη στοίχιση στο κέντρο, και αναδίπλωση κειμένου (text wrapping).

### Dependency Property Information:

**Identifier field:** NavigationTextStyleProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ένα **resource Style** με όνομα **SlideShowNavigationTextStyle** που καθορίζει μερικά χαρακτηριστικά για το κείμενο της πλοήγησης (navigation text): κιτρινωπό χρώμα κειμένου, γραμματοσειρά "Courier New", κατακόρυφη στοίχιση στο κέντρο, έντονη γραφή και padding στα 5 pixels. Στη συνέχεια, το θέτουμε στην **ιδιότητα NavigationTextStyle** του SlideShow control με μία StaticResource έκφραση.

#### XAML:

```
<Window.Resources>
```

```
...
```

```
<!-- custom NavigationTextStyle -->
<Style x:Key="SlideShowNavigationTextStyle" TargetType="{x:Type TextBlock}">
    <Setter Property="Foreground" Value="#FFF176"/>
    <Setter Property="FontFamily" Value="Courier New" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="FontWeight" Value="Bold"/>
    <Setter Property="Padding" Value="5,0"/>
</Style>
```

```
...  
</Window.Resources>  
...  
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"  
CaptionStyle="{StaticResource SlideShowCaptionStyle }"  
NavigationTextStyle="{StaticResource SlideShowNavigationTextStyle}"  
ImageChanged="slideShow_ImageChanged" />
```

**VB:**

```
Dim slideShowNavigationTextStyle As New Style(GetType(TextBlock))  
  
With slideShowNavigationTextStyle  
    .Setters.Add(New Setter With {.Property = ForegroundProperty, _  
        .Value = New SolidColorBrush(ColorConverter.ConvertFromString("#FFF176"))})  
    .Setters.Add(New Setter With {.Property = FontFamilyProperty, _  
        .Value = New FontFamily("Monotype Corsiva")})  
    .Setters.Add(New Setter With {.Property = VerticalAlignmentProperty, _  
        .Value = VerticalAlignment.Center})  
    .Setters.Add(New Setter With {.Property = FontWeightProperty, _  
        .Value = FontWeights.Bold})  
    .Setters.Add(New Setter With {.Property = TextBlock.PaddingProperty, _  
        .Value = New Thickness(5,0,5,0)})  
  
End With  
  
slideShow.NavigationTextStyle = slideShowNavigationTextStyle
```



## NextButtonInSpaceStyle

Καθορίζει το **στυλ** για το **button "Next"** που βρίσκεται στον χώρο της εικόνας (**in-space**). Είναι τύπου **Style**.

### Σύνταξη:

**VB:**

```
Public Property NextButtonInSpaceStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα **Style** που θα εφαρμοστεί στο **button "Next"** που βρίσκεται στον χώρο της εικόνας (**in-space**).

Το **default style** περιλαμβάνει ένα μη προσαρμόσιμο γραφικό μεγέθους **64x64 pixels**.

### Dependency Property Information:

**Identifier field:** `NextButtonInSpaceStyleProperty`

### Παρατηρήσεις:

Για να αλλάξουμε το **style** του **button** πρέπει να αλλάξουμε το **ControlTemplate**. Το παρακάτω παράδειγμα δείχνει τον τρόπο.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ένα **resource Style** **template** που θα χρησιμοποιηθεί ως βάση για τα **styles** των **in-space buttons** **"Previous"** και **"Next"**. Συγκεκριμένα, δημιουργούμε το **resource SlideShowButtonInSpaceTemplateStyle** πάνω στο οποίο δημιουργούνται τα **resources SlideShowPreviousButtonInSpaceStyle** και **SlideShowNextButtonInSpaceStyle**. Στη συνέχεια, τα θέτουμε στις αντίστοιχες **ιδιότητες PreviousButtonInSpaceStyle** και **NextButtonInSpaceStyle**.

Βέβαια, το παράδειγμα είναι σχετικά απλό. Θα μπορούσαμε να δημιουργήσουμε πιο πολύπλοκα **styles** με έτοιμα γραφικά.

**XAML:**

```
<Window.Resources>
...
<!-- custom in-space button base style -->
<Style x:Key="SlideShowButtonInSpaceTemplateStyle" TargetType="{x:Type Button}">
```

```
<Setter Property="Cursor" Value="Hand"/>
<Setter Property="Template" >
  <Setter.Value >
    <ControlTemplate TargetType="{x:Type Button}">
      <Border CornerRadius="50" Width="64" Height="64"
        BorderBrush="White" BorderThickness="1" >
        <ContentPresenter HorizontalAlignment="Center"
          VerticalAlignment="Center" />
      </Border>
    </ControlTemplate>
  </Setter.Value
</Setter>

</Style>

<!-- custom in-space button style για το "Previous" -->
<Style x:Key="SlideShowPreviousButtonInSpaceStyle" TargetType="{x:Type Button}"
  BasedOn="{StaticResource SlideShowButtonInSpaceTemplateStyle }">

  <Setter Property="Content" >
    <Setter.Value >
      <TextBlock Text="&lt;" FontSize="32" Foreground="White" />
    </Setter.Value>
  </Setter>
  <Style.Triggers >
    <Trigger Property="IsMouseOver" Value="True">
      <Setter Property="Content" >
        <Setter.Value >
          <TextBlock Text="&lt;" FontSize="32"
            Foreground="Yellow" />
        </Setter.Value>
      </Setter>
    </Trigger>
  </Style.Triggers>

</Style>

<!-- custom in-space button style για το "Next" -->
<Style x:Key="SlideShowNextButtonInSpaceStyle" TargetType="{x:Type Button}"
  BasedOn="{StaticResource SlideShowButtonInSpaceTemplateStyle }">

  <Setter Property="Content"
    <Setter.Value >
      <TextBlock Text="&gt;" FontSize="32" Foreground="White" />
    </Setter.Value>
  </Setter>
  <Style.Triggers >
    <Trigger Property="IsMouseOver" Value="True">
      <Setter Property="Content" >
        <Setter.Value >
          <TextBlock Text="&gt;" FontSize="32"
            Foreground="Yellow" />
        </Setter.Value>
      </Setter>
    </Trigger>
  </Style.Triggers>

</Style>

...
```

```
</Window.Resources>
```

...

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"  
CaptionStyle="{StaticResource SlideShowCaptionStyle }"  
PreviousButtonInSpaceStyle="{StaticResource  
SlideShowPreviousButtonInSpaceStyle}"  
NextButtonInSpaceStyle="{StaticResource  
SlideShowNextButtonInSpaceStyle}"  
ImageChanged="slideShow_ImageChanged" />
```





## NextButtonSmallStyle

Καθορίζει το **στυλ** για το μικρό **button** "Next" που βρίσκεται στον χώρο της πλοήγησης. Είναι τύπου **Style**.

### Σύνταξη:

**VB:**

```
Public Property NextButtonSmallStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα **Style** που θα εφαρμοστεί στο **button** "Next" που βρίσκεται στον χώρο της πλοήγησης (στο κάτω δεξί μέρος).

Το default style περιλαμβάνει ένα μη προσαρμόσιμο γραφικό μεγέθους 16x16 pixels.

### Dependency Property Information:

**Identifier field:** NextButtonSmallStyleProperty

### Παρατηρήσεις:

Για να αλλάξουμε το style του **button** πρέπει να αλλάξουμε το **ControlTemplate**. Το παρακάτω παράδειγμα δείχνει τον τρόπο.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ένα **resource Style** template που θα χρησιμοποιηθεί ως βάση για τα styles των small buttons "Previous" και "Next". Συγκεκριμένα, δημιουργούμε το **resource SlideShowButtonInSpaceTemplateStyle** πάνω στο οποίο δημιουργούνται τα **resources SlideShowPreviousButtonInSpaceStyle** και **SlideShowNextButtonInSpaceStyle**. Στη συνέχεια, τα θέτουμε στις αντίστοιχες **ιδιότητες PreviousButtonInSpaceStyle** και **NextButtonInSpaceStyle**.

Βέβαια, το παράδειγμα είναι σχετικά απλό. Θα μπορούσαμε να δημιουργήσουμε πιο πολύπλοκα styles με έτοιμα γραφικά.

**XAML:**

```
<Window.Resources>
```

```
...
```

```
<!-- custom small button base style -->
```

```
<Style x:Key="SlideShowButtonSmallTemplateStyle" TargetType="{x:Type Button}">
```

---

```

<Setter Property="Cursor" Value="Hand"/>
<Setter Property="Template" >
  <Setter.Value >
    <ControlTemplate TargetType="{x:Type Button}">
      <Border Width="16" Height="16"
        VerticalAlignment="Center"
        HorizontalAlignment="Center" >
        <ContentPresenter HorizontalAlignment="Center"
          VerticalAlignment="Center" />
      </Border>
    </ControlTemplate>
  </Setter.Value >
</Setter>

</Style>

<!-- custom small button style για το "Previous" -->
<Style x:Key="SlideShowPreviousButtonSmallStyle" TargetType="{x:Type Button}"
  BasedOn="{StaticResource SlideShowButtonSmallTemplateStyle}">

  <Setter Property="Content" >
    <Setter.Value >
      <TextBlock Text="&lt;" FontSize="12" Foreground="Blue"
        FontWeight="ExtraBold" />
    </Setter.Value >
  </Setter>
  <Style.Triggers >
    <Trigger Property="IsMouseOver" Value="True">
      <Setter Property="Content" >
        <Setter.Value >
          <TextBlock Text="&lt;" FontSize="12"
            Foreground="Cyan"
            FontWeight="ExtraBold" />
        </Setter.Value >
      </Setter>
    </Trigger>
  </Style.Triggers>

</Style>

<!-- custom small button style για το "Next" -->
<Style x:Key="SlideShowNextButtonSmallStyle" TargetType="{x:Type Button}"
  BasedOn="{StaticResource SlideShowButtonSmallTemplateStyle}">

  <Setter Property="Content" >
    <Setter.Value >
      <TextBlock Text="&gt;" FontSize="12" Foreground="Blue"
        FontWeight="ExtraBold" />
    </Setter.Value >
  </Setter>
  <Style.Triggers >
    <Trigger Property="IsMouseOver" Value="True">
      <Setter Property="Content" >
        <Setter.Value >
          <TextBlock Text="&gt;" FontSize="12"
            Foreground="Cyan"
            FontWeight="ExtraBold" />
        </Setter.Value >
      </Setter>
    </Trigger>
  </Style.Triggers>

```

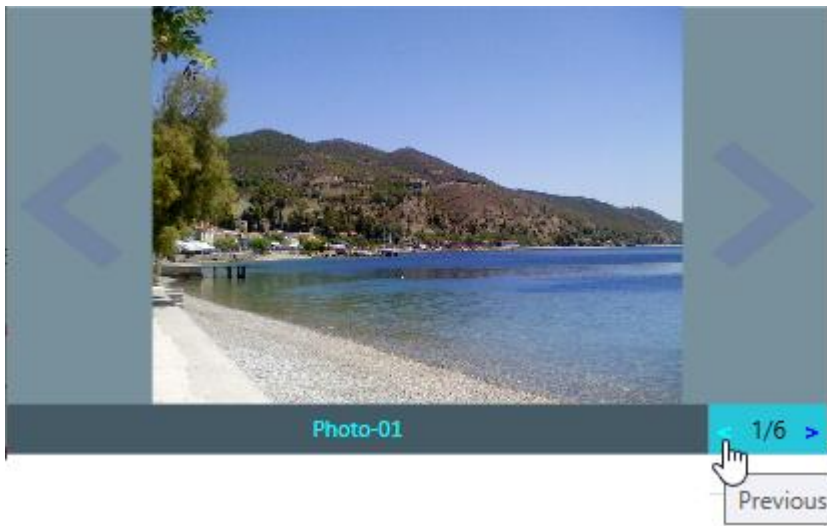
```
</Style>
```

```
...
```

```
</Window.Resources>
```

```
...
```

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"  
CaptionStyle="{StaticResource SlideShowCaptionStyle }"  
PreviousButtonSmallStyle="{StaticResource  
SlideShowPreviousButtonSmallStyle}"  
NextButtonSmallStyle="{StaticResource  
SlideShowNextButtonSmallStyle}"  
ImageChanged="slideShow_ImageChanged" />
```



## PreviousButtonInSpaceStyle

Καθορίζει το **στυλ για το button "Previous" που βρίσκεται στον χώρο της εικόνας (in-space)**. Είναι τύπου **Style**.

### Σύνταξη:

**VB:**

```
Public Property PreviousButtonInSpaceStyle As Style
```

**Τύπος:** **System.Windows.Style**

Προσδιορίζουμε ένα Style που θα εφαρμοστεί στο button "Previous" που βρίσκεται στον χώρο της εικόνας (in-space). Το default style περιλαμβάνει ένα μη προσαρμόσιμο γραφικό μεγέθους 64x64 pixels.

### Dependency Property Information:

**Identifier field:** PreviousButtonInSpaceStyleProperty

### Παρατηρήσεις:

Για να αλλάξουμε το style του button πρέπει να αλλάξουμε το ControlTemplate. Το παρακάτω παράδειγμα δείχνει τον τρόπο.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ένα **resource Style** template που θα χρησιμοποιηθεί ως βάση για τα styles των in-space buttons "Previous" και "Next". Συγκεκριμένα, δημιουργούμε το **resource SlideShowButtonInSpaceTemplateStyle** πάνω στο οποίο δημιουργούνται τα **resources SlideShowPreviousButtonInSpaceStyle** και **SlideShowNextButtonInSpaceStyle**. Στη συνέχεια, τα θέτουμε στις αντίστοιχες **ιδιότητες PreviousButtonInSpaceStyle** και **NextButtonInSpaceStyle**.

Βέβαια, το παράδειγμα είναι σχετικά απλό. Θα μπορούσαμε να δημιουργήσουμε πιο πολύπλοκα styles με έτοιμα γραφικά.

**XAML:**

```
<Window.Resources>
```

```
...
```

```
<!-- custom in-space button base style -->
```

```
<Style x:Key="SlideShowButtonInSpaceTemplateStyle" TargetType="{x:Type Button}">
```

---

```
<Setter Property="Cursor" Value="Hand"/>
<Setter Property="Template" >
  <Setter.Value >
    <ControlTemplate TargetType="{x:Type Button}">
      <Border CornerRadius="50" Width="64" Height="64"
        BorderBrush="White" BorderThickness="1" >
        <ContentPresenter HorizontalAlignment="Center"
          VerticalAlignment="Center" />
      </Border>
    </ControlTemplate>
  </Setter.Value>
</Setter>
</Style>

<!-- custom in-space button style για το "Previous" -->
<Style x:Key="SlideShowPreviousButtonInSpaceStyle" TargetType="{x:Type Button}"
  BasedOn="{StaticResource SlideShowButtonInSpaceTemplateStyle }">

  <Setter Property="Content" >
    <Setter.Value >
      <TextBlock Text="&lt;" FontSize="32" Foreground="White" />
    </Setter.Value>
  </Setter>
  <Style.Triggers >
    <Trigger Property="IsMouseOver" Value="True">
      <Setter Property="Content" >
        <Setter.Value >
          <TextBlock Text="&lt;" FontSize="32"
            Foreground="Yellow" />
        </Setter.Value>
      </Setter>
    </Trigger>
  </Style.Triggers>

</Style>

<!-- custom in-space button style για το "Next" -->
<Style x:Key="SlideShowNextButtonInSpaceStyle" TargetType="{x:Type Button}"
  BasedOn="{StaticResource SlideShowButtonInSpaceTemplateStyle }">

  <Setter Property="Content"
    <Setter.Value >
      <TextBlock Text="&gt;" FontSize="32" Foreground="White" />
    </Setter.Value>
  </Setter>
  <Style.Triggers >
    <Trigger Property="IsMouseOver" Value="True">
      <Setter Property="Content" >
        <Setter.Value >
          <TextBlock Text="&gt;" FontSize="32"
            Foreground="Yellow" />
        </Setter.Value>
      </Setter>
    </Trigger>
  </Style.Triggers>

</Style>

...
```

```
</Window.Resources>
```

...

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"  
CaptionStyle="{StaticResource SlideShowCaptionStyle }"  
PreviousButtonInSpaceStyle="{StaticResource  
SlideShowPreviousButtonInSpaceStyle}"  
NextButtonInSpaceStyle="{StaticResource  
SlideShowNextButtonInSpaceStyle}"  
ImageChanged="slideShow_ImageChanged" />
```



## PreviousButtonSmallStyle

Καθορίζει το στυλ για το μικρό button "Previous" που βρίσκεται στον χώρο της πλοήγησης. Είναι τύπου `Style`.

### Σύνταξη:

**VB:**

```
Public Property PreviousButtonSmallStyle As Style
```

**Τύπος:** `System.Windows.Style`

Προσδιορίζουμε ένα `Style` που θα εφαρμοστεί στο button "Previous" που βρίσκεται στον χώρο της πλοήγησης (στο κάτω δεξί μέρος). Το default style περιλαμβάνει ένα μη προσαρμόσιμο γραφικό μεγέθους 16x16 pixels.

### Dependency Property Information:

**Identifier field:** `PreviousButtonSmallStyleProperty`

### Παρατηρήσεις:

Για να αλλάξουμε το style του button πρέπει να αλλάξουμε το `ControlTemplate`. Το παρακάτω παράδειγμα δείχνει τον τρόπο.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ένα **resource Style** template που θα χρησιμοποιηθεί ως βάση για τα styles των small buttons "Previous" και "Next". Συγκεκριμένα, δημιουργούμε το **resource SlideShowButtonInSpaceTemplateStyle** πάνω στο οποίο δημιουργούνται τα **resources SlideShowPreviousButtonInSpaceStyle** και **SlideShowNextButtonInSpaceStyle**. Στη συνέχεια, τα θέτουμε στις αντίστοιχες **ιδιότητες PreviousButtonInSpaceStyle** και **NextButtonInSpaceStyle**. Βέβαια, το παράδειγμα είναι σχετικά απλό. Θα μπορούσαμε να δημιουργήσουμε πιο πολύπλοκα styles με έτοιμα γραφικά.

**XAML:**

```
<Window.Resources>
...
<!-- custom small button base style -->
<Style x:Key="SlideShowButtonSmallTemplateStyle" TargetType="{x:Type Button}">
```

```

<Setter Property="Cursor" Value="Hand"/>
<Setter Property="Template" >
  <Setter.Value >
    <ControlTemplate TargetType="{x:Type Button}">
      <Border Width="16" Height="16"
        VerticalAlignment="Center"
        HorizontalAlignment="Center" >
        <ContentPresenter HorizontalAlignment="Center"
          VerticalAlignment="Center" />
      </Border>
    </ControlTemplate>
  </Setter.Value
</Setter>

</Style>

<!-- custom small button style για το "Previous" -->
<Style x:Key="SlideShowPreviousButtonSmallStyle" TargetType="{x:Type Button}"
  BasedOn="{StaticResource SlideShowButtonSmallTemplateStyle}">

  <Setter Property="Content" >
    <Setter.Value >
      <TextBlock Text="&lt;" FontSize="12" Foreground="Blue"
        FontWeight="ExtraBold" />
    </Setter.Value>
  </Setter>
  <Style.Triggers >
    <Trigger Property="IsMouseOver" Value="True">
      <Setter Property="Content" >
        <Setter.Value >
          <TextBlock Text="&lt;" FontSize="12"
            Foreground="Cyan"
            FontWeight="ExtraBold" />
        </Setter.Value>
      </Setter>
    </Trigger>
  </Style.Triggers>

</Style>

<!-- custom small button style για το "Next" -->
<Style x:Key="SlideShowNextButtonSmallStyle" TargetType="{x:Type Button}"
  BasedOn="{StaticResource SlideShowButtonSmallTemplateStyle}">

  <Setter Property="Content"
    <Setter.Value >
      <TextBlock Text="&gt;" FontSize="12" Foreground="Blue"
        FontWeight="ExtraBold" />
    </Setter.Value>
  </Setter>
  <Style.Triggers >
    <Trigger Property="IsMouseOver" Value="True">
      <Setter Property="Content" >
        <Setter.Value >
          <TextBlock Text="&gt;" FontSize="12"
            Foreground="Cyan"
            FontWeight="ExtraBold" />
        </Setter.Value>
      </Setter>
    </Trigger>
  </Style.Triggers>

```



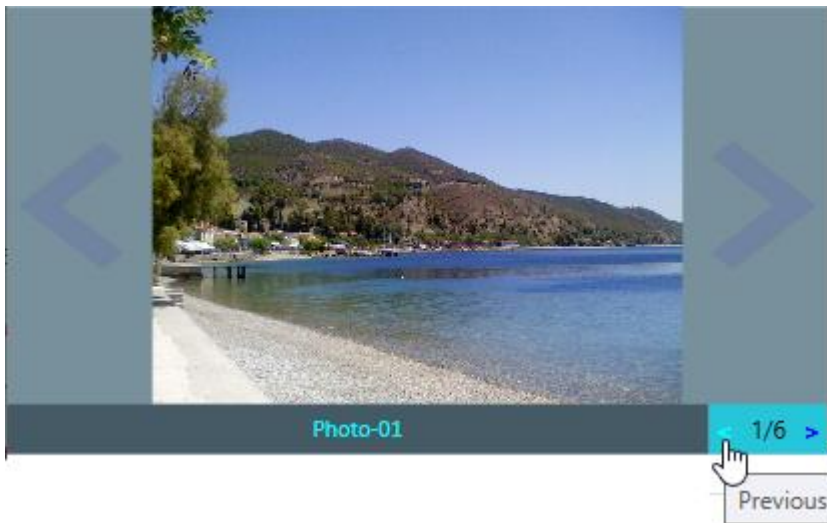
</Style>

...

</Window.Resources>

...

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
CaptionStyle="{StaticResource SlideShowCaptionStyle }"
PreviousButtonSmallStyle="{StaticResource
                          SlideShowPreviousButtonSmallStyle}"
NextButtonSmallStyle="{StaticResource
                       SlideShowNextButtonSmallStyle}"
ImageChanged="slideShow_ImageChanged" />
```



## ShowButtonsInSpace

Καθορίζει αν θα εμφανίζονται τα buttons "Previous/Next" στον χώρο της εικόνας (in-space). Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property ShowButtonsInSpace As Boolean
```

#### Τύπος: System.Boolean

Προσδιορίζουμε μία τιμή Boolean για το αν τα buttons "Previous/Next" θα είναι ορατά στον χώρο της εικόνας (in-space).

Η default τιμή είναι True.

### Dependency Property Information:

**Identifier field:** ShowButtonsInSpaceProperty

### Παράδειγμα:

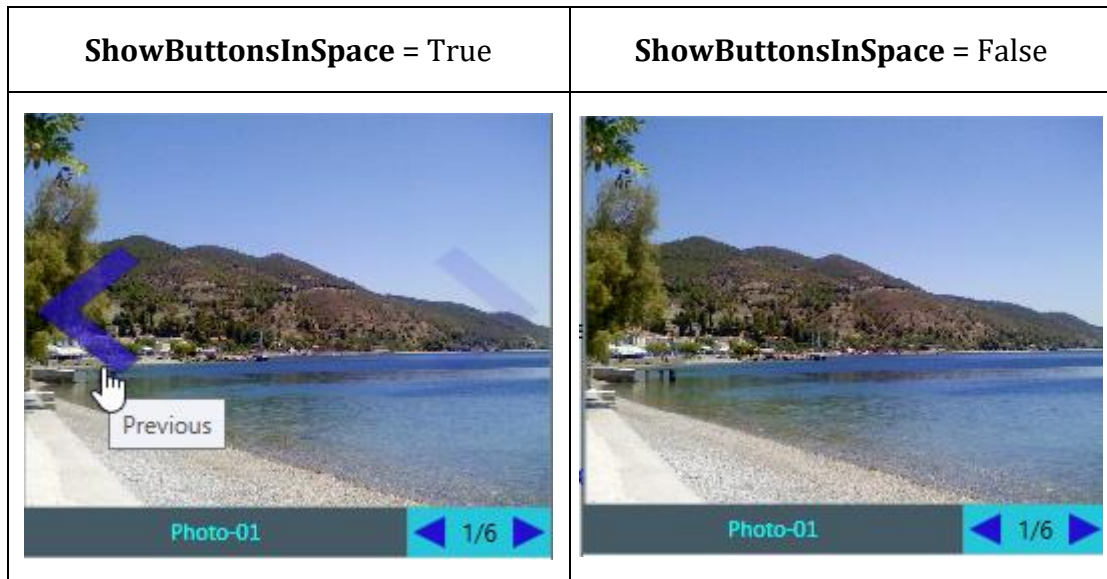
Στο επόμενο παράδειγμα, καθορίζουμε ότι τα buttons "Previous/Next" δεν θα εμφανίζονται στον χώρο της εικόνας (in-space).

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn" ShowButtonsInSpace="False" ... />
```

#### VB:

```
slideShow.ShowButtonsInSpace = False
```



## ShowButtonsSmall

Καθορίζει αν θα εμφανίζονται τα μικρά buttons "Previous/Next" στον χώρο της πλοήγησης. Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property ShowButtonsSmall As Boolean
```

**Τύπος:** System.Boolean

Προσδιορίζουμε μία τιμή Boolean για το αν τα buttons "Previous/Next" θα είναι ορατά στον χώρο της πλοήγησης.  
Η default τιμή είναι True.

### Dependency Property Information:

**Identifier field:** ShowButtonsSmallProperty

### Παράδειγμα:



Στο επόμενο παράδειγμα, καθορίζουμε ότι τα buttons "Previous/Next" δεν θα εμφανίζονται στον χώρο της πλοήγησης.

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"  
                ShowButtonsSmall="False" ... />
```

#### VB:

```
slideShow.ShowButtonsSmall = False
```

<b>ShowButtonsSmall = True</b>	<b>ShowButtonsSmall = False</b>
 <p>The image shows a slide show preview for a landscape photo. The photo is centered within a light blue frame. On the left and right sides of the frame, there are semi-transparent blue vertical bars containing white left and right arrowheads. At the bottom of the frame, there is a dark blue bar with the text "Photo-01" on the left and a small cyan box containing "1/6" on the right, flanked by white left and right arrowheads.</p>	 <p>The image shows a slide show preview for the same landscape photo. The photo is centered within a light blue frame. On the left and right sides of the frame, there are semi-transparent blue vertical bars. At the bottom of the frame, there is a dark blue bar with the text "Photo-01" on the left and a small cyan box containing "1/6" on the right.</p>

## ShowCaption

Καθορίζει αν θα εμφανίζεται η λεζάντα της εικόνας. Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property ShowCaption As Boolean
```

**Τύπος:** **System.Boolean**

Προσδιορίζουμε μία τιμή Boolean για το αν η λεζάντα θα εμφανίζεται.  
Η default τιμή είναι True.

### Dependency Property Information:

**Identifier field:** ShowCaptionProperty

### Παράδειγμα:



Στο επόμενο παράδειγμα, καθορίζουμε ότι η λεζάντα δεν θα εμφανίζεται.

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"  
                ShowCaption="False" ... />
```

#### VB:

```
slideShow.ShowCaption = False
```

<b>ShowCaption = True</b>	<b>ShowCaption = False</b>
 <p>A slide show preview window with a light blue background. It displays a landscape photo of a lake and mountains. At the bottom left, the text "Photo-01" is visible. At the bottom center, there are navigation arrows and the text "1/6".</p>	 <p>A slide show preview window with a light blue background. It displays the same landscape photo as the left window. At the bottom center, there are navigation arrows and the text "1/6". The caption "Photo-01" is not present.</p>

## ShowNavigationText

Καθορίζει αν θα εμφανίζεται το κείμενο της πλοήγησης (π.χ. 1/10) στον χώρο της πλοήγησης. Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property ShowNavigationText As Boolean
```

**Τύπος:** System.Boolean

Προσδιορίζουμε μία τιμή Boolean για το αν θα εμφανίζεται το κείμενο της πλοήγησης. Η default τιμή είναι True.

### Dependency Property Information:

**Identifier field:** ShowNavigationTextProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ότι το κείμενο της πλοήγησης δεν θα εμφανίζεται.

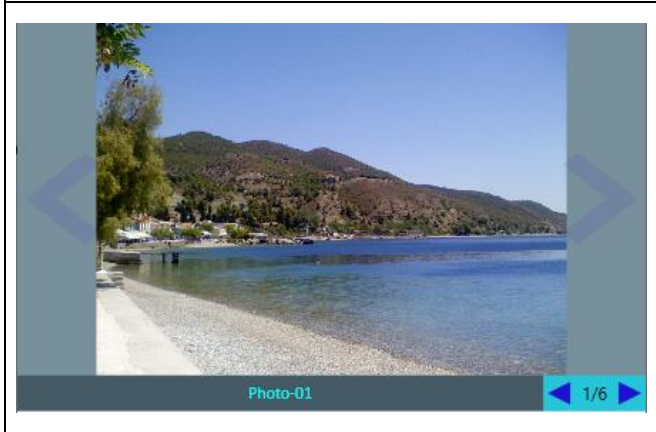
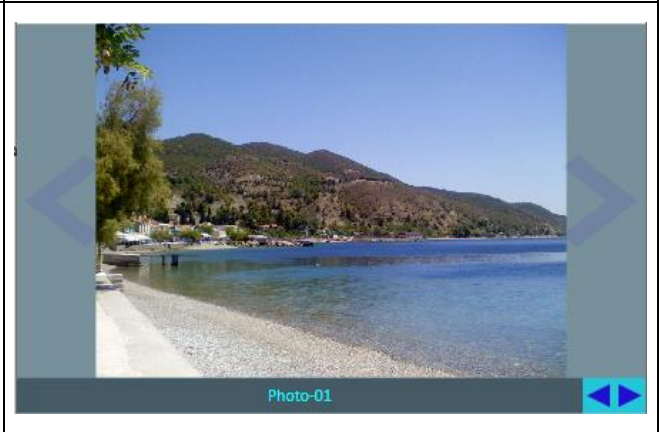
#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn" ShowNavigationText="False" ... />
```

#### VB:

```
slideShow.ShowNavigationText = False
```



<b>ShowNavigationText = True</b>	<b>ShowNavigationText = False</b>
 A screenshot of a slide show interface. The main area shows a landscape photo of a lake and mountains. On the left and right sides, there are semi-transparent grey vertical bars containing light blue navigation arrows. At the bottom, a dark blue bar contains the text "Photo-01" on the left and "1/6" in the center, flanked by small blue navigation arrows.	 A screenshot of a slide show interface, identical to the one on the left. However, the bottom dark blue bar only contains the text "Photo-01" on the left and blue navigation arrows on the right, with no page number displayed.

## ShowToolTipsForPreviousNextButtons

Καθορίζει αν θα εμφανίζονται τα tooltips για τα buttons "Previous/Next". Είναι τύπου **Boolean**.

### Σύνταξη:

#### VB:

```
Public Property ShowToolTipsForPreviousNextButtons As Boolean
```

#### Τύπος: System.Boolean

Προσδιορίζουμε μία τιμή Boolean για το αν θα εμφανίζονται τα tooltips για τα buttons "Previous/Next" τόσο στον χώρο της εικόνας (in-space) όσο στο χώρο της πλοήγησης στο κάτω δεξιό μέρος.

Η default τιμή είναι True.

### Dependency Property Information:

**Identifier field:** ShowToolTipsForPreviousNextButtonsProperty

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ότι τα tooltips δεν θα εμφανίζονται.

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
    ShowToolTipsForPreviousNextButtons="False" ... />
```

#### VB:

```
slideShow.ShowToolTipsForPreviousNextButtons = False
```

## SlideDuration

Καθορίζει τη διάρκεια παραμονής της κάθε εικόνας στην οθόνη, σε ένα auto-play slideshow. Είναι τύπου **Duration**.

### Σύνταξη:

#### VB:

```
Public Property SlideDuration As Duration
```

#### Τύπος: **System.Windows.Duration**

Προσδιορίζουμε μία τιμή **Duration** που είναι ο χρόνος παραμονής της εικόνας στην οθόνη, σε ένα auto-play slideshow.

Η default τιμή είναι `TimeSpan.FromSeconds(5)`, δηλαδή 5 secs.

### Dependency Property Information:

**Identifier field:** `SlideDurationProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε τον χρόνο παραμονής των εικόνων στην οθόνη σε 6,5 secs.

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
                SlideDuration="0:0:6.5" ... />
```

#### VB:

```
slideShow.SlideDuration = TimeSpan.FromSeconds(6.5)
```

## SlidingEffect

Καθορίζει το γενικό εφέ εισόδου για όλες τις εικόνες της λίστας. Είναι τύπου **SlidingEffectEnum**.

### Σύνταξη:

#### VB:

```
Public Property SlidingEffect As SlidingEffectEnum
```

**Τύπος:** **Zeus.WPF.Controls.ImageControls.SlideShow.SlidingEffectEnum**

Προσδιορίζουμε μία τιμή **SlidingEffectEnum** που θα αφορά το γενικό εφέ εισόδου για όλες τις εικόνες της λίστας.

Η default τιμή είναι **SlidingEffectEnum.None**.

### Dependency Property Information:

**Identifier field:** **SlidingEffectProperty**

### Παρατηρήσεις:

Το εφέ αυτό θα χρησιμοποιηθεί μόνο όταν η ιδιότητα **UseSameSlidingEffectForAllImages** είναι **True**. Αλλιώς, θα χρησιμοποιηθεί το κάθε εφέ της εικόνας, όπως το έχουμε προσδιορίσει κατά την εισαγωγή της στη λίστα των εικόνων.

Αν προσδιοριστεί κάποιο εφέ, τότε θα χρησιμοποιηθεί είτε στην χειροκίνητη αλλαγή εικόνων είτε στην αυτόματη με το auto-play. Η **χρονική διάρκεια του εφέ** καθορίζεται από την ιδιότητα **SlidingEffectDuration**.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ως γενικό, ομοιόμορφο, εφέ εισόδου για όλες τις εικόνες, το **FadeIn** (πρόκειται για ένα *animation* που επενεργεί επί της ιδιότητας *Opacity* του ενσωματωμένου *Image element*, από την τιμή 0 έως 1, σε χρόνο που καθορίζεται από την ιδιότητα *SlidingEffectDuration*)

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn" ... />
```

#### VB:

```
Imports Zeus.WPF.Controls.ImageControls
```

```
slideShow.SlidingEffect = SlideShow.SlidingEffectEnum.FadeIn
```

## SlidingEffectDuration

Καθορίζει τη διάρκεια του εφέ εισόδου. Είναι τύπου **Duration**.

### Σύνταξη:

#### VB:

```
Public Property SlidingEffectDuration As Duration
```

**Τύπος:** **System.Windows.Duration**

Προσδιορίζουμε μία τιμή `Duration` που είναι η χρονική διάρκεια του εφέ. Η default τιμή είναι `TimeSpan.FromSeconds(0.3)`, δηλαδή 0,3 secs.

### Dependency Property Information:

**Identifier field:** `SlidingEffectDurationProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε χρονική διάρκεια των εφέ σε 0,5 secs.

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
    SlidingEffectDuration="0:0:0.5" ... />
```

#### VB:

```
slideShow.SlidingEffectDuration = TimeSpan.FromSeconds(0.5)
```

## SlidingMoveInEffectOffset

Καθορίζει την αρχική offset μετακίνηση της εικόνας σε σχέση με την αρχική της θέση. Π.χ. μία τιμή 100 σημαίνει 100 pixels offset κατά X ή/και Y, ανάλογα με το εφέ μετακίνησης που έχουμε επιλέξει. Είναι τύπου **Double**.

### Σύνταξη:

#### VB:

```
Public Property SlidingMovingEffectOffset As Double
```

#### Τύπος: System.Double

Προσδιορίζουμε μία τιμή Double που αφορά το offset από την αρχική θέση (κατά τον x ή/και τον y άξονα) και χρησιμοποιείται στο ανάλογο MoveInXXX εφέ .

Η default τιμή είναι 100 pixels.

### Dependency Property Information:

**Identifier field:** SlidingMovingEffectOffsetProperty

### Παρατηρήσεις:

Η τιμή της ιδιότητας αυτής **επιδρά πάνω σε όλα τα MoveInXXX** εφέ.

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε το offset μετακίνησης στα 50 pixels. Στο slideShow1 χρησιμοποιούμε το MoveInFromLeft εφέ, έτσι το offset θα είναι -50 μόνο κατά τον x-άξονα (θα εισέρχεται μόνο από αριστερά). Στο slideShow2 χρησιμοποιούμε το MoveInFromTopLeft εφέ, έτσι το offset θα είναι -50 κατά τον x-άξονα και y-άξονα (θα εισέρχεται από πάνω-αριστερά).

#### XAML:

```
<zeus:SlideShow Name="slideShow1" Background="#78909C"  
SlidingEffect="MoveInFromLeft"  
SlidingMoveInEffectOffset="50" ... />
```

```
<zeus:SlideShow Name="slideShow2" Background="#78909C"  
SlidingEffect="MoveInFromTopLeft"  
SlidingMoveInEffectOffset="50" ... />
```

**VB:**

**Imports** Zeus.WPF.Controls.ImageControls

slideShow1.**SlidingEffect** = SlideShow.SlidingEffectEnum.MoveInFromLeft

slideShow1.**SlidingMoveInEffectOffset** = 50

slideShow2.**SlidingEffect** = SlideShow.SlidingEffectEnum.MoveInFromTopLeft

slideShow2.**SlidingMoveInEffectOffset** = 50

## ToolTipTextForNextButton

Καθορίζει το κείμενο του tooltip για τα buttons "Next". Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ToolTipTextForNextButton As String
```

#### Τύπος: **System.String**

Προσδιορίζουμε μία τιμή String που θα αποτελεί το κείμενο του tooltip για τα buttons "Next".

Η default τιμή είναι "Next".

### Dependency Property Information:

**Identifier field:** ToolTipTextForNextButtonProperty

### Παρατηρήσεις:

Τα tooltips για τα buttons "Next" και "Previous" εμφανίζονται μόνο όταν η ιδιότητα **ShowToolTipsForPreviousNextButtons** είναι **True**.

### Παράδειγμα:

Στο επόμενο παράδειγμα, αλλάζουμε τα tooltips των buttons "Next" και "Previous" σε "Επόμενη" και "Προηγούμενη", αντίστοιχα.

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
    ToolTipTextForPreviousButton="Προηγούμενη"
    ToolTipTextForNextButton="Επόμενη"
    CaptionStyle="{StaticResource SlideShowCaptionStyle }"
    ImageChanged="slideShow_ImageChanged" />
```

#### VB:

```
With slideShow
```

```
    .ToolTipTextForPreviousButton = "Προηγούμενη"
    .ToolTipTextForNextButton = "Επόμενη"
```



End With



## ToolTipTextForPreviousButton

Καθορίζει το κείμενο του tooltip για τα buttons "Previous". Είναι τύπου **String**.

### Σύνταξη:

#### VB:

```
Public Property ToolTipTextForPreviousButton As String
```

#### Τύπος: **System.String**

Προσδιορίζουμε μία τιμή String που θα αποτελεί το κείμενο του tooltip για τα buttons "Previous".

Η default τιμή είναι "Previous".

### Dependency Property Information:

**Identifier field:** ToolTipTextForPreviousButtonProperty

### Παρατηρήσεις:

Τα tooltips για τα buttons "Next" και "Previous" εμφανίζονται μόνο όταν η ιδιότητα **ShowToolTipTextForPreviousNextButtons** είναι **True**.

### Παράδειγμα:

Στο επόμενο παράδειγμα, αλλάζουμε τα tooltips των buttons "Next" και "Previous" σε "Επόμενη" και "Προηγούμενη", αντίστοιχα.

#### XAML:

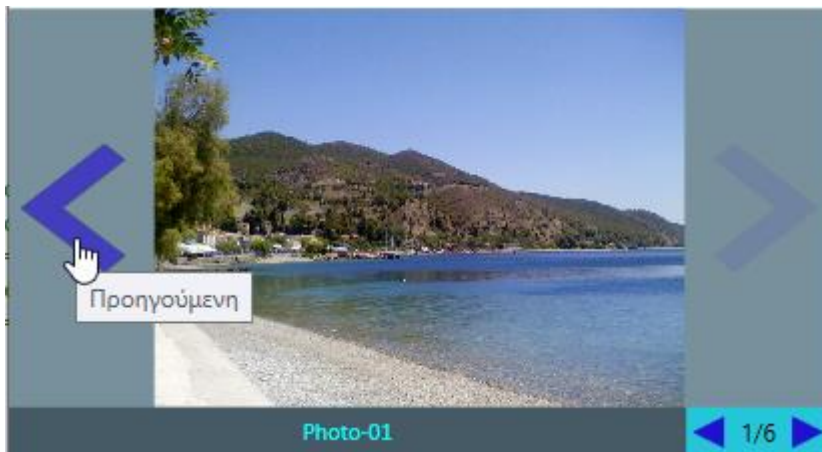
```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
    ToolTipTextForPreviousButton="Προηγούμενη"
    ToolTipTextForNextButton="Επόμενη"
    Looping="True"
    CaptionStyle="{StaticResource SlideShowCaptionStyle }"
    ImageChanged="slideShow_ImageChanged" />
```

#### VB:

```
With slideShow
```

```
    .ToolTipTextForPreviousButton = "Προηγούμενη"
    .ToolTipTextForNextButton = "Επόμενη"
```

End With



## UseSameSlidingEffectForAllImages

Καθορίζει αν θα χρησιμοποιηθεί το ίδιο εφέ, για όλες τις εικόνες, όπως έχει προσδιοριστεί στην ιδιότητα `SlidingEffect`. Είναι τύπου `Boolean`.

### Σύνταξη:

#### VB:

```
Public Property UseSameSlidingEffectForAllImages As Boolean
```

**Τύπος:** `System.Boolean`

Προσδιορίζουμε μία τιμή `Boolean` για το αν θα χρησιμοποιηθεί το γενικό εφέ, όπως έχει καθοριστεί στην ιδιότητα `SlidingEffect`.  
Η default τιμή είναι `True`.

### Dependency Property Information:

**Identifier field:** `UseSameSlidingEffectForAllImagesProperty`

### Παράδειγμα:

Στο επόμενο παράδειγμα, καθορίζουμε ότι θα χρησιμοποιηθεί διαφορετικό εφέ για κάθε εικόνα, όπως αυτό προσδιορίζεται κατά την εισαγωγή της εικόνας στη λίστα των εικόνων. Συνεπώς, η ιδιότητα `UseSameSlidingEffectForAllImages` τίθεται σε `False`.

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
    CaptionStyle="{StaticResource SlideShowCaptionStyle }"
    ImageChanged="slideShow_ImageChanged" />
```

#### VB:

```
Imports Zeus.WPF.Controls.ImageControls
```

```
Private Sub Window_Loaded(sender As Object, e As RoutedEventArgs)
```

```
    'Γεμίζουμε τη λίστα των εικόνων.
```

```
    slideShow.ImageInfoList = CreateListOfImagesWithSpecificeffectForEveryImage()
```

```
End Sub
```

```
'Δημιουργεί τη λίστα των slides. Κάθε slide περιέχει το απόλυτο path της εικόνας,
'το caption και το εφέ εισόδου.
```

```
Private Function CreateListOfImagesWithSpecificEffectForEveryImage()  
    As List(Of ImageInfo)  
  
    Dim imgList As New List(Of ImageInfo)  
  
    With imgList  
  
        .Add(New ImageInfo(AppPath & "Images\Photo-01.jpg", "Photo-01", _  
            Slideshow.SlidingEffectEnum.FadeIn))  
  
        .Add(New ImageInfo(AppPath & "Images\Photo-02.jpg", "Photo-02", _  
            Slideshow.SlidingEffectEnum.MoveInFromLeft))  
  
        .Add(New ImageInfo(AppPath & "Images\Αγριοβοτανο-1.jpg", "Αγριοβοτανο-1", _  
            Slideshow.SlidingEffectEnum.MoveInFromTopLeft))  
  
        .Add(New ImageInfo(AppPath & "Images\Αγριοβοτανο-2.jpg", "Αγριοβοτανο-2", _  
            Slideshow.SlidingEffectEnum.ZoomInFromMiddle))  
  
        .Add(New ImageInfo(AppPath & "Images\Ανδρος.jpg", "Ανδρος", _  
            Slideshow.SlidingEffectEnum.ZoomInFromTopLeft))  
  
        .Add(New ImageInfo(AppPath & "Images\Ελληνικά.jpg", "Ελληνικά", _  
            Slideshow.SlidingEffectEnum.ZoomInFromBottomRight))  
  
    End With  
  
    Return imgList  
  
End Function
```

## Συμβάντα

Όνομα	Περιγραφή
<a href="#">ImageChanged</a>	Ενεργοποιείται όταν αλλάζει η εικόνα.
<a href="#">SlideShowPaused</a>	Ενεργοποιείται όταν το auto-play slideshow έχει διακοπεί.
<a href="#">SlideShowResumed</a>	Ενεργοποιείται όταν το auto-play slideshow συνεχίζεται από το σημείο της διακοπής.
<a href="#">SlideShowStarted</a>	Ενεργοποιείται όταν το auto-play slideshow ξεκινάει από την αρχή.
<a href="#">SlideShowStopped</a>	Ενεργοποιείται όταν το auto-play slideshow έχει σταματήσει. Στην παρούσα έκδοση, είναι ίδιο με τη διακοπή.

## ImageChanged

Ενεργοποιείται **όταν αλλάζει η εικόνα**.

**Σύνταξη:**

**VB (ορισμός):**

```
Public Custom Event ImageChanged As RoutedEventHandler
```

**XAML attribute usage:**

```
<zeus:SlideShow ImageChanged="eventHanlder" ... />
```

**Παράδειγμα:**

Στο επόμενο παράδειγμα, ενημερώνουμε την ιδιότητα NavigationText με τον αριθμό της τρέχουσας εικόνας που εμφανίζεται σε συνδυασμό με το πλήθος των εικόνων, σε μορφή "x/n". Η ενημέρωση γίνεται στον ImageChanged event handler:

**XAML:**

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
    Looping="True"
    CaptionStyle="{StaticResource SlideShowCaptionStyle }"
    ImageChanged="slideShow_ImageChanged" />
```

**VB:**

```
Private Sub slideShow_ImageChanged(sender As Object, e As RoutedEventArgs)
    Dim slideShow As SlideShow = TryCast(sender, SlideShow)
    If slideShow IsNot Nothing Then
        With slideShow
            .NavigationText = String.Concat(If(.ImageInfoList.Count > 0, _
                .CurrentImageIndex + 1, 0), "/", .ImageInfoList.Count)
        End With
    End If
End Sub
```

Αρχικά, είμαστε στην 1η εικόνα:



Και μόλις πατήσουμε το button "Next" μεταβαίνουμε στην 2η εικόνα. Το ImageChanged event ενεργοποιείται και ενημερώνει το navigation text κάτω δεξιά:





## SlideShowPaused

Ενεργοποιείται όταν το auto-play slideshow έχει διακοπεί.

### Σύνταξη:

#### VB (ορισμός):

```
Public Custom Event SlideShowPaused As RoutedEventHandler
```

#### XAML attribute usage:

```
<zeus:SlideShow SlideShowPaused="eventHanlder" ... />
```

### Παράδειγμα:

Στο επόμενο παράδειγμα, απλά εμφανίζουμε ένα μήνυμα στην οθόνη ότι το auto-play slideshow διακόπηκε μόλις ο χρήστης πατήσει το κουμπί Stop/Pause:

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
    Looping="True"
    CaptionStyle="{StaticResource SlideShowCaptionStyle}"
    ImageChanged="slideShow_ImageChanged"
    SlideShowPaused="slideShow_SlideShowPaused"/>
```

#### VB:

...

```
'Καλείται η σχετική μέθοδος στο κατάλληλο σημείο. Π.χ. σε ένα button Click handler.
'Mόλις γίνει η κλήση τότε ενεργοποιείται το αντίστοιχο slideshow event.
slideShow.PauseSlideShow()
```

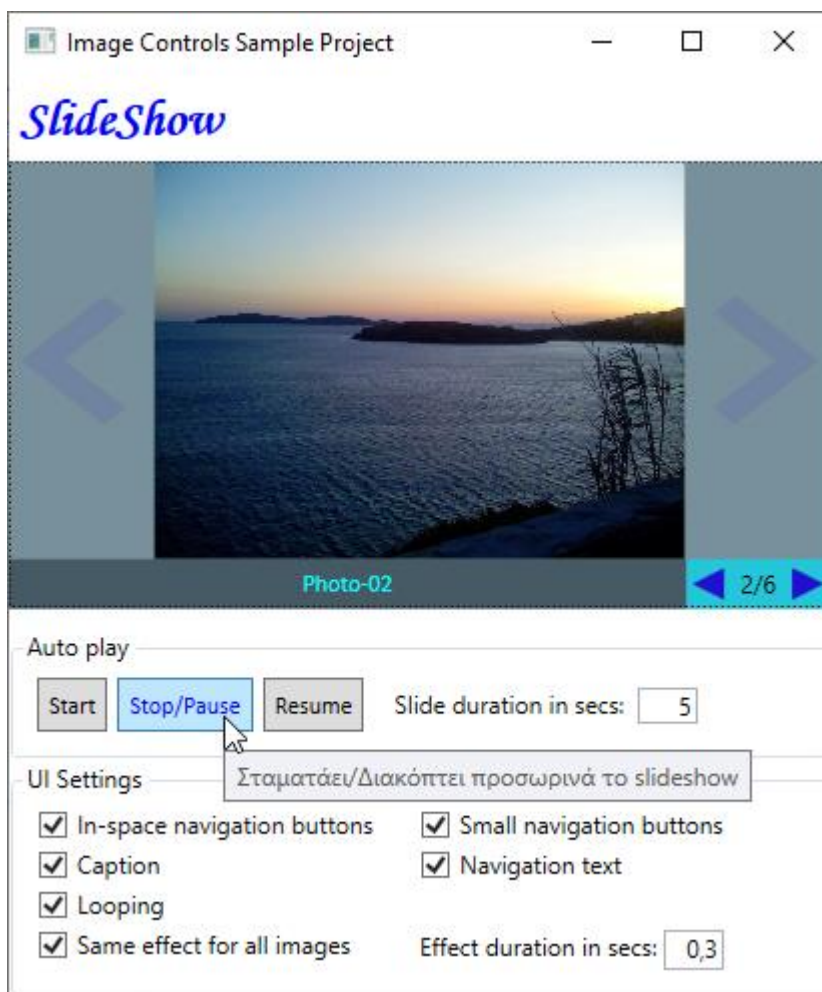
...

```
Private Sub slideShow_SlideShowPaused(sender As Object, e As RoutedEventArgs)
```

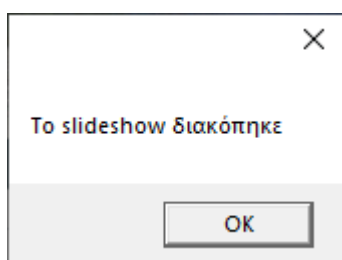
```
    MessageBox.Show("To slideshow διακόπηκε")
```

```
End Sub
```

Το slideshow έχει ξεκινήσει. Ο χρήστης πατάει το button Stop/Pause...



...και εμφανίζεται το σχετικό μήνυμα:



## SlideShowResumed

Ενεργοποιείται όταν το auto-play slideshow συνεχίζεται από το σημείο της διακοπής.

### Σύνταξη:

#### VB (ορισμός):

```
Public Custom Event SlideShowResumed As RoutedEventHandler
```

#### XAML attribute usage:

```
<zeus:SlideShow SlideShowResumed="eventHanlder" ... />
```

### Παράδειγμα:

Στο επόμενο παράδειγμα, απλά εμφανίζουμε ένα μήνυμα στην οθόνη ότι το auto-play slideshow συνεχίζεται μόλις ο χρήστης πατήσει το κουμπί Resume:

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
    Looping="True"
    CaptionStyle="{StaticResource SlideShowCaptionStyle }"
    ImageChanged="slideShow_ImageChanged"
    SlideShowResumed="slideShow_SlideShowResumed"/>
```

#### VB:

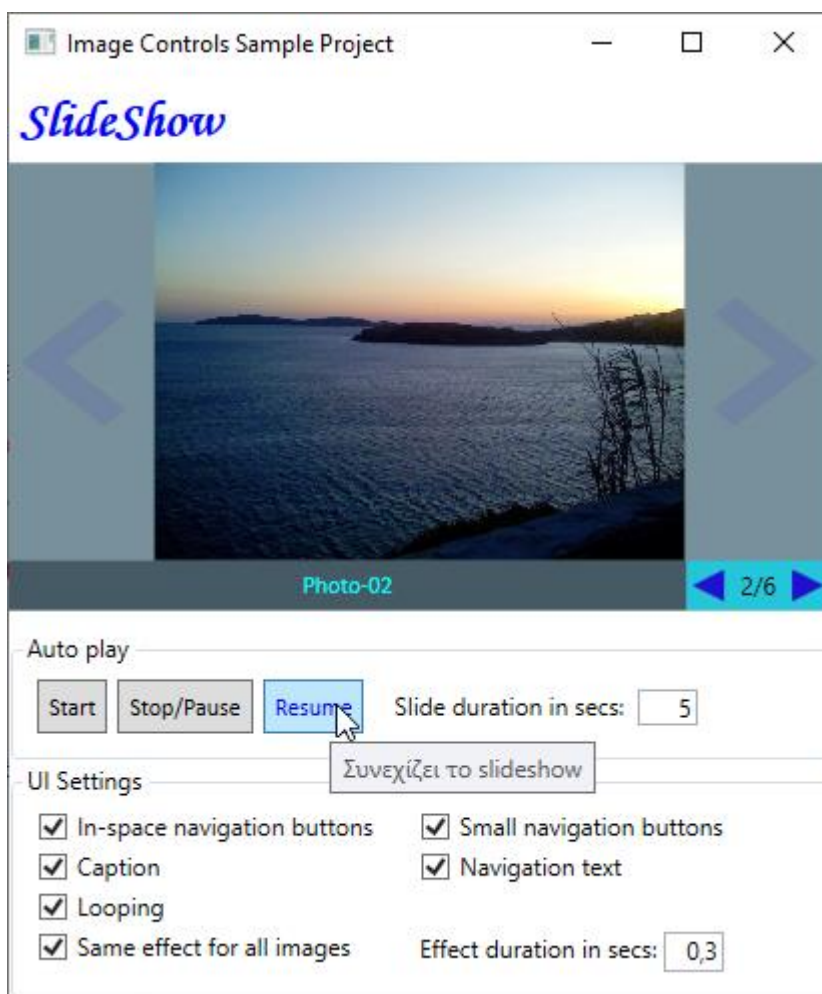
...

```
'Καλείται η σχετική μέθοδος στο κατάλληλο σημείο. Π.χ. σε ένα button Click handler.
'Mόλις γίνει η κλήση τότε ενεργοποιείται το αντίστοιχο slideshow event.
slideShow.ResumeSlideShow()
```

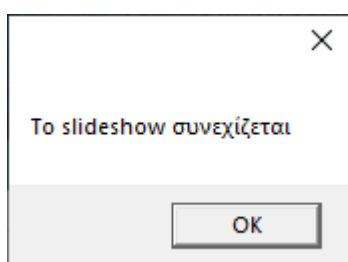
...

```
Private Sub slideShow_SlideShowResumed(sender As Object, e As RoutedEventArgs)
    MessageBox.Show("Το slideshow συνεχίζεται")
End Sub
```

Το slideshow έχει διακοπεί. Ο χρήστης πατάει το button Resume...



...και εμφανίζεται το σχετικό μήνυμα:



## SlideShowStarted

Ενεργοποιείται όταν το auto-play slideshow ξεκινάει από την αρχή.

Σύνταξη:

**VB (ορισμός):**

```
Public Custom Event SlideShowStarted As RoutedEventHandler
```

**XAML attribute usage:**

```
<zeus:SlideShow SlideShowStarted="eventHanlder" ... />
```

**Παράδειγμα:**

Στο επόμενο παράδειγμα, απλά εμφανίζουμε ένα μήνυμα στην οθόνη ότι το auto-play slideshow ξεκίνησε μόλις ο χρήστης πατήσει το κουμπί Start:

**XAML:**

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
    Looping="True"
    CaptionStyle="{StaticResource SlideShowCaptionStyle }"
    ImageChanged="slideShow_ImageChanged"
    SlideShowStarted ="slideShow_SlideShowStarted"/>
```

**VB:**

...

'Καλείται η σχετική μέθοδος στο κατάλληλο σημείο. Π.χ. σε ένα button Click handler.  
'Μόλις γίνει η κλήση τότε ενεργοποιείται το αντίστοιχο slideshow event.

```
slideShow.StartSlideShow()
```

...

```
Private Sub slideShow_SlideShowStarted(sender As Object, e As RoutedEventArgs)
```

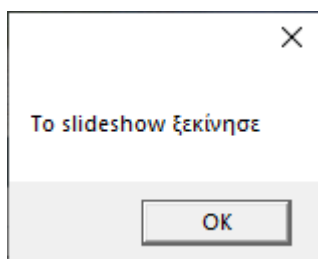
```
    MessageBox.Show("To slideshow ξεκίνησε")
```

```
End Sub
```

Ο χρήστης πατάει το button Start. Το slideshow ξεκινάει...



...και εμφανίζεται το σχετικό μήνυμα:



## SlideShowStopped

Ενεργοποιείται **όταν το auto-play slideshow έχει σταματήσει**. Στην παρούσα έκδοση, είναι ίδιο με τη διακοπή.

### Σύνταξη:

#### VB (ορισμός):

```
Public Custom Event SlideShowStopped As RoutedEventHandler
```

#### XAML attribute usage:

```
<zeus:SlideShow SlideShowStopped="eventHanlder" ... />
```

### Παράδειγμα:

Στο επόμενο παράδειγμα, απλά εμφανίζουμε ένα μήνυμα στην οθόνη ότι το auto-play slideshow σταμάτησε μόλις ο χρήστης πατήσει το κουμπί Stop/Pause:

#### XAML:

```
<zeus:SlideShow Name="slideShow" Background="#78909C" SlidingEffect="FadeIn"
    Looping="True"
    CaptionStyle="{StaticResource SlideShowCaptionStyle }"
    ImageChanged="slideShow_ImageChanged"
    SlideShowStopped="slideShow_SlideShowStopped"/>
```

#### VB:

...

'Καλείται η σχετική μέθοδος στο κατάλληλο σημείο. Π.χ. σε ένα button Click handler.  
'Μόλις γίνει η κλήση τότε ενεργοποιείται το αντίστοιχο slideshow event.

```
slideShow.StopSlideShow()
```

...

```
Private Sub slideShow_SlideShowStopped(sender As Object, e As RoutedEventArgs)
```

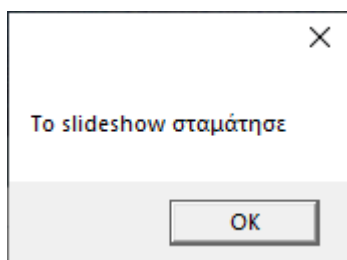
```
    MessageBox.Show("Το slideshow σταμάτησε")
```

```
End Sub
```

Το slideshow έχει ξεκινήσει. Ο χρήστης πατάει το button Stop/Pause...



...και εμφανίζεται το σχετικό μήνυμα:





## Μέθοδοι

Όνομα	Περιγραφή
<a href="#">GotoSlide(nbrSlide As Integer)</a>	<b>Μεταβαίνει στην εικόνα με βάση τον αριθμό της στη λίστα, ξεκινώντας από το 1.</b>
<a href="#">GotoSlide(slideCaption As String)</a>	<b>Μεταβαίνει στην εικόνα με βάση τη λεζάντα (caption).</b>
<a href="#">PauseSlideShow()</a>	<b>Διακόπτει το auto-play slideshow.</b> Η κλήση της μεθόδου ενεργοποιεί το συμβάν <a href="#">SlideShowPaused</a> .
<a href="#">ResumeSlideShow()</a>	<b>Συνεχίζει το auto-play slideshow από το τρέχον σημείο.</b> Η κλήση της μεθόδου ενεργοποιεί το συμβάν <a href="#">SlideShowResumed</a> .
<a href="#">StartSlideShow()</a>	<b>Ξεκινάει το auto-play slideshow από την αρχή.</b> Η κλήση της μεθόδου ενεργοποιεί το συμβάν <a href="#">SlideShowStarted</a> .
<a href="#">StopSlideShow()</a>	<b>Σταματάει το auto-play slideshow.</b> Η κλήση της μεθόδου ενεργοποιεί το συμβάν <a href="#">SlideShowStopped</a> .  Στην παρούσα έκδοση, η διαφορά της από την <a href="#">PauseSlideShow()</a> είναι μόνο σημασιολογική.

## GotoSlide(nbrSlide)

Μεταβαίνει στην εικόνα με βάση τον αριθμό της στη λίστα, ξεκινώντας από το 1.

Σύνταξη:

**VB :**

```
Public Sub GotoSlide(  
    nbrSlide As Integer  
)
```

Παράμετροι:

*nbrSlide*

Τύπος: Integer

Ο αριθμός της εικόνας στη λίστα, ξεκινώντας από το 1.

**Παράδειγμα:**

Στο επόμενο παράδειγμα, μεταβαίνουμε στην 5η εικόνα της λίστας:

**VB:**

```
slideShow.GotoSlide(5)
```

## GotoSlide(slideCaption)

Μεταβαίνει στην εικόνα με βάση τη λεζάντα (caption).

Σύνταξη:

**VB :**

```
Public Sub GotoSlide(  
    slideCaption As String  
)
```

Παράμετροι:

*slideCaption*  
Τύπος: String  
Το κείμενο της λεζάντας.

**Παράδειγμα:**

Στο επόμενο παράδειγμα, μεταβαίνουμε στην εικόνα που έχει λεζάντα "Photo-02":

**VB:**

```
slideShow.GotoSlide("photo-02")
```

- **Δεν γίνεται διάκριση κεφαλαίων-πεζών.** Έτσι, είτε δίνουμε "Photo-02" είτε "photo-02", θα ήταν το ίδιο.

## PauseSlideShow

**Διακόπτει το auto-play slideshow.** Η κλήση της μεθόδου ενεργοποιεί το συμβάν SlideShowPaused.

**Σύνταξη:**

**VB :**

```
Public Sub PauseSlideShow()
```

**Παράδειγμα:**

**VB:**

```
slideShow.PauseSlideShow()
```

## ResumeSlideShow

**Συνεχίζει το auto-play slideshow από το τρέχον σημείο.** Η κλήση της μεθόδου ενεργοποιεί το συμβάν SlideShowResumed.

### Σύνταξη:

**VB :**

```
Public Sub ResumeSlideShow()
```

### Παράδειγμα:

**VB:**

```
slideShow.ResumeSlideShow()
```

## StartSlideShow

Ξεκινάει το **auto-play slideshow** από την αρχή. Η κλήση της μεθόδου ενεργοποιεί το συμβάν `SlideShowStarted`.

**Σύνταξη:**

**VB :**

```
Public Sub StartSlideShow()
```

**Παράδειγμα:**

**VB:**

```
slideShow.StartSlideShow()
```

## StopSlideShow

**Σταματάει το auto-play slideshow.** Η κλήση της μεθόδου ενεργοποιεί το συμβάν `SlideShowStopped`.

### Σύνταξη:

**VB :**

```
Public Sub StopSlideShow()
```

### Παράδειγμα:

**VB:**

```
slideShow.StopSlideShow()
```

- Στην παρούσα έκδοση, η διαφορά της από την `PauseSlideShow()` είναι μόνο σημασιολογική.

## Styles και Templates

- Parts και States
- Το default Style και ControlTemplate
- Ένα προσαρμοσμένο Style και ControlTemplate



## Parts και States

Το default ControlTemplate περιλαμβάνει κάποια **part names** και **visual states**. Μπορείτε να τροποποιήσετε το default ControlTemplate ώστε να δώσετε στο control μία μοναδική εμφάνιση.

Ο παρακάτω πίνακας εμφανίζει τα **part names** του **SlideShow**:

Part	Τύπος	Περιγραφή
PART_Previous_InSpace	Button	Το button "Previous" στο χώρο της εικόνας (in-space).
PART_Next_InSpace	Button	Το button "Next" στο χώρο της εικόνας (in-space).
PART_Previous_Small	Button	Το button "Previous" στο χώρο της πλοήγησης (navigation area).
PART_Next_Small	Button	Το button "Next" στο χώρο της πλοήγησης (navigation area).
PART_Caption	TextBlock	Το TextBlock για την λεζάντα.
PART_NavigationText	TextBlock	Το TextBlock για το κείμενο της πλοήγησης (navigation text).
PART_Image	Image	Το Image για την εμφάνιση της εικόνας.

Ο παρακάτω πίνακας εμφανίζει τα **visual states** του **SlideShow**:

VisualState	VisualStateGroup	Περιγραφή
Normal	CommonStates	Το default state.
MouseOver	CommonStates	Το state όταν ο δείκτης του ποντικιού είναι πάνω από το control.
Disabled	CommonStates	Το state όταν το control είναι disabled.
Focused	FocusStates	Το state όταν το control έχει το focus.
Unfocused	FocusStates	Το state όταν το control δεν έχει το focus.

*Το default ControlTemplate έχει καθορισμένο μόνο το Disabled state. Μπορείτε να δημιουργήσετε ένα custom ControlTemplate για να το αλλάξετε ή/και για να καθορίσετε τα υπόλοιπα.*

## To default Style και ControlTemplate

Ο XAML κώδικας για το **default Style** και **ControlTemplate** φαίνεται παρακάτω. Μπορείτε να βασιστείτε σε αυτόν για να δημιουργήσετε μία μικρή ή μεγάλη παραλλαγή του δικού σας custom Style και ControlTemplate:

```
xmlns:z="clr-namespace:Zeus.WPF.Controls.ImageControls"

<Style TargetType="{x:Type z:SlideShow}">
    <Style.Resources >
        <!-- Styles for Previous/Next in-space buttons-->
        <Style x:Key="InSpaceButtonStyleTemplate" TargetType="{x:Type Button }">
            <Setter Property="Template" >
                <Setter.Value>
                    <ControlTemplate TargetType="{x:Type Button}">
                        <Border Name="border" Background="Transparent" >
                            <VisualStateManager.VisualStateGroups>
                                <VisualStateGroup Name="CommonStates">
                                    <VisualStateGroup.Transitions >
                                        <VisualTransition GeneratedDuration="0:0:0.1">
                                        </VisualTransition>
                                    </VisualStateGroup.Transitions>
                                    <VisualState Name="Normal">
                                        <Storyboard >
                                            <DoubleAnimation
                                                Storyboard.TargetName="border"
                                                Storyboard.TargetProperty="Opacity"
                                                To="0.1" />
                                        </Storyboard>
                                    </VisualState>
                                    <VisualState Name="MouseOver" >
                                        <Storyboard >
                                            <DoubleAnimation
                                                Storyboard.TargetName="border"
                                                Storyboard.TargetProperty="Opacity"
                                                To="0.8" />
                                        </Storyboard >
                                    </VisualState>
                                    <VisualState Name="IsPressed" >
                                        <Storyboard >
```

```

        <DoubleAnimation
            Storyboard.TargetName="border"
            Storyboard.TargetProperty="Opacity"
            To="0.8" />
    </Storyboard >
</VisualState>

    <VisualState Name="Disabled">
        <Storyboard >
            <DoubleAnimation
                Storyboard.TargetName="border"
                Storyboard.TargetProperty="Opacity"
                To="0.0" />
        </Storyboard >
    </VisualState>

</VisualStateGroup>

</VisualStateManager.VisualStateGroups>

<ContentPresenter />

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>
</Style>

<Style x:Key="InSpaceButtonStyle" TargetType="{x:Type Button }"
    BasedOn="{StaticResource InSpaceButtonStyleTemplate}">

    <Setter Property="HorizontalAlignment" Value="Center" />
    <Setter Property="Width" Value="64" />
    <Setter Property="Height" Value="64" />
    <Setter Property="Cursor" Value="Hand"/>
    <Setter Property="Background" Value="Transparent" />
    <Setter Property="BorderBrush" Value="Transparent" />

</Style>

<!-- Previous in-space default style-->
<Style x:Key="PreviousInSpaceButtonDefaultStyle" TargetType="{x:Type Button }"
    BasedOn="{StaticResource InSpaceButtonStyle }">

    <Setter Property="Content" >
        <Setter.Value>

            <Rectangle Width="64" Height="64" >
                <Rectangle.Fill>
                    <DrawingBrush Stretch="Uniform">
                        <DrawingBrush.Drawing>
                            <DrawingGroup>
                                <DrawingGroup.Children>
                                    <GeometryDrawing Brush="#FF230FD2"
Geometry="F1 M 141.399,0.499994L 176.064,27.5951L 70.4193,108.908L 176.437,189.586L
141.972,216.833L 35.4833,135.798L 35.3836,135.875L 27.9961,130.1L 0.5,109.176L
0.861164,108.891L 0.718704,108.779L 11.4417,100.526L 34.9656,81.9287L 35.2867,82.173L
141.399,0.499994 Z ">
                                <GeometryDrawing.Pen>

```

```

        <Pen LineJoin="Round"
            Brush="#FF000000"/>
        </GeometryDrawing.Pen>
    </GeometryDrawing>
</DrawingGroup.Children>
</DrawingGroup>
</DrawingBrush.Drawing>
</DrawingBrush>
</Rectangle.Fill>
</Rectangle>

</Setter.Value>
</Setter>

<Style.Triggers>

    <Trigger Property="IsMouseOver" Value="True">
        <Setter Property="Opacity" Value="0.8" />
    </Trigger>

    <Trigger Property="IsPressed" Value="True">
        <Setter Property="Opacity" Value="0.8"/>
        <Setter Property="Content" >
            <Setter.Value>
                <Rectangle Width="64" Height="64" >
                    <Rectangle.Fill>
                        <DrawingBrush Stretch="Uniform">
                            <DrawingBrush.Drawing>
                                <DrawingGroup>
                                    <DrawingGroup.Children>
                                        <GeometryDrawing
                                            Brush="#FF006432"
Geometry="F1 M 181.295,21.8333L 215.96,48.9284L 110.315,130.241L 216.333,210.919L
181.868,238.167L 75.3793,157.131L 75.2796,157.208L 67.8921,151.434L 40.396,130.51L
40.7572,130.224L 40.6147,130.113L 51.3377,121.859L 74.8616,103.262L 75.1826,103.506L
181.295,21.8333 Z ">
                                    </GeometryDrawing.Pen>
                                    <Pen LineJoin="Round"
                                        Brush="#FF000000"/>
                                    </GeometryDrawing.Pen>
                                </GeometryDrawing>
                            </DrawingGroup.Children >
                        </DrawingGroup>
                    </DrawingBrush.Drawing>
                </DrawingBrush>
            </Rectangle.Fill>
        </Rectangle>

            </Setter.Value>
        </Setter>
    </Trigger>

</Style.Triggers>

</Style>

<!-- Next in-space default style-->
<Style x:Key="NextInSpaceButtonDefaultStyle" TargetType="{x:Type Button }"
    BasedOn="{StaticResource InSpaceButtonStyle }">

```

```

<Setter Property="Content" >
  <Setter.Value>

    <Rectangle Width="64" Height="64" >
      <Rectangle.Fill>
        <DrawingBrush Stretch="Uniform">
          <DrawingBrush.Drawing>
            <DrawingGroup>
              <DrawingGroup.Children>
                <GeometryDrawing Brush="#FF230FD2"
Geometry="F1 M 75.4347,21.8333L 40.7698,48.9285L 146.414,130.241L 40.396,210.919L
74.8617,238.167L 181.35,157.131L 181.45,157.208L 188.837,151.434L 216.333,130.51L
215.972,130.224L 216.115,130.113L 205.392,121.859L 181.868,103.262L 181.547,103.506L
75.4347,21.8333 Z ">

                    <GeometryDrawing.Pen>
                      <Pen LineJoin="Round"
                        Brush="#FF000000"/>
                    </GeometryDrawing.Pen>
                  </GeometryDrawing>
                </DrawingGroup.Children>
              </DrawingGroup>
            </DrawingBrush.Drawing>
          </DrawingBrush>
        </Rectangle.Fill>
      </Rectangle>

    </Setter.Value>
  </Setter>

<Style.Triggers>

  <Trigger Property="IsMouseOver" Value="True">
    <Setter Property="Opacity" Value="0.8" />
  </Trigger>

  <Trigger Property="IsPressed" Value="True">

    <Setter Property="Opacity" Value="0.8"/>
    <Setter Property="Content" >
      <Setter.Value>

        <Rectangle Width="64" Height="64" >
          <Rectangle.Fill>
            <DrawingBrush Stretch="Uniform">
              <DrawingBrush.Drawing>
                <DrawingGroup>
                  <DrawingGroup.Children>
                    <GeometryDrawing Brush="#FF006432"
Geometry="F1 M 35.5387,0.499994L 0.873798,27.5951L 106.518,108.908L 0.500004,189.586L
34.9657,216.833L 141.454,135.798L 141.554,135.875L 148.941,130.1L 176.437,109.176L
176.076,108.891L 176.219,108.779L 165.496,100.526L 141.972,81.9287L 141.651,82.173L
35.5387,0.499994 Z ">

                            <GeometryDrawing.Pen>
                              <Pen LineJoin="Round"
                                Brush="#FF000000"/>
                            </GeometryDrawing.Pen>
                          </GeometryDrawing>
                        </DrawingGroup.Children>
                      </DrawingGroup>
                    </DrawingBrush.Drawing>
                  </DrawingBrush>
                </Rectangle.Fill>
              </Rectangle>

            </Setter.Value>
          </Setter>
        </Style.Triggers>
      </Content>
    </Setter>
  </Style.Triggers>

```

```

        </Rectangle.Fill>
    </Rectangle>

        </Setter.Value>
    </Setter>
</Trigger>

</Style.Triggers>

</Style>

<!-- Small Previous/Next buttons, in the down-right navigation area.-->

<Style x:Key="SmallButtonStyleTemplate" TargetType="{x:Type Button }">
    <Setter Property="Template" >
        <Setter.Value>
            <ControlTemplate TargetType="{x:Type Button}">
                <Border Name="border" Background="Transparent" >
                    <ContentPresenter />
                </Border>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>

<Style x:Key="SmallButtonStyle" TargetType="{x:Type Button }"
    BasedOn="{StaticResource SmallButtonStyleTemplate }" >
    <Setter Property="HorizontalAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Width" Value="22" />
    <Setter Property="Height" Value="22" />
    <Setter Property="Cursor" Value="Hand"/>
    <Setter Property="Background" Value="Transparent" />
    <Setter Property="BorderBrush" Value="Transparent" />
</Style>

<!-- Previous small default style-->
<Style x:Key="PreviousSmallButtonDefaultStyle" TargetType="{x:Type Button }"
    BasedOn="{StaticResource SmallButtonStyle }">
    <Setter Property="Content" >
        <Setter.Value>
            <Rectangle Width="16" Height="16" >
                <Rectangle.Fill>
                    <DrawingBrush Stretch="Uniform">
                        <DrawingBrush.Drawing>
                            <DrawingGroup>
                                <DrawingGroup.Children>
                                    <GeometryDrawing Brush="#FF230FD2"
Geometry="F1 M 29.8677,130.014L 223.185,225.637L 221.998,32.0276L 29.8677,130.014 Z " >
                                        <GeometryDrawing.Pen>

```

```

        <Pen LineJoin="Round"
            Brush="#FF000000"/>
        </GeometryDrawing.Pen>
    </GeometryDrawing>
</DrawingGroup.Children>
</DrawingGroup>
</DrawingBrush.Drawing>
</DrawingBrush>
</Rectangle.Fill>
</Rectangle>

</Setter.Value>
</Setter>

<Style.Triggers>

    <Trigger Property="IsMouseOver" Value="True">
        <Setter Property="Content" >
            <Setter.Value>

                <Rectangle Width="16" Height="16" >
                    <Rectangle.Fill>
                        <DrawingBrush Stretch="Uniform">
                            <DrawingBrush.Drawing>
                                <DrawingGroup>
                                    <DrawingGroup.Children>
                                        <GeometryDrawing Brush="#FF64FF00"
Geometry="F1 M 29.8677,130.014L 223.185,225.637L 221.998,32.0276L 29.8677,130.014 Z ">
                                            <GeometryDrawing.Pen>
                                                <Pen LineJoin="Round"
                                                    Brush="#FF000000"/>
                                            </GeometryDrawing.Pen>
                                        </GeometryDrawing>
                                    </DrawingGroup.Children>
                                </DrawingGroup>
                            </DrawingBrush.Drawing>
                        </DrawingBrush>
                    </Rectangle.Fill>
                </Rectangle>

                </Setter.Value>
            </Setter>
        </Trigger>

        <Trigger Property="IsPressed" Value="True">
            <Setter Property="Content" >
                <Setter.Value>

                    <Rectangle Width="16" Height="16" >
                        <Rectangle.Fill>
                            <DrawingBrush Stretch="Uniform">
                                <DrawingBrush.Drawing>
                                    <DrawingGroup>
                                        <DrawingGroup.Children>
                                            <GeometryDrawing Brush="#FF006432"
Geometry="F1 M 29.8677,130.014L 223.185,225.637L 221.998,32.0276L 29.8677,130.014 Z ">
                                                <GeometryDrawing.Pen>
                                                    <Pen LineJoin="Round"
                                                        Brush="#FF000000"/>
                                                </GeometryDrawing.Pen>
                                            </GeometryDrawing>
                                        </DrawingGroup.Children>
                                    </DrawingGroup>
                                </DrawingBrush.Drawing>
                            </DrawingBrush>
                        </Rectangle.Fill>
                    </Rectangle>

                    </Setter.Value>
                </Setter>
            </Trigger>
        </Style.Triggers>
    </Style>
</Content>

```

```

        </DrawingGroup>
        </DrawingBrush.Drawing>
    </DrawingBrush>
    </Rectangle.Fill>
</Rectangle>

    </Setter.Value>
</Setter>
</Trigger>

<Trigger Property="IsEnabled" Value="False">

    <Setter Property="Opacity" Value="0.2"/>

</Trigger>

</Style.Triggers>

</Style>

<!-- Next small default style-->
<Style x:Key="NextSmallButtonDefaultStyle" TargetType="{x:Type Button}"
    BasedOn="{StaticResource SmallButtonStyle }">

    <Setter Property="Content" >
        <Setter.Value>

            <Rectangle Width="16" Height="16" >
                <Rectangle.Fill>
                    <DrawingBrush Stretch="Uniform">
                        <DrawingBrush.Drawing>
                            <DrawingGroup>
                                <DrawingGroup.Children>
                                    <GeometryDrawing Brush="#FF230FD2"
Geometry="F1 M 222.593,129.372L 29.8147,32.6676L 29.9171,226.281L 222.593,129.372 Z " >
                                        <GeometryDrawing.Pen>
                                            <Pen LineJoin="Round"
Brush="#FF000000"/>
                                        </GeometryDrawing.Pen>
                                    </GeometryDrawing>
                                </DrawingGroup.Children>
                            </DrawingGroup>
                        </DrawingBrush.Drawing>
                    </DrawingBrush>
                </Rectangle.Fill>
            </Rectangle>

        </Setter.Value>
    </Setter>

    <Style.Triggers>

        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="Content" >
                <Setter.Value>

                    <Rectangle Width="16" Height="16" >
                        <Rectangle.Fill>
                            <DrawingBrush Stretch="Uniform">
                                <DrawingBrush.Drawing>
                                    <DrawingGroup>
                                        <DrawingGroup.Children>

```



```

        <GeometryDrawing Brush="#FF64FF00"
Geometry="F1 M 222.593,129.372L 29.8147,32.6676L 29.9171,226.281L 222.593,129.372 Z " >
        <GeometryDrawing.Pen>
            <Pen LineJoin="Round"
                Brush="#FF000000"/>
        </GeometryDrawing.Pen>
        </GeometryDrawing>
    </DrawingGroup.Children>
</DrawingGroup>
</DrawingBrush.Drawing>
</DrawingBrush>
</Rectangle.Fill>
</Rectangle>

    </Setter.Value>
</Setter>
</Trigger>

<Trigger Property="IsPressed" Value="True">
    <Setter Property="Content" >
        <Setter.Value>

            <Rectangle Width="16" Height="16" >
                <Rectangle.Fill>
                    <DrawingBrush Stretch="Uniform">
                        <DrawingBrush.Drawing>
                            <DrawingGroup>
                                <DrawingGroup.Children>
                                    <GeometryDrawing Brush="#FF006432"
Geometry="F1 M 222.593,129.372L 29.8147,32.6676L 29.9171,226.281L 222.593,129.372 Z " >
                                    <GeometryDrawing.Pen>
                                        <Pen LineJoin="Round"
                                            Brush="#FF000000"/>
                                    </GeometryDrawing.Pen>
                                    </GeometryDrawing>
                                </DrawingGroup.Children>
                            </DrawingGroup>
                        </DrawingBrush.Drawing>
                    </DrawingBrush>
                </Rectangle.Fill>
            </Rectangle>

            </Setter.Value>
        </Setter>
    </Trigger>

    <Trigger Property="IsEnabled" Value="False">

        <Setter Property="Opacity" Value="0.2"/>

    </Trigger>

</Style.Triggers>

</Style>

<!-- Caption default styles -->
<Style x:Key="CaptionBorderDefaultStyle" TargetType="{x:Type Border }">

    <Setter Property="Background" Value="#455A64" />
    <Setter Property="Border.Padding" Value="5" />

```

```
</Style>

<Style x:Key="CaptionDefaultStyle" TargetType="{x:Type TextBlock }">
    <Setter Property="Background" Value="Transparent" />
    <Setter Property="Foreground" Value="White" />
    <Setter Property="FontSize" Value="12" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="TextWrapping" Value="Wrap" />
</Style>

<!-- Navigation area default border style -->
<Style x:Key="NavigationAreaDefaultStyle" TargetType="{x:Type Border }">
    <Setter Property="Background" Value="#26C6DA" />
    <Setter Property="Border.Padding" Value="0" />
    <Setter Property="Margin" Value="0,0,0,0" />
</Style>

<!-- NavigationText default styles -->
<Style x:Key="NavigationTextDefaultStyle" TargetType="{x:Type TextBlock }">
    <Setter Property="Background" Value="Transparent" />
    <Setter Property="Foreground" Value="Black" />
    <Setter Property="FontSize" Value="12" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Center" />
    <Setter Property="Margin" Value="5,0" />
    <Setter Property="TextWrapping" Value="Wrap" />
</Style>

</Style.Resources>

<Setter Property="PreviousButtonInSpaceStyle"
    Value="{StaticResource PreviousInSpaceButtonDefaultStyle}"/>

<Setter Property="NextButtonInSpaceStyle"
    Value="{StaticResource NextInSpaceButtonDefaultStyle}"/>

<Setter Property="PreviousButtonSmallStyle"
    Value="{StaticResource PreviousSmallButtonDefaultStyle}"/>

<Setter Property="NextButtonSmallStyle"
    Value="{StaticResource NextSmallButtonDefaultStyle}"/>

<Setter Property="CaptionBorderStyle"
    Value="{StaticResource CaptionBorderDefaultStyle}"/>

<Setter Property="CaptionStyle" Value="{StaticResource CaptionDefaultStyle}"/>

<Setter Property="NavigationAreaBorderStyle"
    Value="{StaticResource NavigationAreaDefaultStyle}"/>

<Setter Property="NavigationTextStyle"
    Value="{StaticResource NavigationTextDefaultStyle}"/>

<!-- ===== -->
<!-- The ControlTemplate of the control -->
<!-- ===== -->
```

```
<Setter Property="Template">
  <Setter.Value>
    <ControlTemplate TargetType="{x:Type z:SlideShow}">
      <ControlTemplate.Resources >
        <z:BooleanToVisibilityConverter
          x:Key="booleanToVisibilityConverter"/>
      </ControlTemplate.Resources>
      <Border Background="{TemplateBinding Background}"
        BorderBrush="{TemplateBinding BorderBrush}"
        BorderThickness="{TemplateBinding BorderThickness}">
        <!-- Visual States -->
        <VisualStateManager.VisualStateGroups >
          <VisualStateGroup Name="CommonStates">
            <VisualStateGroup.Transitions >
              <VisualTransition GeneratedDuration="0:0:0.2">
              </VisualTransition>
            </VisualStateGroup.Transitions>
            <VisualState Name="Normal"/>
            <VisualState Name="MouseOver" />
            <VisualState Name="Disabled">
              <Storyboard >
                <DoubleAnimation
                  Storyboard.TargetName="grdContainer"
                  Storyboard.TargetProperty="(Grid.Opacity)"
                  To="0.4" />
              </Storyboard>
            </VisualState>
          </VisualStateGroup>
          <VisualStateGroup Name="FocusStates">
            <VisualStateGroup.Transitions >
              <VisualTransition GeneratedDuration="0:0:0.2">
              </VisualTransition>
            </VisualStateGroup.Transitions>
            <VisualState Name="Focused" />
            <VisualState Name="Unfocused"/>
          </VisualStateGroup>
        </VisualStateManager.VisualStateGroups>
        <!-- Content -->
        <Grid Name="grdContainer" Margin="{TemplateBinding Padding}"
          Background="Transparent"
          IsEnabled="{TemplateBinding IsEnabled }">
          <Grid.ColumnDefinitions >
            <ColumnDefinition Width="Auto"/>
            <ColumnDefinition Width="*/>
            <ColumnDefinition Width="Auto"/>
          </Grid.ColumnDefinitions >
        </Grid>
      </Border>
    </ControlTemplate>
  </Setter.Value>
</Setter>
```

```

</Grid.ColumnDefinitions>

<Grid.RowDefinitions >
    <RowDefinition Height="*" />
    <RowDefinition Height="Auto" />
</Grid.RowDefinitions>

<!-- Image container -->
<Image Name="PART_Image" Grid.ColumnSpan="3"
        Stretch="Uniform" />

<!-- Buttons Previous/Next (inside image space)-->
<Button Name="PART_Previous_InSpace"
        Style="{TemplateBinding PreviousButtonInSpaceStyle}"
        Visibility="{TemplateBinding ShowButtonsInSpace,
        Converter={StaticResource booleanToVisibilityConverter}}"/>

<Button Name="PART_Next_InSpace" Grid.Column="2"
        Style="{TemplateBinding NextButtonInSpaceStyle}"
        Visibility="{TemplateBinding ShowButtonsInSpace,
        Converter={StaticResource booleanToVisibilityConverter}}"/>

<!-- 2nd row. Caption and small buttons previous/next.-->
<Grid Name="grdCaption" Grid.Row="1" Grid.ColumnSpan="3">

    <Grid.ColumnDefinitions >
        <ColumnDefinition Width="*" />
        <ColumnDefinition Width="Auto" />
    </Grid.ColumnDefinitions>

    <Border Style="{TemplateBinding CaptionBorderStyle}"
            Visibility="{TemplateBinding ShowCaption,
            Converter={StaticResource booleanToVisibilityConverter}}">
        <TextBlock Name="PART_Caption"
            Style="{TemplateBinding CaptionStyle}"
            Text="{TemplateBinding Caption }" />
    </Border>

    <Border Name="borderNavigationArea"
            Style="{TemplateBinding NavigationAreaBorderStyle }"
            Grid.Column="1">
        <StackPanel Name="stkSmallNavigationButtons"
            Orientation="Horizontal" >

            <Button Name="PART_Previous_Small"
                Style="{TemplateBinding PreviousButtonSmallStyle}"
                Visibility="{TemplateBinding ShowButtonsSmall,
                Converter={StaticResource booleanToVisibilityConverter}}"/>

            <TextBlock Name="PART_NavigationText"
                Text="{TemplateBinding NavigationText}"
                Style="{TemplateBinding NavigationTextStyle}"
                Visibility="{TemplateBinding ShowNavigationText,
                Converter={StaticResource booleanToVisibilityConverter}}"/>

            <Button Name="PART_Next_Small"
                Style="{TemplateBinding NextButtonSmallStyle}"
                Visibility="{TemplateBinding ShowButtonsSmall,
                Converter={StaticResource booleanToVisibilityConverter}}"/>

        </StackPanel>
    </Border>

```

```
        </Grid>
    </Grid>

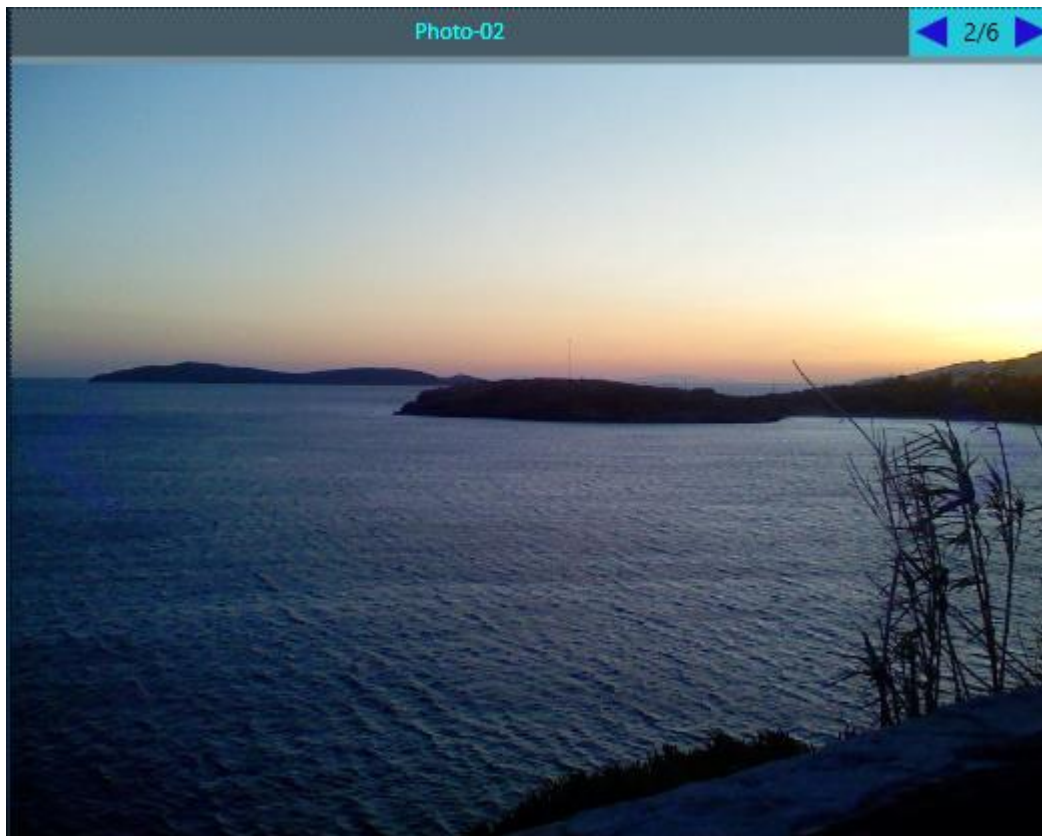
    </Border>
</ControlTemplate>

</Setter.Value>
</Setter>
</Style>
```

- Να σημειώσουμε ότι ο XAML κώδικας των button styles, που αφορά την ιδιότητα Content, παράγεται αυτόματα από την ελεύθερη εφαρμογή **Microsoft Expression Design 4**. Σχεδιάσα με τα εργαλεία του τα γραφικά των buttons και τα έκανα εξαγωγή σε XAML Resource Dictionary. Η γεωμετρία των γραφικών περιγράφεται στην Path μικρο-γλώσσα του WPF και τίθεται στην ιδιότητα Geometry. Αναλυτικότερα, μπορείτε να βρείτε στο βιβλίο μου *"Μάθετε το WPF με τη Visual Basic"*, έκδοση 2018.

## Ένα προσαρμοσμένο Style και ControlTemplate

Παρακάτω, δημιουργούμε ένα ελαφρώς προσαρμοσμένο style κατά το οποίο η λεζάντα και ο χώρος πλοήγησης εμφανίζονται στο πάνω μέρος.



Πολύ λίγες τροποποιήσεις θα κάνουμε στο τμήμα **Content**, τις οποίες τις έχουμε μαρκάρι με **κίτρινο φόντο**. Φυσικά, κάποιος μπορεί να επιφέρει περισσότερες "αισθητικές" τροποποιήσεις.

```
xmlns:z="clr-namespace:Zeus.WPF.Controls.ImageControls"

<Style TargetType="{x:Type z:SlideShow}">
    <Style.Resources >
        <!-- Styles for Previous/Next in-space buttons-->
        <Style x:Key="InSpaceButtonStyleTemplate" TargetType="{x:Type Button }">
            <Setter Property="Template" >
                <Setter.Value>
                    <ControlTemplate TargetType="{x:Type Button}">
                        <Border Name="border" Background="Transparent" >
                            <VisualStateManager.VisualStateGroups>
```

```
<VisualStateGroup Name="CommonStates">
    <VisualStateGroup.Transitions >
        <VisualTransition GeneratedDuration="0:0:0.1">
        </VisualTransition>
    </VisualStateGroup.Transitions>
    <VisualState Name="Normal">
        <Storyboard >
            <DoubleAnimation
                Storyboard.TargetName="border"
                Storyboard.TargetProperty="Opacity"
                To="0.1" />
        </Storyboard>
    </VisualState>
    <VisualState Name="MouseOver" >
        <Storyboard >
            <DoubleAnimation
                Storyboard.TargetName="border"
                Storyboard.TargetProperty="Opacity"
                To="0.8" />
        </Storyboard >
    </VisualState>
    <VisualState Name="IsPressed" >
        <Storyboard >
            <DoubleAnimation
                Storyboard.TargetName="border"
                Storyboard.TargetProperty="Opacity"
                To="0.8" />
        </Storyboard >
    </VisualState>
    <VisualState Name="Disabled">
        <Storyboard >
            <DoubleAnimation
                Storyboard.TargetName="border"
                Storyboard.TargetProperty="Opacity"
                To="0.0" />
        </Storyboard >
    </VisualState>
</VisualStateGroup>
</VisualStateManager.VisualStateGroups>
<ContentPresenter />
</Border>
</ControlTemplate>
</Setter.Value>
</Setter>
</Style>
<Style x:Key="InSpaceButtonStyle" TargetType="{x:Type Button }">
```

```

        BasedOn="{StaticResource InSpaceButtonStyleTemplate}">
        <Setter Property="HorizontalAlignment" Value="Center" />
        <Setter Property="Width" Value="64" />
        <Setter Property="Height" Value="64" />
        <Setter Property="Cursor" Value="Hand"/>
        <Setter Property="Background" Value="Transparent" />
        <Setter Property="BorderBrush" Value="Transparent" />
    </Style>

<!-- Previous in-space default style-->
<Style x:Key="PreviousInSpaceButtonDefaultStyle" TargetType="{x:Type Button }"
        BasedOn="{StaticResource InSpaceButtonStyle }">

    <Setter Property="Content" >
        <Setter.Value>

            <Rectangle Width="64" Height="64" >
                <Rectangle.Fill>
                    <DrawingBrush Stretch="Uniform">
                        <DrawingBrush.Drawing>
                            <DrawingGroup>
                                <DrawingGroup.Children>
                                    <GeometryDrawing Brush="#FF230FD2"
Geometry="F1 M 141.399,0.499994L 176.064,27.5951L 70.4193,108.908L 176.437,189.586L
141.972,216.833L 35.4833,135.798L 35.3836,135.875L 27.9961,130.1L 0.5,109.176L
0.861164,108.891L 0.718704,108.779L 11.4417,100.526L 34.9656,81.9287L 35.2867,82.173L
141.399,0.499994 Z " >
                                        <GeometryDrawing.Pen>
                                            <Pen LineJoin="Round"
Brush="#FF000000"/>
                                        </GeometryDrawing.Pen>
                                    </GeometryDrawing>
                                </DrawingGroup.Children>
                            </DrawingGroup>
                        </DrawingBrush.Drawing>
                    </DrawingBrush>
                </Rectangle.Fill>
            </Rectangle>

        </Setter.Value>
    </Setter>

    <Style.Triggers>

        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="Opacity" Value="0.8" />
        </Trigger>

        <Trigger Property="IsPressed" Value ="True">

            <Setter Property="Opacity" Value="0.8"/>
            <Setter Property="Content" >
                <Setter.Value>

                    <Rectangle Width="64" Height="64" >
                        <Rectangle.Fill>
                            <DrawingBrush Stretch="Uniform">
                                <DrawingBrush.Drawing>
                                    <DrawingGroup>

```



```

        <DrawingGroup.Children>
            <GeometryDrawing
                Brush="#FF006432"
Geometry="F1 M 181.295,21.8333L 215.96,48.9284L 110.315,130.241L 216.333,210.919L
181.868,238.167L 75.3793,157.131L 75.2796,157.208L 67.8921,151.434L 40.396,130.51L
40.7572,130.224L 40.6147,130.113L 51.3377,121.859L 74.8616,103.262L 75.1826,103.506L
181.295,21.8333 Z " >
                <GeometryDrawing.Pen>
                    <Pen LineJoin="Round"
                        Brush="#FF000000"/>
                </GeometryDrawing.Pen>
            </GeometryDrawing>
        </DrawingGroup.Children >
    </DrawingGroup>
</DrawingBrush.Drawing>
</DrawingBrush>
</Rectangle.Fill>
</Rectangle>

        </Setter.Value>
    </Setter>
</Trigger>

</Style.Triggers>
</Style>

<!-- Next in-space default style-->
<Style x:Key="NextInSpaceButtonDefaultStyle" TargetType="{x:Type Button }"
    BasedOn="{StaticResource InSpaceButtonStyle }">

    <Setter Property="Content" >
        <Setter.Value>

            <Rectangle Width="64" Height="64" >
                <Rectangle.Fill>
                    <DrawingBrush Stretch="Uniform">
                        <DrawingBrush.Drawing>
                            <DrawingGroup>
                                <DrawingGroup.Children>
                                    <GeometryDrawing Brush="#FF230FD2"
Geometry="F1 M 75.4347,21.8333L 40.7698,48.9285L 146.414,130.241L 40.396,210.919L
74.8617,238.167L 181.35,157.131L 181.45,157.208L 188.837,151.434L 216.333,130.51L
215.972,130.224L 216.115,130.113L 205.392,121.859L 181.868,103.262L 181.547,103.506L
75.4347,21.8333 Z " >
                                        <GeometryDrawing.Pen>
                                            <Pen LineJoin="Round"
                                                Brush="#FF000000"/>
                                        </GeometryDrawing.Pen>
                                    </GeometryDrawing>
                                </DrawingGroup.Children>
                            </DrawingGroup>
                        </DrawingBrush.Drawing>
                    </DrawingBrush>
                </Rectangle.Fill>
            </Rectangle>

            </Setter.Value>
        </Setter>

    <Style.Triggers>

```

```

<Trigger Property="IsMouseOver" Value="True">
  <Setter Property="Opacity" Value="0.8" />
</Trigger>

<Trigger Property="IsPressed" Value="True">
  <Setter Property="Opacity" Value="0.8"/>
  <Setter Property="Content" >
    <Setter.Value>

      <Rectangle Width="64" Height="64" >
        <Rectangle.Fill>
          <DrawingBrush Stretch="Uniform">
            <DrawingBrush.Drawing>
              <DrawingGroup>
                <DrawingGroup.Children>
                  <GeometryDrawing Brush="#FF006432"
Geometry="F1 M 35.5387,0.499994L 0.873798,27.5951L 106.518,108.908L 0.500004,189.586L
34.9657,216.833L 141.454,135.798L 141.554,135.875L 148.941,130.1L 176.437,109.176L
176.076,108.891L 176.219,108.779L 165.496,100.526L 141.972,81.9287L 141.651,82.173L
35.5387,0.499994 Z ">
                    <GeometryDrawing.Pen>
                      <Pen LineJoin="Round"
Brush="#FF000000"/>
                    </GeometryDrawing.Pen>
                  </GeometryDrawing>
                </DrawingGroup.Children>
              </DrawingGroup>
            </DrawingBrush.Drawing>
          </DrawingBrush>
        </Rectangle.Fill>
      </Rectangle>

    </Setter.Value>
  </Setter>
</Trigger>

</Style.Triggers>

</Style>

<!-- Small Previous/Next buttons, in the down-right navigation area.-->

<Style x:Key="SmallButtonStyleTemplate" TargetType="{x:Type Button }">
  <Setter Property="Template" >
    <Setter.Value>
      <ControlTemplate TargetType="{x:Type Button}">
        <Border Name="border" Background="Transparent" >
          <ContentPresenter />
        </Border>
      </ControlTemplate>
    </Setter.Value>
  </Setter>
</Style>

```

```

<Style x:Key="SmallButtonStyle" TargetType="{x:Type Button }"
      BasedOn="{StaticResource SmallButtonStyleTemplate }" >

    <Setter Property="HorizontalAlignment" Value="Center" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="Width" Value="22" />
    <Setter Property="Height" Value="22" />
    <Setter Property="Cursor" Value="Hand"/>
    <Setter Property="Background" Value="Transparent" />
    <Setter Property="BorderBrush" Value="Transparent" />

</Style>

<!-- Previous small default style-->
<Style x:Key="PreviousSmallButtonDefaultStyle" TargetType="{x:Type Button }"
      BasedOn="{StaticResource SmallButtonStyle }">

    <Setter Property="Content" >
        <Setter.Value>

            <Rectangle Width="16" Height="16" >
                <Rectangle.Fill>
                    <DrawingBrush Stretch="Uniform">
                        <DrawingBrush.Drawing>
                            <DrawingGroup>
                                <DrawingGroup.Children>
                                    <GeometryDrawing Brush="#FF230FD2"
Geometry="F1 M 29.8677,130.014L 223.185,225.637L 221.998,32.0276L 29.8677,130.014 Z " >
                                        <GeometryDrawing.Pen>
                                            <Pen LineJoin="Round"
Brush="#FF000000"/>
                                        </GeometryDrawing.Pen>
                                    </GeometryDrawing>
                                </DrawingGroup.Children>
                            </DrawingGroup>
                        </DrawingBrush.Drawing>
                    </DrawingBrush>
                </Rectangle.Fill>
            </Rectangle>

        </Setter.Value>
    </Setter>

    <Style.Triggers>

        <Trigger Property="IsMouseOver" Value="True">
            <Setter Property="Content" >
                <Setter.Value>

                    <Rectangle Width="16" Height="16" >
                        <Rectangle.Fill>
                            <DrawingBrush Stretch="Uniform">
                                <DrawingBrush.Drawing>
                                    <DrawingGroup>
                                        <DrawingGroup.Children>
                                            <GeometryDrawing Brush="#FF64FF00"
Geometry="F1 M 29.8677,130.014L 223.185,225.637L 221.998,32.0276L 29.8677,130.014 Z " >
                                                <GeometryDrawing.Pen>
                                                    <Pen LineJoin="Round"
Brush="#FF000000"/>
                                                </GeometryDrawing.Pen>
                                            </GeometryDrawing>
                                        </DrawingGroup.Children>
                                    </DrawingGroup>
                                </DrawingBrush.Drawing>
                            </DrawingBrush>
                        </Rectangle.Fill>
                    </Rectangle>

                </Setter.Value>
            </Setter>
        </Trigger>
    </Style.Triggers>

```

```

        </GeometryDrawing>
        </DrawingGroup.Children>
    </DrawingGroup>
    </DrawingBrush.Drawing>
</DrawingBrush>
</Rectangle.Fill>
</Rectangle>

    </Setter.Value>
</Setter>
</Trigger>

<Trigger Property="IsPressed" Value="True">
    <Setter Property="Content" >
        <Setter.Value>

            <Rectangle Width="16" Height="16" >
                <Rectangle.Fill>
                    <DrawingBrush Stretch="Uniform">
                        <DrawingBrush.Drawing>
                            <DrawingGroup>
                                <DrawingGroup.Children>
                                    <GeometryDrawing Brush="#FF006432"
Geometry="F1 M 29.8677,130.014L 223.185,225.637L 221.998,32.0276L 29.8677,130.014 Z ">
                                        <GeometryDrawing.Pen>
                                            <Pen LineJoin="Round"
Brush="#FF000000"/>
                                        </GeometryDrawing.Pen>
                                    </GeometryDrawing>
                                </DrawingGroup.Children>
                            </DrawingGroup>
                        </DrawingBrush.Drawing>
                    </DrawingBrush>
                </Rectangle.Fill>
            </Rectangle>

            </Setter.Value>
        </Setter>
    </Trigger>

    <Trigger Property="IsEnabled" Value="False">
        <Setter Property="Opacity" Value="0.2"/>
    </Trigger>
</Style.Triggers>
</Style>
<!-- Next small default style-->
<Style x:Key="NextSmallButtonDefaultStyle" TargetType="{x:Type Button }"
BasedOn="{StaticResource SmallButtonStyle }">
    <Setter Property="Content" >
        <Setter.Value>

            <Rectangle Width="16" Height="16" >
                <Rectangle.Fill>
                    <DrawingBrush Stretch="Uniform">
                        <DrawingBrush.Drawing>
                            <DrawingGroup>

```

```

        <DrawingGroup.Children>
            <GeometryDrawing Brush="#FF230FD2"
Geometry="F1 M 222.593,129.372L 29.8147,32.6676L 29.9171,226.281L 222.593,129.372 Z " >
                <GeometryDrawing.Pen>
                    <Pen LineJoin="Round"
                        Brush="#FF000000"/>
                </GeometryDrawing.Pen>
            </GeometryDrawing>
        </DrawingGroup.Children>
    </DrawingGroup>
</DrawingBrush.Drawing>
</DrawingBrush>
</Rectangle.Fill>
</Rectangle>

</Setter.Value>
</Setter>

<Style.Triggers>

    <Trigger Property="IsMouseOver" Value="True">
        <Setter Property="Content" >
            <Setter.Value>

                <Rectangle Width="16" Height="16" >
                    <Rectangle.Fill>
                        <DrawingBrush Stretch="Uniform">
                            <DrawingBrush.Drawing>
                                <DrawingGroup>
                                    <DrawingGroup.Children>
                                        <GeometryDrawing Brush="#FF64FF00"
Geometry="F1 M 222.593,129.372L 29.8147,32.6676L 29.9171,226.281L 222.593,129.372 Z " >
                                            <GeometryDrawing.Pen>
                                                <Pen LineJoin="Round"
                                                    Brush="#FF000000"/>
                                            </GeometryDrawing.Pen>
                                        </GeometryDrawing>
                                    </DrawingGroup.Children>
                                </DrawingGroup>
                            </DrawingBrush.Drawing>
                        </DrawingBrush>
                    </Rectangle.Fill>
                </Rectangle>

            </Setter.Value>
        </Setter>
    </Trigger>

    <Trigger Property="IsPressed" Value="True">
        <Setter Property="Content" >
            <Setter.Value>

                <Rectangle Width="16" Height="16" >
                    <Rectangle.Fill>
                        <DrawingBrush Stretch="Uniform">
                            <DrawingBrush.Drawing>
                                <DrawingGroup>
                                    <DrawingGroup.Children>
                                        <GeometryDrawing Brush="#FF006432"
Geometry="F1 M 222.593,129.372L 29.8147,32.6676L 29.9171,226.281L 222.593,129.372 Z " >
                                            <GeometryDrawing.Pen>
                                                <Pen LineJoin="Round"

```

```

        Brush="#FF000000"/>
        </GeometryDrawing.Pen>
    </GeometryDrawing>
</DrawingGroup.Children>
</DrawingGroup>
</DrawingBrush.Drawing>
</DrawingBrush>
</Rectangle.Fill>
</Rectangle>

    </Setter.Value>
</Setter>
</Trigger>

<Trigger Property="IsEnabled" Value="False">

    <Setter Property="Opacity" Value="0.2"/>

</Trigger>

</Style.Triggers>

</Style>

<!-- Caption default styles -->
<Style x:Key="CaptionBorderDefaultStyle" TargetType="{x:Type Border }">

    <Setter Property="Background" Value="#455A64" />
    <Setter Property="Border.Padding" Value="5" />

</Style>

<Style x:Key="CaptionDefaultStyle" TargetType="{x:Type TextBlock }">

    <Setter Property="Background" Value="Transparent" />
    <Setter Property="Foreground" Value="White" />
    <Setter Property="FontSize" Value="12" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="TextWrapping" Value="Wrap" />

</Style>

<!-- Navigation area default border style -->
<Style x:Key="NavigationAreaDefaultStyle" TargetType="{x:Type Border }">

    <Setter Property="Background" Value="#26C6DA" />
    <Setter Property="Border.Padding" Value="0" />
    <Setter Property="Margin" Value="0,0,0,0" />

</Style>

<!-- NavigationText default styles -->
<Style x:Key="NavigationTextDefaultStyle" TargetType="{x:Type TextBlock }">

    <Setter Property="Background" Value="Transparent" />
    <Setter Property="Foreground" Value="Black" />
    <Setter Property="FontSize" Value="12" />
    <Setter Property="VerticalAlignment" Value="Center" />
    <Setter Property="HorizontalAlignment" Value="Center" />
    <Setter Property="Margin" Value="5,0" />
    <Setter Property="TextWrapping" Value="Wrap" />

```

```

    </Style>

</Style.Resources>

<Setter Property="PreviousButtonInSpaceStyle"
    Value="{StaticResource PreviousInSpaceButtonDefaultStyle}"/>

<Setter Property="NextButtonInSpaceStyle"
    Value="{StaticResource NextInSpaceButtonDefaultStyle}"/>

<Setter Property="PreviousButtonSmallStyle"
    Value="{StaticResource PreviousSmallButtonDefaultStyle}"/>

<Setter Property="NextButtonSmallStyle"
    Value="{StaticResource NextSmallButtonDefaultStyle}"/>

<Setter Property="CaptionBorderStyle"
    Value="{StaticResource CaptionBorderDefaultStyle}"/>

<Setter Property="CaptionStyle" Value="{StaticResource CaptionDefaultStyle}"/>

<Setter Property="NavigationAreaBorderStyle"
    Value="{StaticResource NavigationAreaDefaultStyle}"/>

<Setter Property="NavigationTextStyle"
    Value="{StaticResource NavigationTextDefaultStyle}"/>

<!-- ===== -->
<!-- The ControlTemplate of the control -->
<!-- ===== -->

<Setter Property="Template">

    <Setter.Value>

        <ControlTemplate TargetType="{x:Type z:SlideShow}">

            <ControlTemplate.Resources >
                <z:BooleanToVisibilityConverter
                    x:Key="booleanToVisibilityConverter"/>
            </ControlTemplate.Resources>

            <Border Background="{TemplateBinding Background}"
                BorderBrush="{TemplateBinding BorderBrush}"
                BorderThickness="{TemplateBinding BorderThickness}">

                <!-- Visual States -->
                <VisualStateManager.VisualStateGroups >

                    <VisualStateGroup Name="CommonStates">

                        <VisualStateGroup.Transitions >
                            <VisualTransition GeneratedDuration="0:0:0.2">
                                </VisualTransition>
                            </VisualStateGroup.Transitions>

                        <VisualState Name="Normal"/>
                        <VisualState Name="MouseOver" />
                        <VisualState Name="Disabled">
                            <Storyboard >
                                <DoubleAnimation
                                    Storyboard.TargetName="grdContainer"

```

```

        Storyboard.TargetProperty="(Grid.Opacity)"
        To="0.4" />
    </Storyboard>
</VisualState>
</VisualStateGroup>

<VisualStateGroup Name="FocusStates">

    <VisualStateGroup.Transitions >
        <VisualTransition GeneratedDuration="0:0:0.2">
        </VisualTransition>
    </VisualStateGroup.Transitions>

    <VisualState Name="Focused" />
    <VisualState Name="Unfocused"/>

</VisualStateGroup>

</VisualStateManager.VisualStateGroups>

<!-- Content -->

<Grid Name="grdContainer" Margin="{TemplateBinding Padding}"
    Background="Transparent"
    IsEnabled="{TemplateBinding IsEnabled }">

    <Grid.ColumnDefinitions >
        <ColumnDefinition Width="Auto"/>
        <ColumnDefinition Width="*"/>
        <ColumnDefinition Width="Auto"/>
    </Grid.ColumnDefinitions>

    <Grid.RowDefinitions >
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="*"/>
    </Grid.RowDefinitions>

    <!-- Image container -->
    <Image Name="PART_Image" Grid.Row="1" Grid.ColumnSpan="3"
        Stretch="Uniform" />

    <!-- Buttons Previous/Next (inside image space)-->
    <Button Name="PART_Previous_InSpace" Grid.Row="1"
        Style="{TemplateBinding PreviousButtonInSpaceStyle}"
        Visibility="{TemplateBinding ShowButtonsInSpace,
            Converter={StaticResource booleanToVisibilityConverter}}"/>

    <Button Name="PART_Next_InSpace" Grid.Row="1" Grid.Column="2"
        Style="{TemplateBinding NextButtonInSpaceStyle}"
        Visibility="{TemplateBinding ShowButtonsInSpace,
            Converter={StaticResource booleanToVisibilityConverter}}"/>

    <!-- 2nd row. Caption and small buttons previous/next.-->
    <Grid Name="grdCaption" Grid.Row="0" Grid.ColumnSpan="3">

        <Grid.ColumnDefinitions >
            <ColumnDefinition Width="*"/>
            <ColumnDefinition Width="Auto" />
        </Grid.ColumnDefinitions>

        <Border Style="{TemplateBinding CaptionBorderStyle}"

```



```
        Visibility="{TemplateBinding ShowCaption,
Converter={StaticResource booleanToVisibilityConverter}}">
        <TextBlock Name="PART_Caption"
            Style="{TemplateBinding CaptionStyle}"
            Text="{TemplateBinding Caption }" />
    </Border>

    <Border Name="borderNavigationArea"
        Style="{TemplateBinding NavigationAreaBorderStyle }"
        Grid.Column="1">
        <StackPanel Name="stkSmallNavigationButtons"
            Orientation="Horizontal" >

            <Button Name="PART_Previous_Small"
                Style="{TemplateBinding PreviousButtonSmallStyle}"
                Visibility="{TemplateBinding ShowButtonsSmall,
Converter={StaticResource booleanToVisibilityConverter}}"/>

            <TextBlock Name="PART_NavigationText"
                Text="{TemplateBinding NavigationText}"
                Style="{TemplateBinding NavigationTextStyle}"
                Visibility="{TemplateBinding ShowNavigationText,
Converter={StaticResource booleanToVisibilityConverter}}"/>

            <Button Name="PART_Next_Small"
                Style="{TemplateBinding NextButtonSmallStyle}"
                Visibility="{TemplateBinding ShowButtonsSmall,
Converter={StaticResource booleanToVisibilityConverter}}"/>

        </StackPanel>
    </Border>

</Grid>

</Grid>

</Border>

</ControlTemplate>

</Setter.Value>

</Setter>

</Style>
```

## Παράρτημα - Κλάσεις

Το SlideShow χρησιμοποιεί την παρακάτω **κλάση** για τη λειτουργία του:

Όνομα	Περιγραφή
<a href="#">ImageInfo</a>	Περιλαμβάνει πληροφορίες για την εικόνα (απόλυτο path, caption, εφέ).

## ImageInfo

Περιλαμβάνει πληροφορίες για την εικόνα (απόλυτο path, caption, εφέ).

**Σύνταξη:**

**VB:**

```
Public Class ImageInfo
```

**Μέλη:**

Ιδιότητα	Τύπος	Περιγραφή
Path	String	Το απόλυτο path του αρχείου εικόνας.
Caption	String	Η λεζάντα της εικόνας.
SlidingEffect	SlideShow.SlidingEffectEnum	Το εφέ εισόδου.

**Μέθοδος κατασκευής:**

Ιδιότητα
<pre>New(path As String, caption As String)</pre>
<pre>New(path As String, caption As String, slidingEffect As SlideShow.SlidingEffectEnum)</pre>

# Τέλος Manual