Haute École Léonard de Vinci
SCIENCES ET TECHNIQUES

BINV314A
.NET Outils et Concepts d'Application d'Entreprise

Semaine 8

WPF

Introduction

WPF

- **W**indows **P**resentation **F**oundation
- Création d'application client riche
  - Application graphique lourde (bureau)
  - Application client-serveur
- Composants IHM enrichis
  - Images vectorielles
  - Support Direct3D
  - Animations
- Technologie vieillissante
  - Remplacement par .NET MAUI en cours ...

Outils

- Visual Studio 2022
- Blend
  - sketchflow

XAML

- E**x**tensible **A**pplication **M**arkup **L**anguage
  - Basé sur XML (schéma XML défini par Microsoft)
  - Arbre de représentation des composants
- Séparer le code UI du code de traitement
  - MainWindow.xaml
  - MainWindow.xaml.cs (Code Behind)
- Exemple :

```
<StackPanel>
  <ListBox>
    <ListBoxItem Content="One" />
    <ListBoxItem Content="Two" IsSelected="True" />
    <ListBoxItem Content="Three" />
  </ListBox>
</StackPanel>
```

XAML

- Objectifs XAML
  - Décrire en XML une interface graphique
  - Binding -> MVVM
  - Langage universel (UWP, Windows Phone, WPF, …)

XAML - Events

// **XAML File**

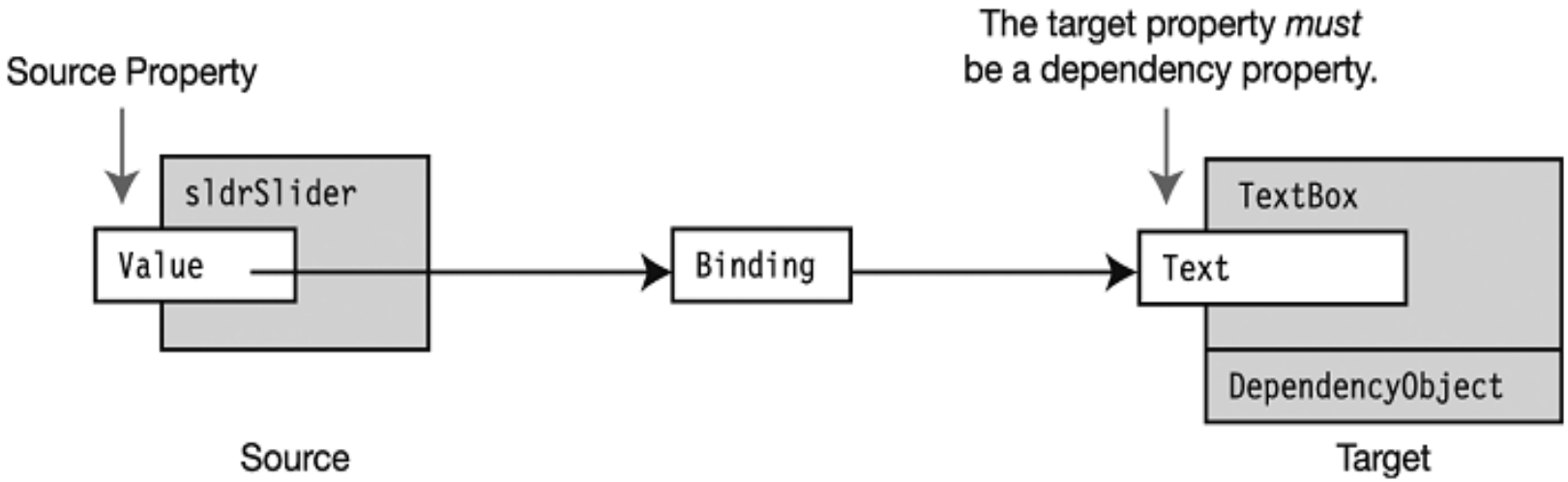
```
<Button Click=”Click_Event”>Hello World</Button>
```

// **C# Code-behind**

```
private void Click_Event(object sender,
RoutedEventArgs e)
{
  Console.WriteLine(“Hello World”);
}
```

XAML - Binding

- Lier facilement des composants (graphiques) ensemble



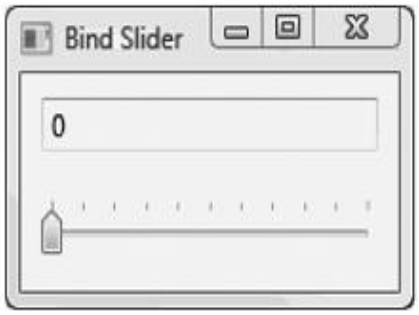
Source : Illustrated WPF – Daniel M. Solis – Apress 2009

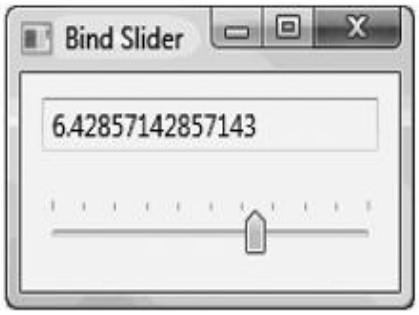
XAML - Binding

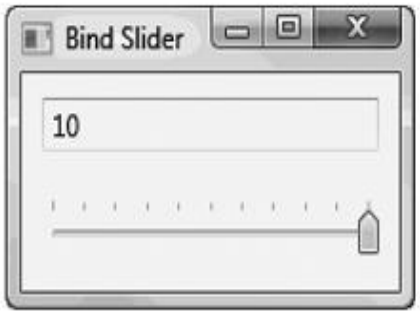
- Binding dans le fichier XAML

  <TextBox Text = {Binding ElementName=slider Path=Value}>

  -> Liaison de la propriété Text de la Textbox avec la propriété Value du slider







Source : Illustrated WPF – Daniel M. Solis – Apress 2009

XAML - Binding

- Binding Direction
  - OneWay, TwoWay, OneWayToSource, OneTime, Default

Mode	Direction of Update	Update When
`OneWay`	S ⟶ T	Immediate
`TwoWay`	S ⟶ T S ⟵ T	Immediate Depends on Value of `UpdateSourceTrigger`
`OneWayToSource`	S ⟵ T	Depends on Value of `UpdateSourceTrigger`

S == Source / T == Target

Source : Illustrated WPF – Daniel M. Solis – Apress 2009

XAML - Binding

For example, if you set the value of `UpdateSourceTrigger` to `PropertyChanged`, as in the following markup, the slider will move immediately when you change the text in the `TextBox`, as long as the text is a valid number:

```
<StackPanel>
   <TextBox Margin="10" Text="{Binding ElementName=sldrSlider, Path=Value,
                    UpdateSourceTrigger=PropertyChanged}" />          ← Set the trigger.
   <Slider Name="sldrSlider" TickPlacement="TopLeft"  Margin="10"/>
</StackPanel>
```

Source : Illustrated WPF – Daniel M. Solis – Apress 2009

XAML - Ressources

- Ressource statique

```
<StackPanel>
   <StackPanel.Resources>
      <SolidColorBrush    x:Key="background" Color="Silver"/>
   </StackPanel.Resources>

   <Button Background="{StaticResource background}">Button 1</Button>
</StackPanel>
```

Key Attribute

Property Element Syntax

Specify the Key.

Markup Extension Class

Key

Source : Illustrated WPF – Daniel M. Solis – Apress 2009

XAML - Ressources

- Ressource dynamique
  - Déclaration
    - … x:Key="gradBrush" …
  - Utilisation(s)
    - Background="{DynamicResource gradBrush}"
  - Modification(s)
    - this.Resources["gradBrush"] = Brushes.Silver;

Source : Illustrated WPF – Daniel M. Solis – Apress 2009

XAML - Styles

- Objectif : appliquer un ensemble de propriétés à un composant
  - Déclaration style

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

  - Utilisation style

```
<TextBox style={StaticRessource monstyle} …>
```

XAML - Styles

- Exemple

```
          Key          Style Name Suffix
           ↓              ↓
<Style x:Key="buttonStyle">
   <Setter Property="Button.Height"     Value="40"   />
   <Setter Property="Button.Width"      Value="110"  />
   <Setter Property="Button.FontSize"   Value="16"   />
   <Setter Property="Button.FontWeight" Value="Bold" />
</Style>
```

Property Attribute — **Setters for a** Named Style Must Include a Class Name — **Value** Attribute

```
<Button Style="{StaticResource buttonStyle}">Button 1</Button>
                              ↑
```

Retrieve the style from the Resources collection.

Source : Illustrated WPF – Daniel M. Solis – Apress 2009

XAML – Data templates

```
                    Declare the control template.
<Window.Resources>  ↓
   <DataTemplate   x:Key="NiceFormat">
      <Border Margin="1" BorderBrush="Blue"
              BorderThickness="2" CornerRadius="2">
         <Grid>
            <Grid.RowDefinitions>
               <RowDefinition/><RowDefinition/>
            </Grid.RowDefinitions>
            <Grid.ColumnDefinitions>
               <ColumnDefinition Width="60"/><ColumnDefinition Width="20"/>
            </Grid.ColumnDefinitions>
            <TextBlock FontWeight="Bold" Grid.Row="0" Grid.Column="0"
                       Text="{Binding FirstName}" Padding="2"/>
                                  ↑
                       Bind to a field in the DataContext.
            <Rectangle Grid.Row="0" Grid.Column="1" Grid.RowSpan="2"
                       Fill="{Binding FavoriteColor}"/>
                                  ↑
                       Bind to a field in the DataContext.
            <TextBlock Padding="2" Grid.Row="1" Grid.Column="0"
                       Text="{Binding Age}"/>
                                  ↑
                    Bind to a field in the DataContext.
         </Grid>
      </Border>
   </DataTemplate>
</Window.Resources>
```

```
<StackPanel Orientation="Horizontal">
   <ListBox Name="listPeople" SelectedIndex="0" VerticalAlignment="Top"
            ItemTemplate="{StaticResource NiceFormat}"/>
                                  ↑
                    Apply the data template.
   <StackPanel Orientation="Vertical" Name="sp" Margin="10, 5"
               DataContext="{Binding ElementName=listPeople, Path=SelectedItem}">
      <Label Name="lblFName" FontWeight="Bold" FontSize="16"/>
      <Label Name="lblAge"/>
      <Label Name="lblColor"/>
   </StackPanel>
</StackPanel>
```

Source : Illustrated WPF – Daniel M. Solis – Apress 2009

XAML - Datacontext

- Lier directement une classe à un composant (graphique)
- Exemple :

```
// fichier EtudiantsData.cs
public class Etudiant {
        private string _nom;
      public Nom {
        get { return _nom;}
    }
}
```

XAML - Datacontext

- Exemple :

```
// fichier EtudiantsData.cs
public class EtudiantsData
{
      private IList<Etudiant> etudList;

        public ListEtud {
                get { return etudList;}
      }
}
```

XAML - Datacontext

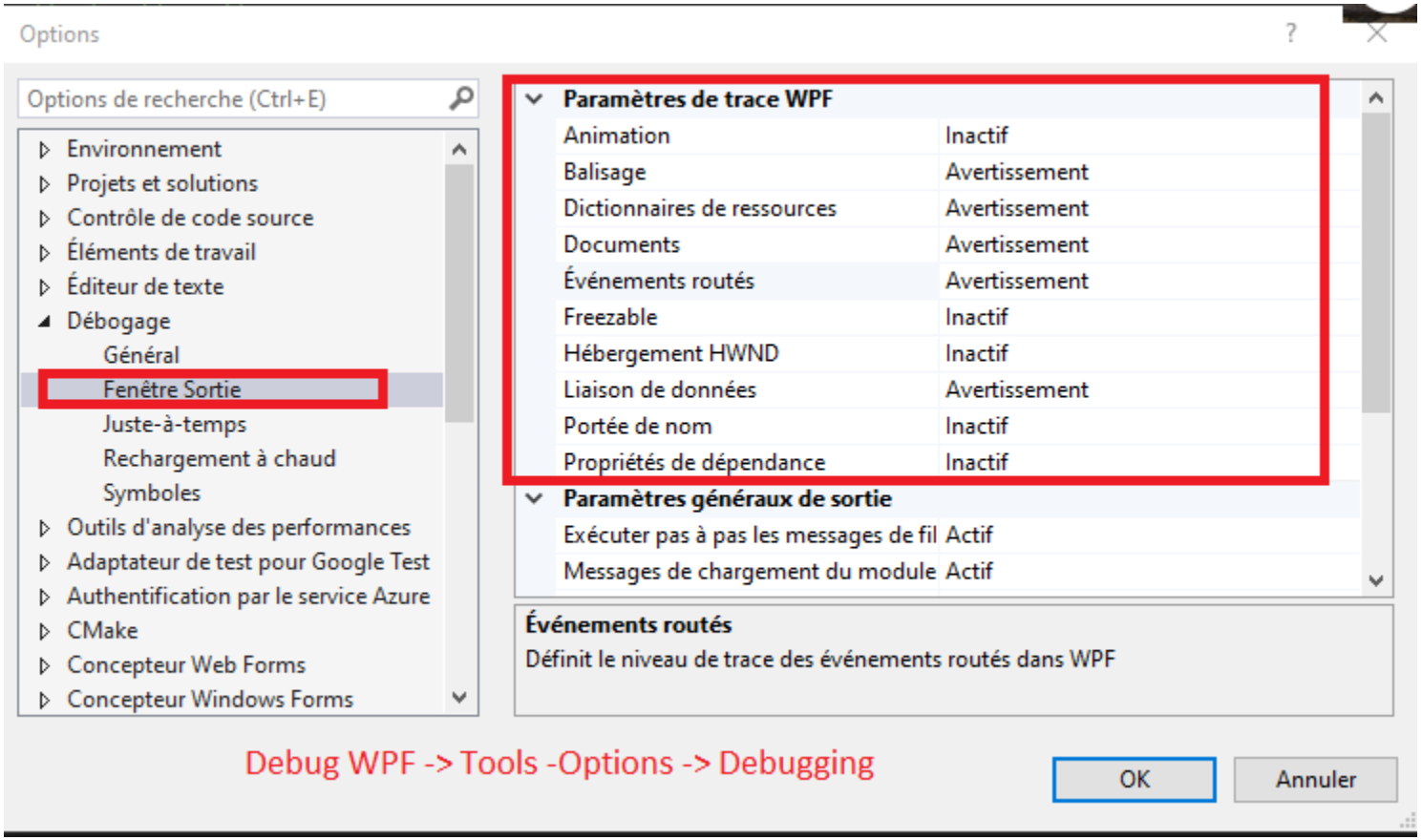
- Exemple :

```
<ComboBox x:Name=«comboEtud»  ..... ItemsSource="{Binding
ListEtud}/>

….

<Label x:Name="nomEtud" Content="{Binding ElementName=comboEtud
,Path=SelectedItem.Nom}" />
```

Debugging WPF



Debugging WPF

- Console.Writeline -> fenêtre de sortie
- MessageBox.Show(message) -> popup