Lecture 4 – Tool Box

Mr. Yousif Garabet Arshak
Computer Science Department
University of Zakho
yousif.arshak@uoz.edu.krd

Outlines

- Xamarin.Forms Tool Box (Controls)
  1. Label
  2. Button
  3. Entry
  4. Editor
  5. Switch
  6. SearchBar
  7. BoxView

Xamarin.Forms Tool Box (Controls)

- Xamarin.Forms empowers you to deliver the same experience to multiple platforms. Using a set of UI elements abstracted from common mobile app controls, like text inputs and buttons, you can quickly create a functional cross-platform UI. Xamarin.Forms compiles these generic controls down to the platform-specific version, maintaining the native look and feel of your app. Likewise, you can easily customize the controls to a specific look and feel for your app. Today, we are excited to debut the Xamarin.Forms Controls Toolbox.
- The toolbox lists the available Xamarin.Forms controls. Furthermore, these can be dragged directly onto the XAML editing surface to create the control on your page!

Label

- The Label view is used for displaying text, both single and multi-line. Labels can have text decorations, colored text, and use custom fonts (families, sizes, and options).

Text decorations

Underline and strikethrough text decorations can be applied to Label instances by setting the Label.TextDecorations property to one or more TextDecorations enumeration members:

- None
- Underline
- Strikethrough

<Label Text="This is underlined text." TextDecorations="Underline" />

<Label Text="This is text with strikethrough." TextDecorations="Strikethrough" />

<Label Text="This is underlined text with strikethrough." TextDecorations="Underline, Strikethrough" />

To read in C# → var text = MyLabel.Text;

Button

- The Button responds to a tap or click that directs an application to carry out a particular task.
- The Button is the most fundamental interactive control in all of Xamarin.Forms. The Button usually displays a short text string indicating a command, but it can also display a bitmap image, or a combination of text and an image. The user presses the Button with a finger or clicks it with a mouse to initiate that command.

XAML → <Button Text="I am a Button“ Clicked="OnButtonClicked" />

C# → var MyEntry = new Entry { Text = "I am an Entry" };

To read in C# →

```
async void OnButtonClicked(object sender, EventArgs args)
{await DisplayAlert ("Alert", "You have been alerted", "OK");}
```

Entry

- The Xamarin.Forms Entry is used for single-line text input. The Entry, like the Editor view, supports multiple keyboard types. Additionally, the Entry can be used as a password field.
- Set and read text
  - The Entry, like other text-presenting views, exposes the Text property. This property can be used to set and read the text presented by the Entry. The following example demonstrates setting the Text property in XAML:

XAML	→	<Entry Text="I am an Entry“ x:Name="myEntry" />
C#	→	var MyEntry = new Entry { Text = "I am an Entry" };
To read in C# →		var text = MyEntry.Text;

Editor

- The Editor control is used to accept multi-line input.
- **Set and read text**
- The Editor, like other text-presenting views, exposes the Text property. This property can be used to set and read the text presented by the Editor. The following example demonstrates setting the Text property in XAML:

XAML →	<Editor Text="I am an Editor" />
C# →	var MyEditor = new Editor { Text = "I am an Editor" };
To read in C# →	var text = MyEditor.Text;

Switch

- The Xamarin.Forms Switch control is a horizontal toggle button that can be manipulated by the user to toggle between on and off states, which are represented by a boolean value. The Switch class inherits from View.
- The Switch control defines the following properties:
  - IsToggled is a boolean value that indicates whether the Switch is on.
  - OnColor is a Color that affects how the Switch is rendered in the toggled, or on, state.
  - ThumbColor is the Color of the switch thumb.

SearchBar

- The Xamarin.Forms SearchBar is a user input control used to initiating a search. The SearchBar control supports placeholder text, query input, search execution, and cancellation.

XAML → <SearchBar Placeholder="Search items..." />

BoxView

- BoxView renders a simple rectangle of a specified width, height, and color. You can use BoxView for decoration, rudimentary graphics, and for interaction with the user through touch.
- Because Xamarin.Forms does not have a built-in vector graphics system, the BoxView helps to compensate. Some of the sample programs described in this article use BoxView for rendering graphics. The BoxView can be sized to resemble a line of a specific width and thickness, and then rotated by any angle using the Rotation property.

Setting BoxView Color and Size

- Typically you'll set the following properties of BoxView:
  - Color to set its color.
  - CornerRadius to set its corner radius.
  - WidthRequest to set the width of the BoxView in device-independent units.
  - HeightRequest to set the height of the BoxView.

```
<ContentPage
xmlns="http://xamarin.com/schemas/2014/forms"

xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
        xmlns:local="clr-namespace:BasicBoxView"
        x:Class="BasicBoxView.MainPage">

   <BoxView Color="CornflowerBlue"
        CornerRadius="10"
        WidthRequest="160"
        HeightRequest="160"
        VerticalOptions="Center"
        HorizontalOptions="Center" />

</ContentPage>
```

Assignment: What we do with these Controls with examples

Each group should make one of these assignments and send back to me

1. Label
2. Button
3. Entry
4. Editor
5. Switch
6. SearchBar
7. BoxView

You have 7 days to send your assignment after 7 days you will lose one degree day by day

Any Questions?