

Introduction to WPF

XAML

- XAML: eXtensible Application Markup Language
- Basic concepts and features around XAML
- Create a simple WPF (Windows Presentation Foundation) application

An Initial Window

Let's take a look at a WPF Initial Window:

```
<Window x:Class="FirstWPFApplication.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="350" Width="525">
    <Grid>

    </Grid>
</Window>
```

XAML

- XAML is XML

If you need an overview of XML, you can go here:

<http://www.w3schools.com/xml/default.asp>

- XAML Elements are .NET Classes

both “<Window>” and “<Grid>” are .NET Classes.

- XAML Namespaces

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

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

XAML Code Behind

```
<Window x:Class="FirstWPFApplication.MainWindow"
```

`x:Class` attribute references the code behind for this XAML window.

“`x:`” means the class attribute come from the XAML namespace noted above.

The value of the attribute references the MainWindow class in the “FirstWPFApplication” namespace.

```
Title="MainWindow" Height="350" Width="525">
```

Simply properties of the Window class

The Grid Element

For now, note that the Window element only allows for a single child. This means that if we want to include more than one Control in our Window, we will want to wrap them in some sort of layout control that allows multiple children. The Grid is just that sort of control.

Attached Properties

Add a Button, and a Label under the Grid as:

```
<Button Content="Button" HorizontalAlignment="Left" Margin="214,59,0,0"  
VerticalAlignment="Top" Width="75" Click="Button_Click_1"/>
```

```
<Label x:Name="theLabel" Content="" HorizontalAlignment="Left"  
Margin="120,142,0,0" VerticalAlignment="Top" Width="261" Background="#FFB3E0F9" />
```

- Button, Label is nested inside the Grid markup
- Most of the attributes are simply properties of the Label class

Event Handler

- Specifying an Event Handler in XAML is just the same as setting any other Property as an attribute. For our button, we want to hook into the “Click” event.
- Visual Studio IntelliSense helps us out here. Inside the Button tag, type “Click=” and Visual Studio will give you an option to create a “New EventHandler”. If you select this, then Visual Studio will automatically create the event handler in the code behind, give it a name, and hook it up in the XAML.

Comments

- Comments in XAML are marked like XML & HTML comments, with the “ `<!-- -->` ” tag.

WPF Documentation on MSDN

- [https://msdn.microsoft.com/en-us/library/ms754130\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/ms754130(v=vs.110).aspx)