

# Introduction to XAML

## Don't Fear the Markup

**Jeremy Clark**  
Developer Educator  
[github.com/jeremybytes](https://github.com/jeremybytes)  
Level: Introductory



# Session Survey

- Your feedback is very important to us
- Please take a moment to complete the session survey found in the mobile app
- Use the QR code or search for “Converge360 Events” in your app store
- Find this session on the Agenda tab
- Click “Session Evaluation”
- Thank you!





# Introduction to XAML

## Don't Fear the Markup

**Jeremy Clark**  
Developer Educator  
[github.com/jeremybytes](https://github.com/jeremybytes)  
Level: Introductory



# What is XAML?

```
<Button x:Name="ClearButton"  
        Content="Clear"  
        Click="ClearButton_Click"/>
```

- XAML = eXtensible Application Markup Language
- Classes/objects as markup
- Gets compiled to IL (just like C#)
- Various Environments
  - WPF
  - WinUI
  - MAUI
  - Uno Platform / Avalonia UI

# XAML Markup

- Namespaces
- Classes / Code Behind
- Properties as Attributes
- Properties as Nested Elements
- Attached Properties
- Comments
- TypeConverter-Enabled Properties
- Content
- Control Styles / Templates
- Data Binding Syntax

# Namespaces

- Built-in Namespaces

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

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

- Custom Namespaces

```
xmlns:local="clr-namespace:WPFSStopWatch"
```



# Classes / Code Behind

- Class in XAML

```
<Window x:Class="WPFS stopwatch.MainWindow" ...>  
</Window>
```

- Class in Code

```
public partial class MainWindow : Window {...}
```

# Properties as Attributes

- Window Properties

```
<Window x:Class="WPFStopWatch.MainWindow"  
        Title="WPF Stop Watch"  
        Height="150" Width="300"  
        Topmost="True" WindowStyle="None">
```

... More elements here

```
</Window>
```



# Properties as Nested Elements

- Grid.RowDefinitions

```
<Grid>
```

```
  <Grid.RowDefinitions>
```

```
    <RowDefinition Height="Auto" />
```

```
    <RowDefinition Height="Auto" />
```

```
  </Grid.RowDefinitions>
```

```
  ... More elements here
```

```
</Grid>
```

# Attached Properties

- Grid.Row

```
<StackPanel Grid.Row="1" Orientation="Horizontal"  
            HorizontalAlignment="Center">
```

... More elements here

```
</StackPanel>
```

- Property belongs to the Grid but is attached to the StackPanel

# Comments

- Just Like HTML / XML Comments

<!-- Output Display -->

```
<TextBlock Grid.Row="0"  
    FontSize="64"  
    HorizontalAlignment="Center"  
    VerticalAlignment="Center"  
    Text="00:00"/>
```

# TypeConverter-Enabled Properties

- Margin / Padding – Type Thickness

```
<Button x:Name="ClearButton"  
        Margin="6" Padding="10,3"  
        Content="Clear"  
        Click="ClearButton_Click"/>
```

Thickness	
Left	= 10
Top	= 3
Right	= 10
Bottom	= 3

- Background / Foreground – Type Brush

```
Background="#FF333333" Foreground="White"
```



# Content 3 Ways

- Nested Element

```
<Button x:Name="StartButton">  
    <Button.Content>Start</Button.Content>  
</Button>
```

- Default Content

```
<Button x:Name="StopButton">Stop</Button>
```

- Attribute

```
<Button x:Name="ClearButton" Content="Clear" />
```

# Control Styles

```
<Style TargetType="Button" x:Key="AppButtonStyle">  
    <Setter Property="FontSize" Value="28" />  
    <Setter Property="Margin" Value="6" />  
    <Setter Property="Padding" Value="10,3" />  
</Style>
```

- Target Type – required
- x:Key – optional
  - With x:Key, the style must be specified on each control
  - Without x:Key, the style applies to all controls of that type

# Using a Static Resource

- Specifying a Control Style

```
<Button x:Name="ClearButton"  
        Content="Clear"  
        Style="{StaticResource AppButtonStyle}"  
/>
```

# Adding a Custom Class to Markup

- Class as a resource

```
<Window.Resources>  
    <local:Ticker x:Key="LocalTicker" />  
</Window.Resources>
```

- The class must have a no-parameter constructor



# Data Binding

```
<TextBlock Grid.Row="0" FontSize="64"  
    HorizontalAlignment="Center"  
    VerticalAlignment="Center"  
    DataContext="{StaticResource LocalTicker}"  
    Text="{Binding DisplayInterval}"/>
```

- DataContext Property: where the data comes from
- Binding Markup Extension: where the property value comes from
  - Note: Data Binding is extremely flexible.

# XAML Markup

- Namespaces
- Classes / Code Behind
- Properties as Attributes
- Properties as Nested Elements
- Attached Properties
- Comments
- TypeConverter-Enabled Properties
- Content
- Control Styles / Templates
- Data Binding Syntax

# Session Survey

- Your feedback is very important to us
- Please take a moment to complete the session survey found in the mobile app
- Use the QR code or search for “Converge360 Events” in your app store
- Find this session on the Agenda tab
- Click “Session Evaluation”
- Thank you!





Thank You!

Jeremy Clark

- [jeremybytes.com](http://jeremybytes.com)
- [github.com/jeremybytes](https://github.com/jeremybytes)

<https://github.com/jeremybytes/vslive2025-sandiego>