Layout

Introduction

- We use the layout controls for containing other controls. The layout controls position the other controls and set their size. Each layout control function differently.

The `Canvas` Layout Control

- Using the `Canvas` control we can specify the distances between the contained controls' left, top, right, and bottom edges and those of the `Canvas`.

The `Canvas` Layout Control

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        x:Class="gaga.MainWindow" Title="Simple Demo"
        Width="400" Height="140" FontSize="14">

    <Canvas>
        <Button Content="Button Left" Width="105" Height="40"
                Canvas.Top="20" Canvas.Left="20"/>
        <Button Content="Button Right" Width="105" Height="40"
                Canvas.Top="20" Canvas.Right="20"/>
    </Canvas>
</Window>
```

The `Canvas` Layout Control

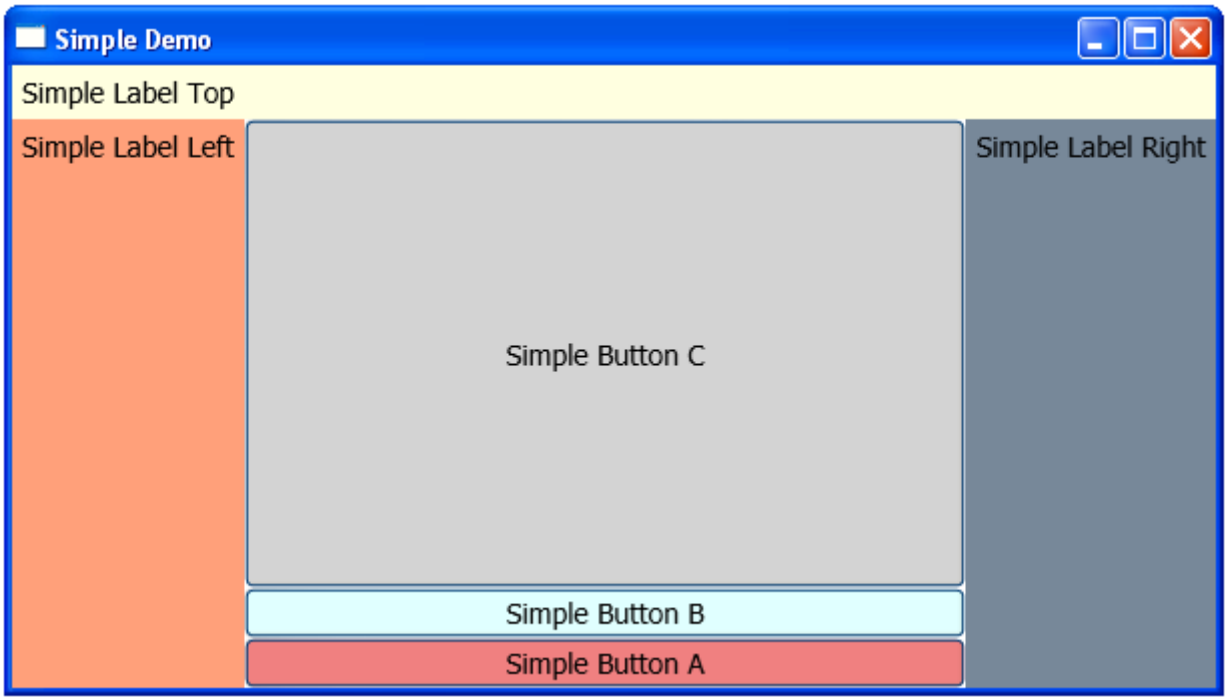
The `DockPanel` Layout Control

- When using the `DockPanel` layout control, the contained controls are docked to its edges.

The `DockPanel` Layout Control

```
<Window1     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             x:Class="gaga.MainWindow" Title="Simple Demo"
             Width="800" Height="600" FontSize="14">

    <DockPanel>
        <Label Background="LightYellow" Content="Simple Label Top"
            DockPanel.Dock="Top"/>
        <Label Background="LightSalmon" Content="Simple Label Left"
            DockPanel.Dock="Left" />
        <Label Background="LightSlateGray" Content="Simple Label Right"
            DockPanel.Dock="Right" />
        <Button DockPanel.Dock="Bottom" Background="LightCoral"
            Content="Simple Button A"/>
        <Button DockPanel.Dock="Bottom" Background="LightCyan"
            Content="Simple Button B"/>
        <Button DockPanel.Dock="Bottom" Background="LightGray"
            Content="Simple Button C"/>
    </DockPanel>

</Window>
```

The `DockPanel` Layout Control



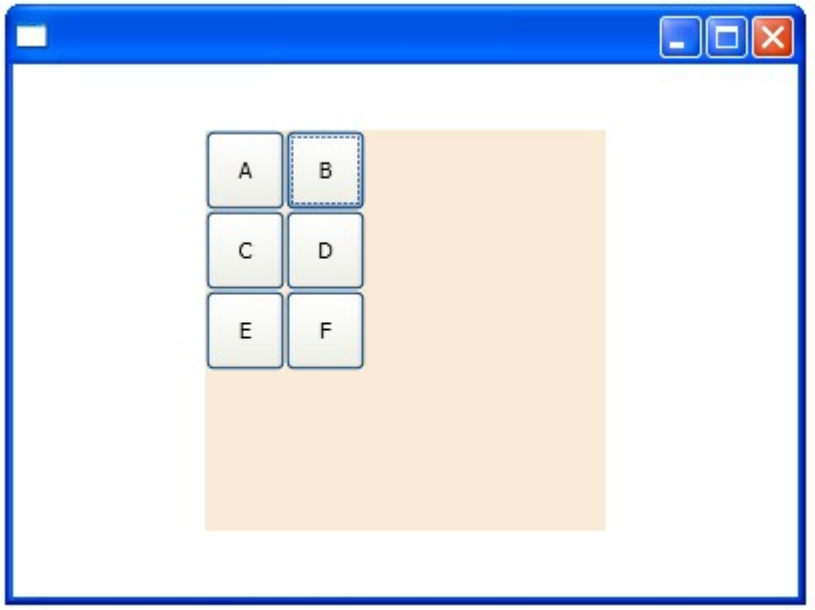
The `Grid` Layout Controller

- This layout controller lays out it child controllers into rows and columns.

The `Grid` Layout Controller

```
<Window      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             x:Class="gaga.MainWindow" Width="400" Height="300">

    <Grid Margin="2" Width="200" Height="200" Background="AntiqueWhite">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="40"/>
            <ColumnDefinition Width="40"/>
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="40"/>
            <RowDefinition Height="40"/>
            <RowDefinition Height="40"/>
        </Grid.RowDefinitions>
        <Button Grid.Row="0" Grid.Column="0" Content="A"/>
        <Button Grid.Row="0" Grid.Column="1" Content="B"/>
        <Button Grid.Row="1" Grid.Column="0" Content="C"/>
        <Button Grid.Row="1" Grid.Column="1" Content="D"/>
        <Button Grid.Row="2" Grid.Column="0" Content="E"/>
        <Button Grid.Row="2" Grid.Column="1" Content="F"/>
    </Grid>

</Window>
```

The `Grid` Layout Controller



The `Grid` Layout Controller

- The `ColumnDefinitions` property holds a collection of `ColumnDefinition` objects that determine the widths of the columns.
- The `RowDefinitions` property is a collection of `RowDefinition` objects that determine the heights of the Grid's rows.

The `Grid` Layout Controller

- The `RowDefinition` and the `ColumnDefinition` objects have properties that determine their sizes. Width for columns. Height for rows.
- We can assign `Width` and `Height` either with absolute values or with proportional ones.
- In order to assign absolute values we should use simple values. We can use a unit indicator to indicate that we want to use a unit other than pixels.

The `Grid` Layout Controller

- The possible unit indicators are `px` (pixels), `in` (inches), `cm` (centimeters) and `pt` (points).
- In order to use a proportional width or height we should assign a number following with an asterisk (`*`).

The `Grid` Layout Controller

```
<Grid Margin=”5”>
    <Grid.ColumnDefinitions>
            <ColumnDefinition Width=”2.75 cm”/>
            <ColumnDefinition Width=”2*”/>
            <ColumnDefinition Width=”1*”/>
    </Grid.ColumnDefinitions>
    <Grid.RowDefinitions>
            <RowDefinition Height=”30 in”/>
            <RowDefinition Height=”*”/>
    </Grid.RowDefinitions>
</Grid>
```

The `Grid` Layout Controller

- We can set proportional values that add up either to 1 or to 100. That will set the sizes as percentages of the remaining space.

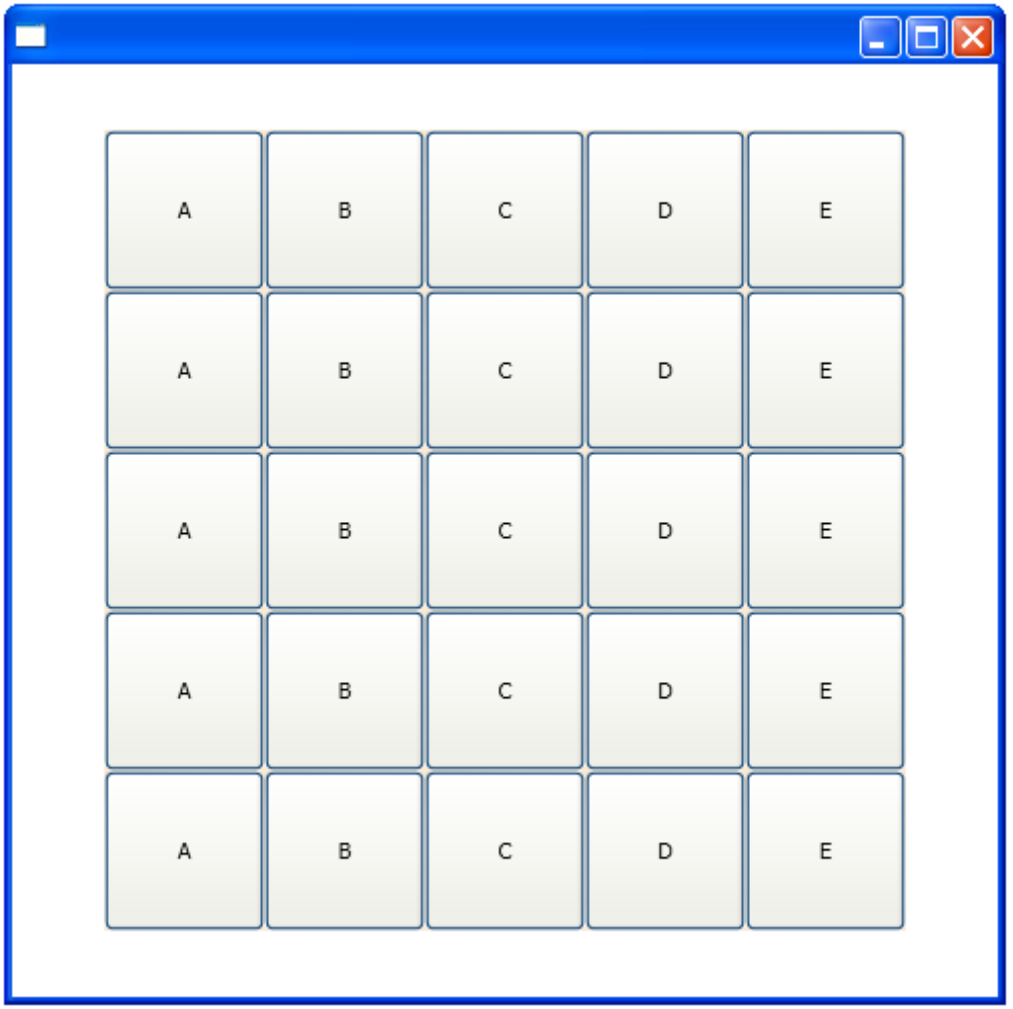
The `Grid` Layout Controller

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    x:Class="gaga.MainWindow" Width="500" Height="500">

    <Grid Margin="2" Width="400" Height="400" Background="AntiqueWhite">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="20*"/>
            <ColumnDefinition Width="20*"/>
            <ColumnDefinition Width="20*"/>
            <ColumnDefinition Width="20*"/>
            <ColumnDefinition Width="20*"/>
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="20*"/>
            <RowDefinition Height="20*"/>
            <RowDefinition Height="20*"/>
            <RowDefinition Height="20*"/>
            <RowDefinition Height="20*"/>
        </Grid.RowDefinitions>
        <Button Grid.Row="0" Grid.Column="0" Content="A"/>
        <Button Grid.Row="0" Grid.Column="1" Content="B"/>
        <Button Grid.Row="0" Grid.Column="2" Content="C"/>
        <Button Grid.Row="0" Grid.Column="3" Content="D"/>
        <Button Grid.Row="0" Grid.Column="4" Content="E"/>
```

The `Grid` Layout Controller

```
        <Button Grid.Row="1" Grid.Column="0" Content="A"/>
        <Button Grid.Row="1" Grid.Column="1" Content="B"/>
        <Button Grid.Row="1" Grid.Column="2" Content="C"/>
        <Button Grid.Row="1" Grid.Column="3" Content="D"/>
        <Button Grid.Row="1" Grid.Column="4" Content="E"/>
        <Button Grid.Row="2" Grid.Column="0" Content="A"/>
        <Button Grid.Row="2" Grid.Column="1" Content="B"/>
        <Button Grid.Row="2" Grid.Column="2" Content="C"/>
        <Button Grid.Row="2" Grid.Column="3" Content="D"/>
        <Button Grid.Row="2" Grid.Column="4" Content="E"/>
        <Button Grid.Row="3" Grid.Column="0" Content="A"/>
        <Button Grid.Row="3" Grid.Column="1" Content="B"/>
        <Button Grid.Row="3" Grid.Column="2" Content="C"/>
        <Button Grid.Row="3" Grid.Column="3" Content="D"/>
        <Button Grid.Row="3" Grid.Column="4" Content="E"/>
        <Button Grid.Row="4" Grid.Column="0" Content="A"/>
        <Button Grid.Row="4" Grid.Column="1" Content="B"/>
        <Button Grid.Row="4" Grid.Column="2" Content="C"/>
        <Button Grid.Row="4" Grid.Column="3" Content="D"/>
        <Button Grid.Row="4" Grid.Column="4" Content="E"/>
    </Grid>
</Window>
```

The `Grid` Layout Controller



The `Grid` Layout Controller

- Using the `RowSpan` and the `ColumnSpan` properties we can have the child covering more than one row or one column.
- Using `RowSpan` the child will span over two rows. Using `ColumnSpan` the child will span over two columns.

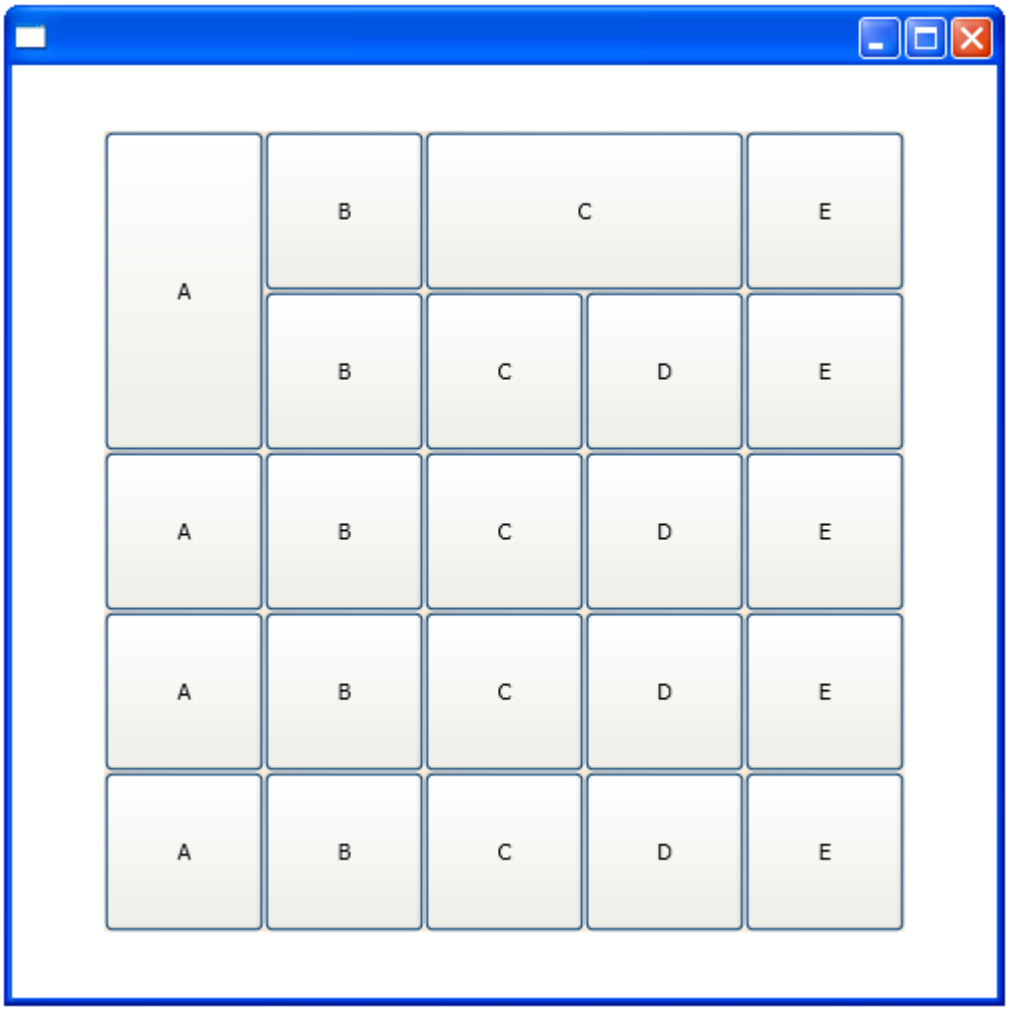
The `Grid` Layout Controller

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    x:Class="gaga.MainWindow" Width="500" Height="500">

    <Grid Margin="2" Width="400" Height="400" Background="AntiqueWhite">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="20*"/>
            <ColumnDefinition Width="20*"/>
            <ColumnDefinition Width="20*"/>
            <ColumnDefinition Width="20*"/>
            <ColumnDefinition Width="20*"/>
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="20*"/>
            <RowDefinition Height="20*"/>
            <RowDefinition Height="20*"/>
            <RowDefinition Height="20*"/>
            <RowDefinition Height="20*"/>
        </Grid.RowDefinitions>
        <Button Grid.RowSpan="2"  Grid.Row="0" Grid.Column="0" Content="A"/>
        <Button Grid.Row="0" Grid.Column="1" Content="B"/>
        <Button Grid.Row="0" Grid.Column="2" Grid.ColumnSpan="2" Content="C"/>
        <Button Grid.Row="0" Grid.Column="4" Content="E"/>
```

The `Grid` Layout Controller

```
        <Button Grid.Row="1" Grid.Column="1" Content="B"/>
        <Button Grid.Row="1" Grid.Column="2" Content="C"/>
        <Button Grid.Row="1" Grid.Column="3" Content="D"/>
        <Button Grid.Row="1" Grid.Column="4" Content="E"/>
        <Button Grid.Row="2" Grid.Column="0" Content="A"/>
        <Button Grid.Row="2" Grid.Column="1" Content="B"/>
        <Button Grid.Row="2" Grid.Column="2" Content="C"/>
        <Button Grid.Row="2" Grid.Column="3" Content="D"/>
        <Button Grid.Row="2" Grid.Column="4" Content="E"/>
        <Button Grid.Row="3" Grid.Column="0" Content="A"/>
        <Button Grid.Row="3" Grid.Column="1" Content="B"/>
        <Button Grid.Row="3" Grid.Column="2" Content="C"/>
        <Button Grid.Row="3" Grid.Column="3" Content="D"/>
        <Button Grid.Row="3" Grid.Column="4" Content="E"/>
        <Button Grid.Row="4" Grid.Column="0" Content="A"/>
        <Button Grid.Row="4" Grid.Column="1" Content="B"/>
        <Button Grid.Row="4" Grid.Column="2" Content="C"/>
        <Button Grid.Row="4" Grid.Column="3" Content="D"/>
        <Button Grid.Row="4" Grid.Column="4" Content="E"/>
    </Grid>
</Window>
```

The `Grid` Layout Controller



The `ScrollViewer` Layout Controller

- This layout control can hold a single child within a scrollable region.
- The `HorizontalScrollBarVisibility` and the `VerticalScrollBarVisibility` properties can take the following values: `Auto`, `Visible`, `Disabled` and `Hidden`. When using the `Disabled` value the child will be re-sized accordingly.

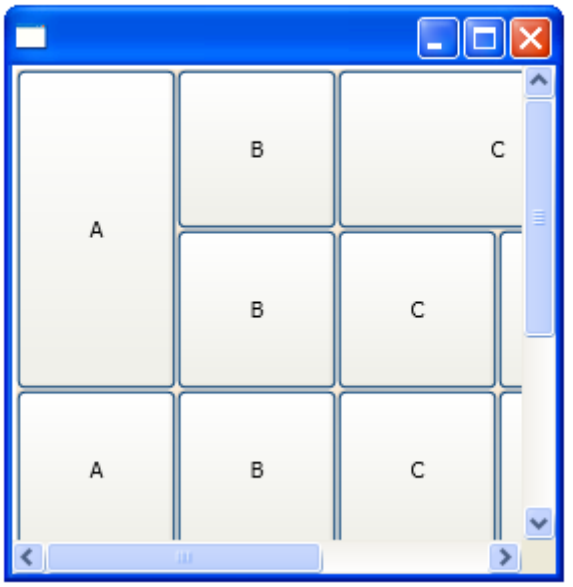
The `ScrollViewer` Layout Controller

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        x:Class="gaga.MainWindow" Width="200" Height="200">

    <ScrollViewer HorizontalScrollBarVisibility="Auto"
                  VerticalScrollBarVisibility="Auto" >

    <Grid Margin="2" Width="400" Height="400" Background="AntiqueWhite">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="20*"/>
            <ColumnDefinition Width="20*"/>
            <ColumnDefinition Width="20*"/>
            <ColumnDefinition Width="20*"/>
            <ColumnDefinition Width="20*"/>
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="20*"/>
            <RowDefinition Height="20*"/>
            <RowDefinition Height="20*"/>
            <RowDefinition Height="20*"/>
            <RowDefinition Height="20*"/>
        </Grid.RowDefinitions>
        <Button Grid.RowSpan="2"  Grid.Row="0" Grid.Column="0" Content="A"/>
        <Button Grid.Row="0" Grid.Column="1" Content="B"/>
```

The `ScrollViewer` Layout Controller

```
        <Button Grid.Row="0" Grid.Column="2" Grid.ColumnSpan="2" Content="C"/>
        <Button Grid.Row="0" Grid.Column="4" Content="E"/>
        <Button Grid.Row="1" Grid.Column="1" Content="B"/>
        <Button Grid.Row="1" Grid.Column="2" Content="C"/>
        <Button Grid.Row="1" Grid.Column="3" Content="D"/>
        <Button Grid.Row="1" Grid.Column="4" Content="E"/>
        <Button Grid.Row="2" Grid.Column="0" Content="A"/>
        <Button Grid.Row="2" Grid.Column="1" Content="B"/>
        <Button Grid.Row="2" Grid.Column="2" Content="C"/>
        <Button Grid.Row="2" Grid.Column="3" Content="D"/>
        <Button Grid.Row="2" Grid.Column="4" Content="E"/>
        <Button Grid.Row="3" Grid.Column="0" Content="A"/>
        <Button Grid.Row="3" Grid.Column="1" Content="B"/>
        <Button Grid.Row="3" Grid.Column="2" Content="C"/>
        <Button Grid.Row="3" Grid.Column="3" Content="D"/>
        <Button Grid.Row="3" Grid.Column="4" Content="E"/>
        <Button Grid.Row="4" Grid.Column="0" Content="A"/>
        <Button Grid.Row="4" Grid.Column="1" Content="B"/>
        <Button Grid.Row="4" Grid.Column="2" Content="C"/>
        <Button Grid.Row="4" Grid.Column="3" Content="D"/>
        <Button Grid.Row="4" Grid.Column="4" Content="E"/>
    </Grid>

    </ScrollViewer>

</Window>
```

The `ScrollViewer` Layout Controller



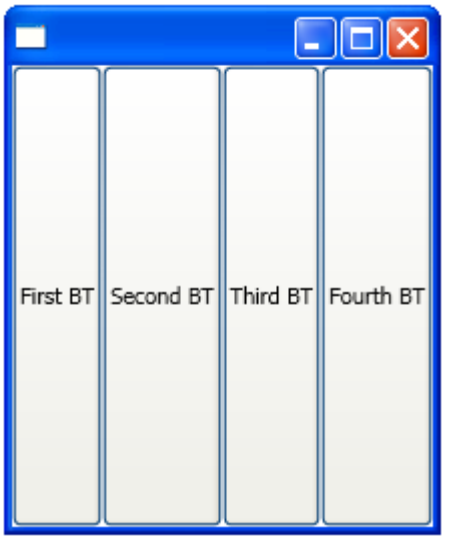
The `StackPanel` Layout Controller

- The `StackPanel` layout control displays its children in a single row or in a single column. If a child doesn't fit then it will be clipped.
- The Orientation property takes one of the following two possible values: `Vertical` and `Horizontal`.

The `StackPanel` Layout Controller

```
<Window     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
            xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
            x:Class="gaga.MainWindow" Width="200" Height="200">

    <StackPanel Orientation="Horizontal">
        <Button Content="First BT"/>
        <Button Content="Second BT"/>
        <Button Content="Third BT"/>
        <Button Content="Fourth BT"/>
    </StackPanel>

</Window>
```

The `StackPanel` Layout Controller



The `TabControl` Layout Control

- The `TabControl` layout control displays a series of tabs the user can select from.
- Each `TabItem` can hold one single child only. That child can be a layout control.

The `TabControl` Layout Control

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        x:Class="gaga.MainWindow" Width="600" Height="480" Icon="pixpix.png">

    <TabControl HorizontalAlignment="Left">

        <TabItem>
            <TabItem.Header>
                Butchart Gardens
            </TabItem.Header>
            <Grid>
                <Image Source="butchart.jpg"/>
            </Grid>
        </TabItem>

        <TabItem>
            <TabItem.Header>
                UBC Library
            </TabItem.Header>
            <Grid>
                <Image Source="vancouver.jpg"/>
            </Grid>
        </TabItem>
```

The `TabControl` Layout Control

```
        <TabItem>
            <TabItem.Header>
                Canadian Rockies
            </TabItem.Header>
            <Grid>
                <Image Source="banf.jpg"/>
            </Grid>
        </TabItem>

    </TabControl>

</Window>
```

The `TabControl` Layout Control

The `ToolBar` Control

- The `ToolBar` control represents the tool bar area where we can place various tools such as buttons and combo boxes.

The `ToolBarTray` Control

- The `ToolBarTray` control is the parent control that can hold one or more `ToolBar` controls.
- The `ToolBarTray` control rearranges the `ToolBar` controls it holds.
- The `Band` and the `BandIndex` are two properties we can use in order to control the exact positioning within the `ToolBarTray` area.

The `ToolBarTray` Control

- The `Band` property holds the numeric value of the band starting with `0` (`0`,`1`...). The `BandIndex` value starting with `0` (`0`,`1`...) determines the position of the `ToolBar` within the band.

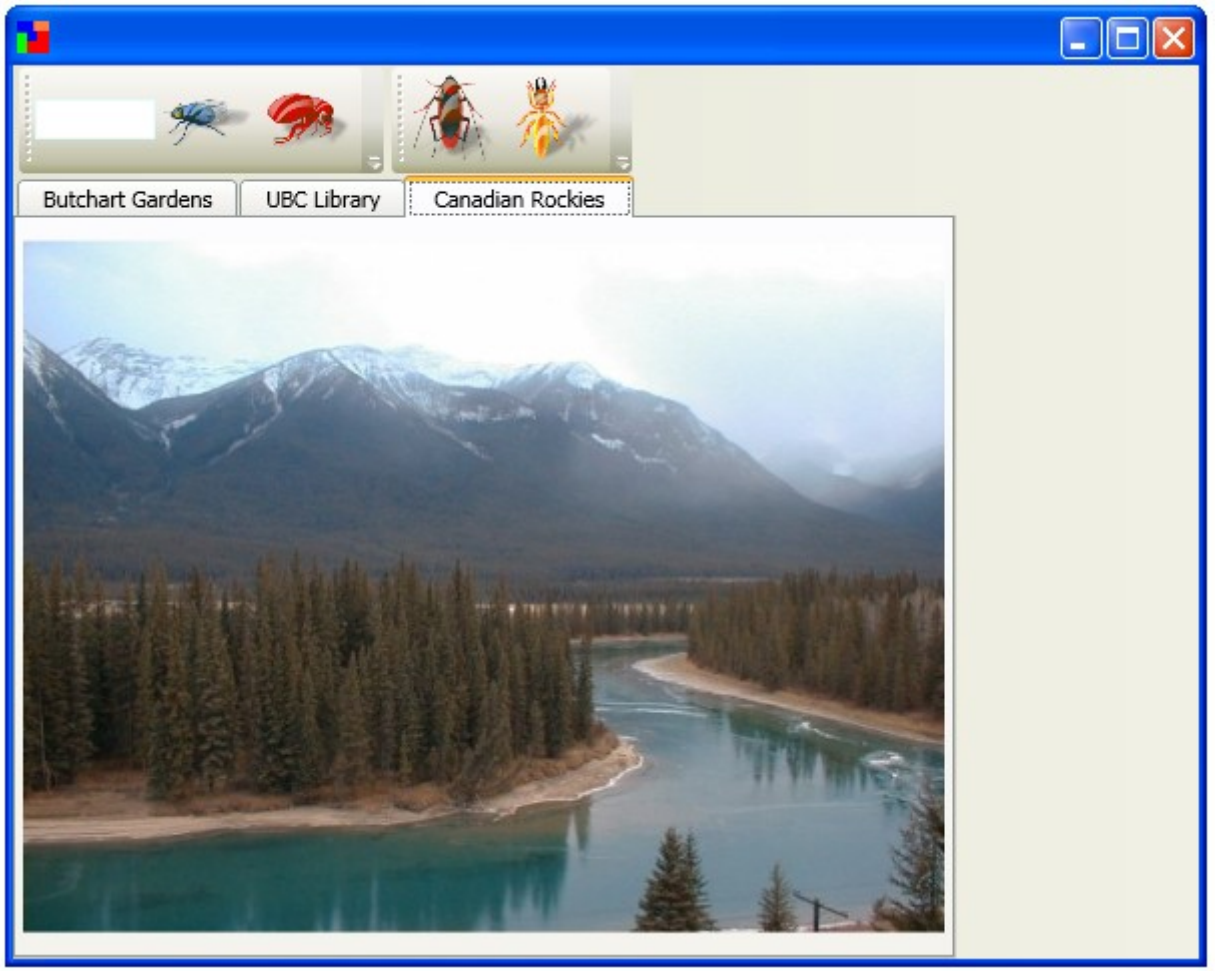
The `ToolBarTray` Control

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        x:Class="gaga.MainWindow" Width="600" Height="480" Icon="pixpix.png">

    <Grid>
        <ToolBarTray Grid.Row="0" HorizontalAlignment="Stretch">
            <ToolBar Band="0" BandIndex="0">
                <TextBox BorderBrush="LightCyan" Width="60"
                     ToolBar.OverflowMode="Never"/>
                <Button ToolBar.OverflowMode="Never">
                    <Image Source="fly.ico" Height="44" Width="44"/>
                </Button>
                <Button ToolBar.OverflowMode="Never">
                    <Image Source="flea.ico" Height="44" Width="44"/>
                </Button>
            </ToolBar>
            <ToolBar Band="0" BandIndex="1">
                <Button ToolBar.OverflowMode="Never">
                    <Image Source="cockroach.ico" Height="44" Width="44"/>
                </Button>
                <Button ToolBar.OverflowMode="Never">
                    <Image Source="termite.ico" Height="44" Width="44"/>
                </Button>
            </ToolBar>
        </ToolBarTray>
```

The `ToolBarTray` Control

```
    <TabControl HorizontalAlignment="Left" Margin="0,55,0,0" Width="471">
        <TabItem>
            <TabItem.Header>Butchart Gardens</TabItem.Header>
            <Grid>
                <Image Source="butchart.jpg" Height="373" Margin="0,2,0,0" />
            </Grid>
        </TabItem>

        <TabItem>
            <TabItem.Header>UBC Library</TabItem.Header>
            <Grid>
                <Image Source="vancouver.jpg"/>
            </Grid>
        </TabItem>

        <TabItem>
            <TabItem.Header>Canadian Rockies</TabItem.Header>
            <Grid>
                <Image Source="banf.jpg"/>
            </Grid>
        </TabItem>
    </TabControl>
    </Grid>
</Window>
```

The `ToolBarTray` Control



The `UniformGrid` Control

- The `UniformGrid` layout control displays its child controls within a grid, that its rows and columns have the same size.
- We can use the `Rows` and `Columns` properties for setting the number of rows and columns we want.
- The `UniformGrid` control divides its area in order to form the required number of rows and columns.

The `UniformGrid` Control

- The `UniformGrid` control places its child controls within the cells. Each child is placed within one cell. If the child is too big it will be clipped.

The `UniformGrid` Control

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    x:Class="gaga.MainWindow" Width="600" Height="480" Icon="pixpix.png">
        <UniformGrid Columns="4" Rows="3">
            <Button Content="A" Margin="4"/>
            <Button Content="B" Margin="4"/>
            <Button Content="C" Margin="4"/>
            <Button Content="D" Margin="4"/>
            <Button Content="E" Margin="4"/>
            <Button Content="F" Margin="4"/>
            <Button Content="G" Margin="4"/>
            <Button Content="H" Margin="4"/>
            <Button Content="I" Margin="4"/>
            <Button Content="J" Margin="4"/>
            <Button Content="K" Margin="4"/>
            <Button Content="L" Margin="4"/>
            <Button Content="M" Margin="4"/>
            <Button Content="N" Margin="4"/>
            <Button Content="O" Margin="4"/>
            <Button Content="P" Margin="4"/>
            <Button Content="Q" Margin="4"/>
            <Button Content="R" Margin="4"/>
            <Button Content="S" Margin="4"/>
            <Button Content="T" Margin="4"/>
        </UniformGrid>
</Window>
```

The `UniformGrid` Control

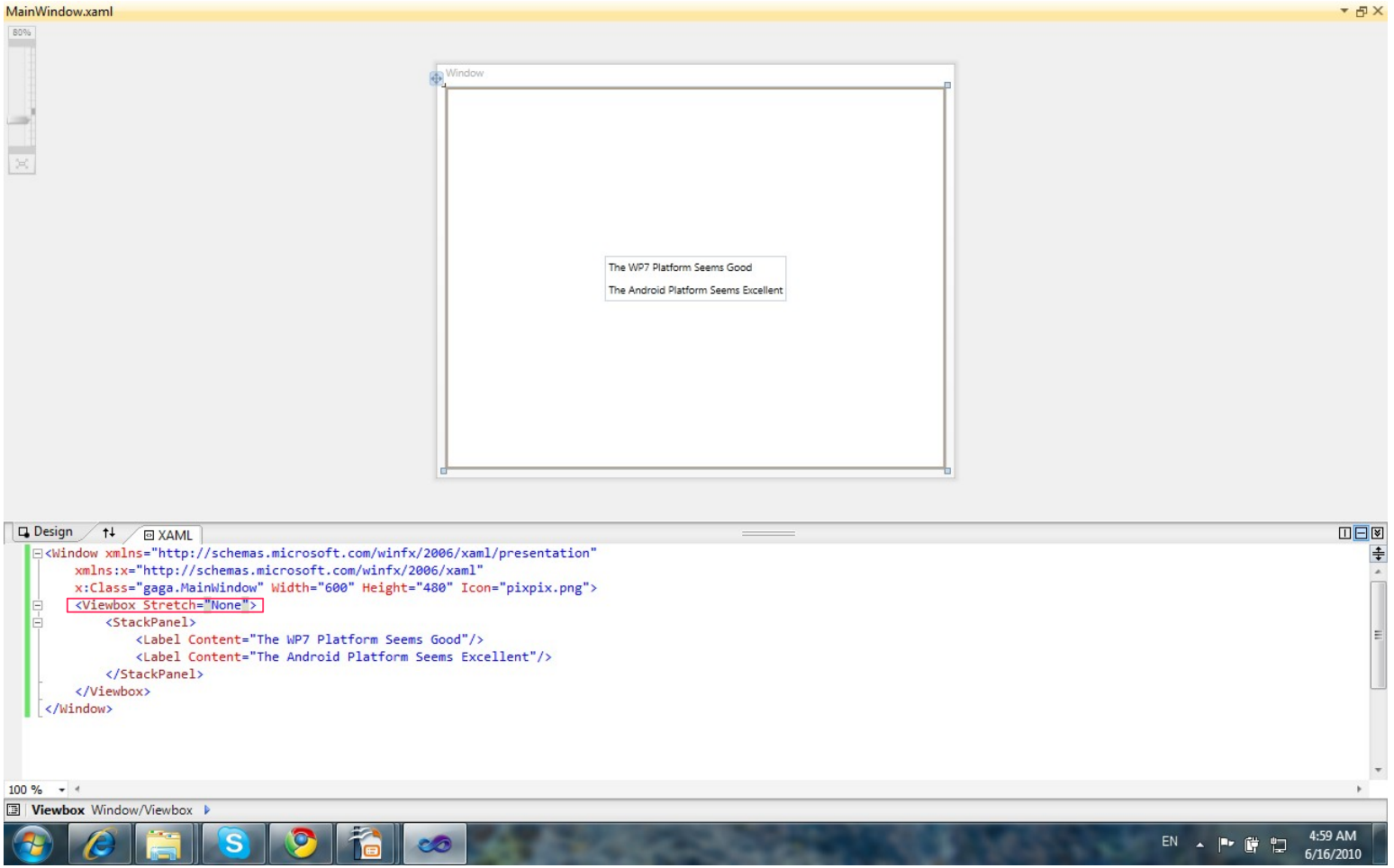
The `ViewBox` Control

- The `ViewBox` layout control can contain one child only stretching it accordingly.
- The contained child can be a another container as well.
- The Stretch property determines the way the child will be stretched. Its possible values are `None`, `Fill`, `Uniform` and `UniformToFill`.

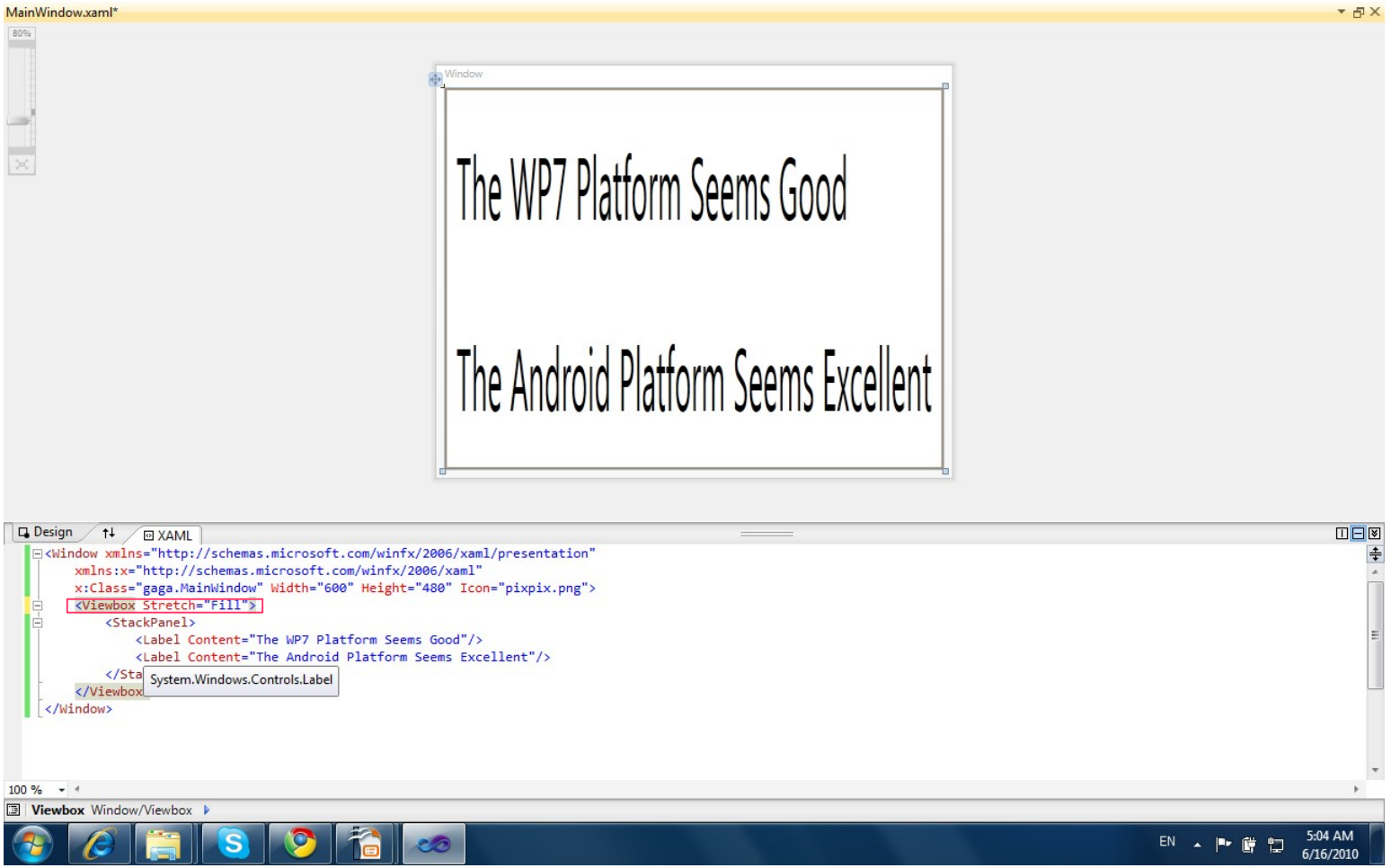
The `ViewBox` Control

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    x:Class="gaga.MainWindow" Width="600" Height="480" Icon="pixpix.png">
            <Viewbox Stretch="None">
                <StackPanel>
                    <Label Content="The WP7 Platform Seems Good"/>
                    <Label Content="The Android Platform Seems Excellent"/>
                </StackPanel>
            </Viewbox>
</Window>
```

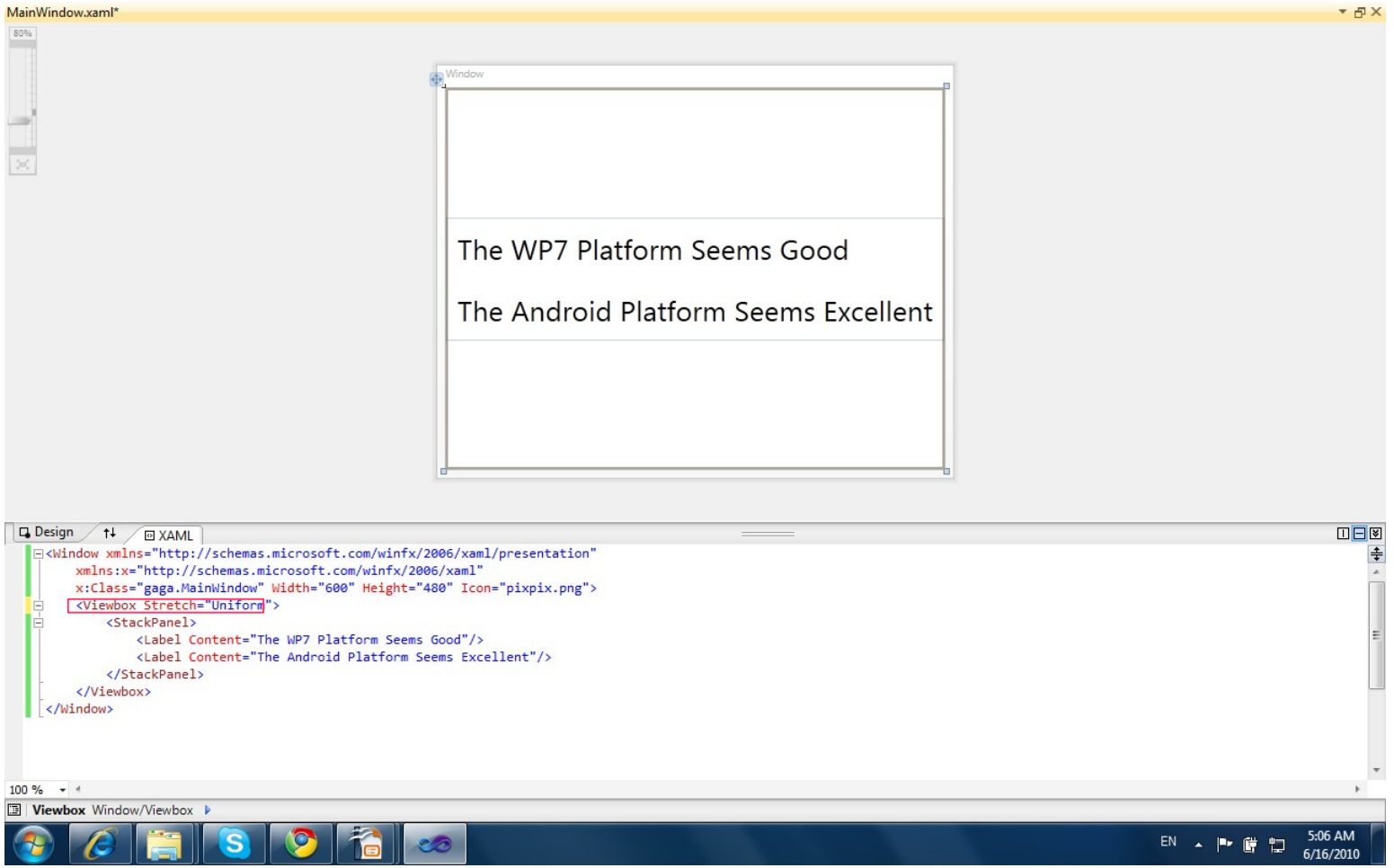
The `ViewBox` Control



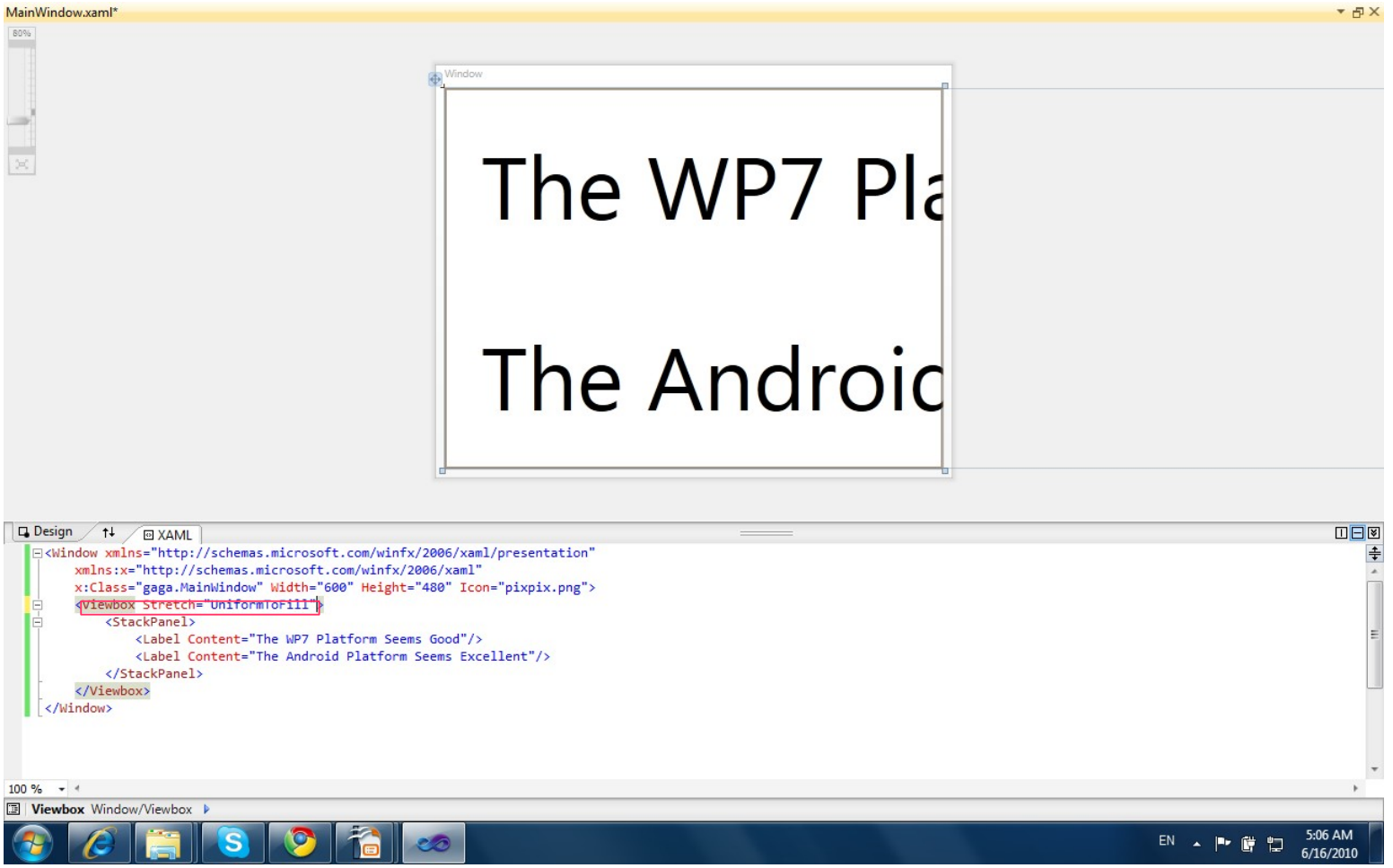
The `ViewBox` Control



The `ViewBox` Control



The `ViewBox` Control



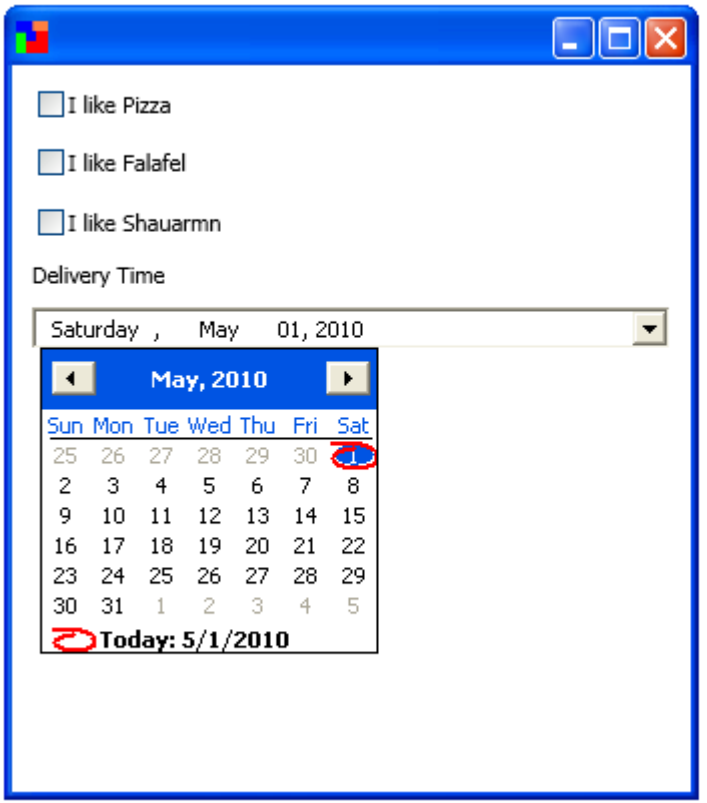
The `WindowsFormsHost` Control

- The `WindowsFormsHost` layout control can hold windows forms controls.
- In order to use this control make sure you add a reference to the `WindowsFormsIntegration` library.

The `WindowsFormsHost` Control

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:wfi="clrnamespace:System.Windows.Forms;assembly=System.Windows.Forms"

    x:Class="gaga.MainWindow" Width="600" Height="480" Icon="pixpix.png">

    <StackPanel Margin="5,5,5,5">

        <CheckBox Content="I like Pizza" Margin="8" />
        <CheckBox Content="I like Falafel" Margin="8" />
        <CheckBox Content="I like Shauarmn" Margin="8" />
        <Label Content="Delivery Time" />
        <WindowsFormsHost Margin="5">
            <WindowsFormsHost.Child>
                <wfi:DateTimePicker Enabled="True" />
            </WindowsFormsHost.Child>
        </WindowsFormsHost>

    </StackPanel>

</Window>
```

The `WindowsFormsHost` Control



The `WrapPanel` Layout Control

- The `WrapPanel` layout control arranges controls in a row.
- Unlike `StackPanel` it starts a new row when the current one runs out of room.

The `WrapPanel` Layout Control

```
<Window     xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
            xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
            x:Class="gaga.MainWindow" Width="600" Height="480"
            Icon="pixpix.png">
    <WrapPanel HorizontalAlignment="Stretch" VerticalAlignment="Stretch"
        Orientation="Horizontal">

        <Button Content="Israel" Margin="15" Width="150" Height="45"/>
        <Button Content="Canada" Margin="15" Width="150" Height="45"/>
        <Button Content="France" Margin="15" Width="150" Height="45"/>
        <Button Content="India" Margin="15" Width="150" Height="45"/>
        <Button Content="Mexico" Margin="15" Width="150" Height="45"/>

    </WrapPanel>
</Window>
```

The `WrapPanel` Layout Control

The `Expander` Layout Control

- When using the `Expander` layout control we get a header and an icon the user can click in order to toggle the display of a single specific child.

The `Expander` Layout Control

```
<Window      xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" FontSize="14"
             x:Class="gaga.MainWindow" Title="Simple Demo" Width="800" Height="600">

    <StackPanel>

        <Expander    HorizontalAlignment="Left" VerticalAlignment="Top"
                     Header="Personal Information" Margin="2" IsExpanded="True">
            <Grid Margin="3" Width="400">
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="140"/>
                    <ColumnDefinition Width="*"/>
                </Grid.ColumnDefinitions>
                <Grid.RowDefinitions>
                    <RowDefinition Height="30"/>
                    <RowDefinition Height="30"/>
                    <RowDefinition Height="30"/>
                </Grid.RowDefinitions>
                <Label Grid.Row="0" Grid.Column="0" Content="First Name:"/>
                <TextBox Grid.Row="0" Grid.Column="1" Height="20" />
                <Label Grid.Row="1" Grid.Column="0" Content="Last Name:"/>
                <TextBox Grid.Row="1" Grid.Column="1" Height="20" />
                <Label Grid.Row="2" Grid.Column="0" Content="Email Address:"/>
                <TextBox Grid.Row="2" Grid.Column="1" Height="20" />
            </Grid>
        </Expander>
```

The `Expander` Layout Control

```
        <Expander    Margin="2" HorizontalAlignment="Left" VerticalAlignment="Top"
                     Header="Home Address">
            <Grid Margin="3" Width="400">
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="140"/>
                    <ColumnDefinition Width="*"/>
                </Grid.ColumnDefinitions>
                <Grid.RowDefinitions>
                    <RowDefinition Height="30"/>
                    <RowDefinition Height="30"/>
                    <RowDefinition Height="30"/>
                </Grid.RowDefinitions>
                <Label Grid.Row="0" Grid.Column="0" Content="Street:"/>
                <TextBox Grid.Row="0" Grid.Column="1" Height="20"/>
                <Label Grid.Row="1" Grid.Column="0" Content="City:"/>
                <TextBox Grid.Row="1" Grid.Column="1" Height="20"/>
                <Label Grid.Row="2" Grid.Column="0" Content="Country:"/>
                <TextBox Grid.Row="2" Grid.Column="1" Height="20"/>
            </Grid>
        </Expander>

    </StackPanel>

</Window>
```

The `Expander` Layout Control

The `StatusBar` Control

- The `StatusBar` control is displayed on the bottom of the window. We can use that control for displaying status information.

The `StatusBar` Control

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        x:Class="gaga.MainWindow" Width="200" Height="200" Icon="pixpix.png">

    <StatusBar  HorizontalAlignment="Stretch"
                VerticalAlignment="Bottom"
                FontWeight="Bold">
        <StatusBar.Background>
            <LinearGradientBrush StartPoint="0.5,1" EndPoint="0.5,0">
                <GradientStop Color="Black" Offset="0"/>
                <GradientStop Color="Red" Offset="1"/>
            </LinearGradientBrush>
        </StatusBar.Background>
        <StatusBarItem>
            <Label Foreground="White" Name="status_label"
                 Content="This is The Status Bar"/>
        </StatusBarItem>
    </StatusBar>

</Window>
```

The `StatusBar` Control