

Control Templates

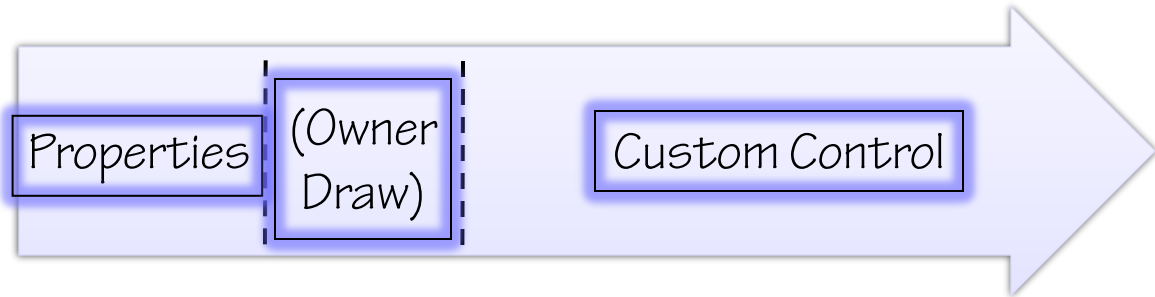


Outline

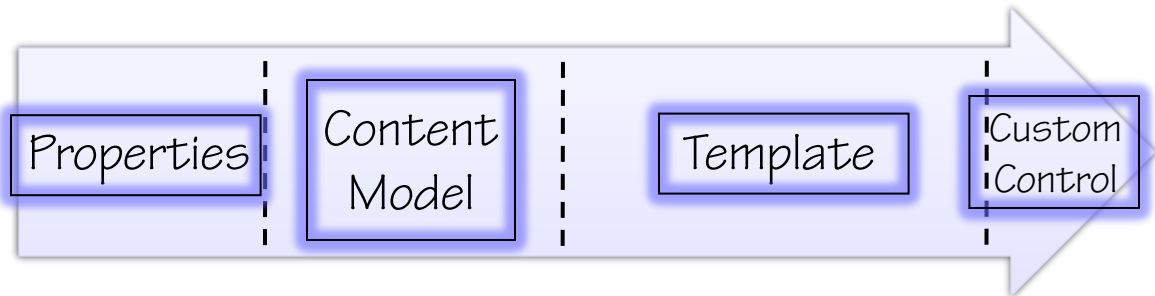
- Customization continuum
- 'Lookless' controls
- ControlTemplate and TemplateBinding
- Triggers
- Contract between control and template
- Visual and logical trees

Customization Continuum

**Before
WPF:**

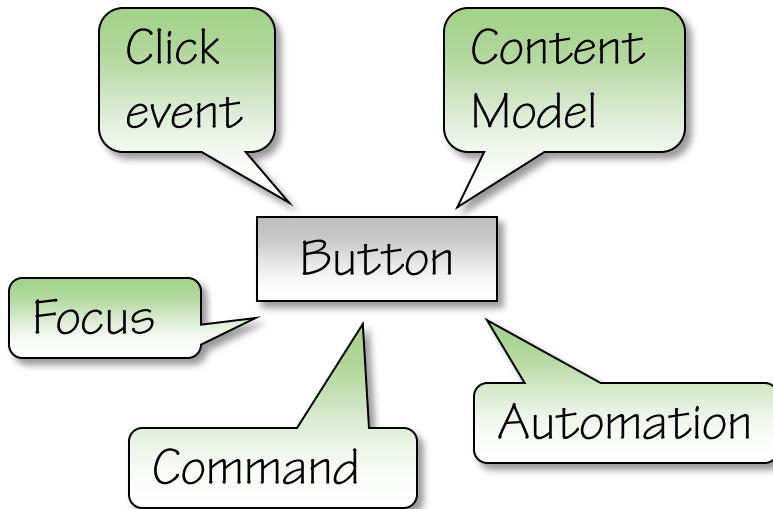


WPF:



Lookless Controls

Control



Template



Control Templates

```
<Button Width="75" Height="35">  
  <Button.Template>  
    <ControlTemplate TargetType="{x:Type Button}">  
      <Ellipse Fill="Red" />  
    </ControlTemplate>  
  </Button.Template>  
  
  _Click me  
</Button>
```



TemplateBinding

```
<Button Width="75" Height="35">  
  <Button.Template>  
    <ControlTemplate TargetType="{x:Type Button}">  
      <Ellipse Fill="{TemplateBinding Background}" />  
    </ControlTemplate>  
  </Button.Template>  
  
  _Click me  
</Button>
```



Triggers

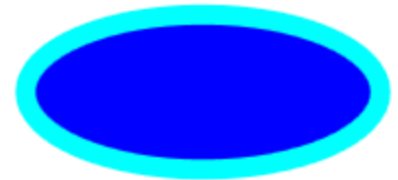
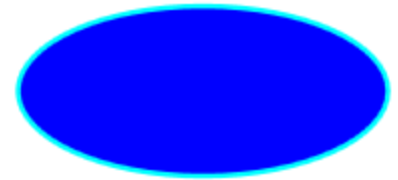
```
<Button Width="75" Height="35">
  <Button.Template>
    <ControlTemplate TargetType="{x:Type Button}">
      <Ellipse Fill="{TemplateBinding Background}" />
      <ControlTemplate.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
          <Setter Property="Background" Value="Blue" />
        </Trigger>
        <Trigger Property="IsPressed" Value="True">
          <Setter Property="Background" Value="Cyan" />
        </Trigger>
      </ControlTemplate.Triggers>
    </ControlTemplate>
  </Button.Template>

  _Click me
</Button>
```

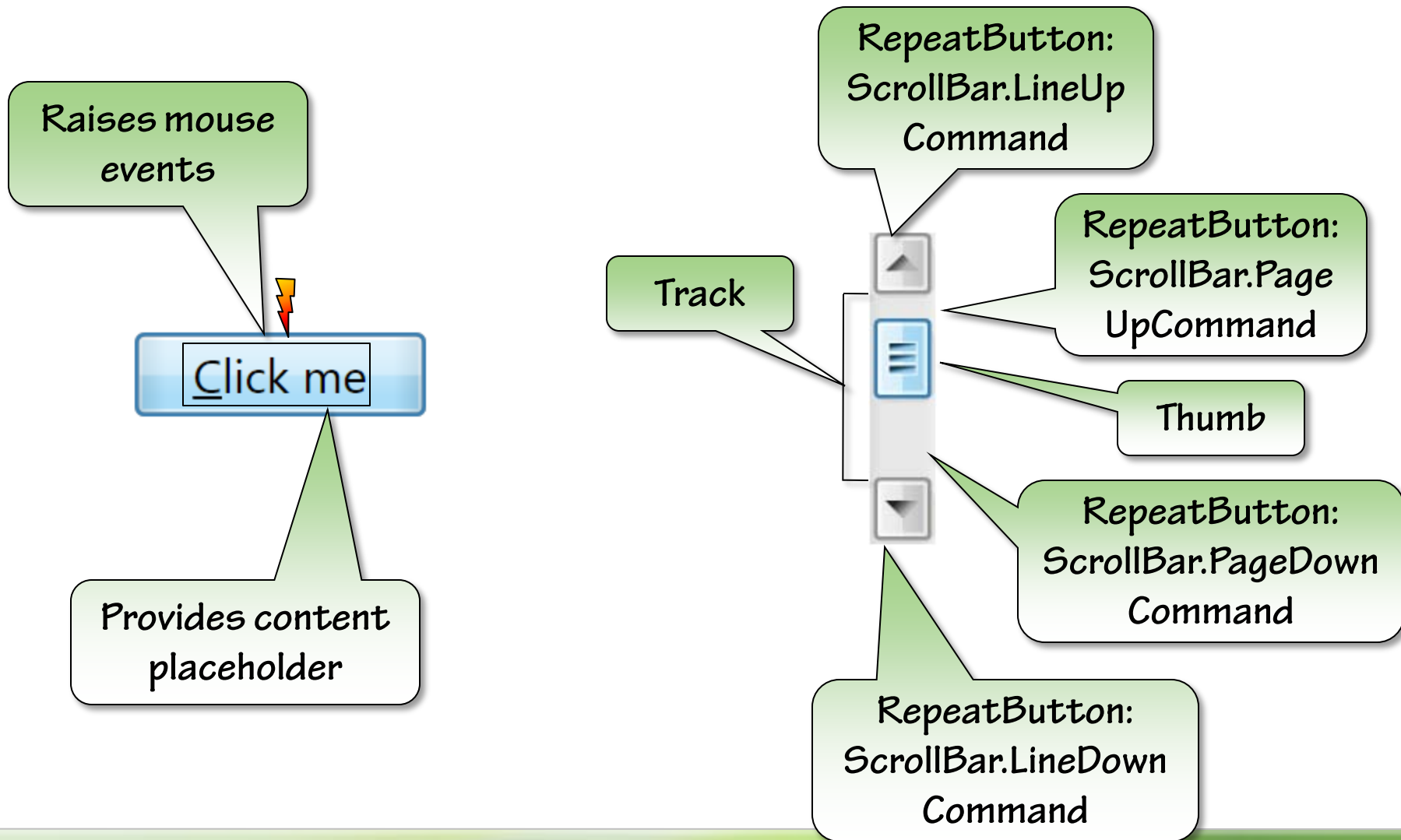
Triggers Targeting Template Elements

```
<Button Width="75" Height="35">
  <Button.Template>
    <ControlTemplate TargetType="{x:Type Button}">
      <Ellipse x:Name="buttonFace"
        Fill="{TemplateBinding Background}" />
      <ControlTemplate.Triggers>
        <Trigger Property="IsMouseOver" Value="True">
          <Setter TargetName="buttonFace"
            Property="Fill" Value="Blue" />
          <Setter TargetName="buttonFace"
            Property="Stroke" Value="Aqua" />
        </Trigger>
        <Trigger Property="IsPressed" Value="True">
          <Setter TargetName="buttonFace"
            Property="StrokeThickness" Value="4" />
        </Trigger>
      </ControlTemplate.Triggers>
    </ControlTemplate>
  </Button.Template>

  _Click me
</Button>
```



Contract between Control and Template



Contract Kinds

- **Named parts**
- **Common idioms**
 - ContentControl, HeaderedControlControl
 - ItemsControl
 - HeaderedItemsControl
- **Control-specific**

Named Parts



```
<ProgressBar Height="15" Width="100" Value="30" Maximum="100">
  <ProgressBar.Template>
    <ControlTemplate TargetType="{x:Type ProgressBar}">
      <Grid>
        <Border Name="PART_Track" CornerRadius="2"
          BorderBrush="Black" BorderThickness="1">
          <Border.Background>
            <LinearGradientBrush StartPoint="0,0" EndPoint="0,1">
              <GradientStop Color="Black" />
              <GradientStop Color="#e00" Offset="1" />
            </LinearGradientBrush>
          </Border.Background>
        </Border>
        <Border Name="PART_Indicator" CornerRadius="2"
          Margin="1" Background="#900" BorderBrush="#fa0"
          BorderThickness="1" HorizontalAlignment="Left" />
      </Grid>
    </ControlTemplate>
  </ProgressBar.Template>
</ProgressBar>
```



ContentControl

```
<Button>
  <Button.Template>
    <ControlTemplate TargetType="{x:Type Button}">
      <Grid>
        <Ellipse x:Name="buttonFace" Margin="0,2,0,0"
          Fill="{TemplateBinding Background}"
          StrokeThickness="1" Stroke="Gray" />
        <Border Margin="6">
          <ContentPresenter
            HorizontalAlignment="{TemplateBinding
              HorizontalContentAlignment}"
            VerticalAlignment="{TemplateBinding
              VerticalContentAlignment}"
            Margin="{TemplateBinding Padding}"
            RecognizesAccessKey="True" />
        </Border>
      </Grid>
    </ControlTemplate>
  </Button.Template>

  _Click me
</Button>
```

Click me

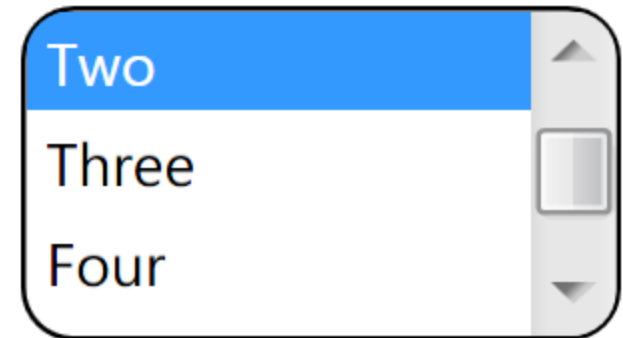


ItemsControl Template Options

- **ItemsPanel**
 - ItemsPanelTemplate
- **ItemTemplate**
 - DataTemplate
- **Template (whole control)**
 - ControlTemplate

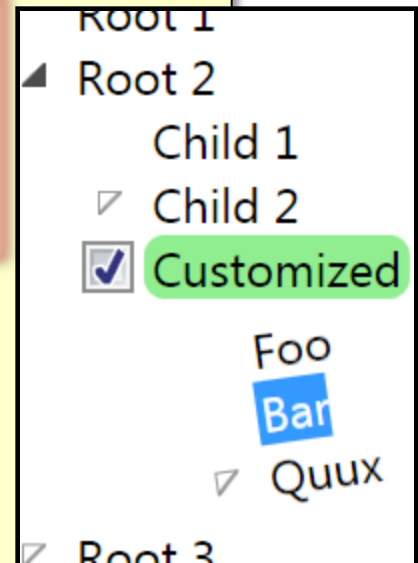
ItemsControl Full Template

```
<ListBox ItemsSource="{Binding}"
        Width="120" Height="67">
  <ListBox.Template>
    <ControlTemplate
      TargetType="{x:Type ListBox}">
      <Border BorderThickness="1"
        BorderBrush="Black"
        CornerRadius="10">
        <ScrollViewer>
          <ScrollViewer.Clip>
            <RectangleGeometry
              Rect="0,0,118,65"
              RadiusX="9" RadiusY="9" />
          </ScrollViewer.Clip>
          <VirtualizingStackPanel
            IsItemsHost="True" />
        </ScrollViewer>
      </Border>
    </ControlTemplate>
  </ListBox.Template>
</ListBox>
```

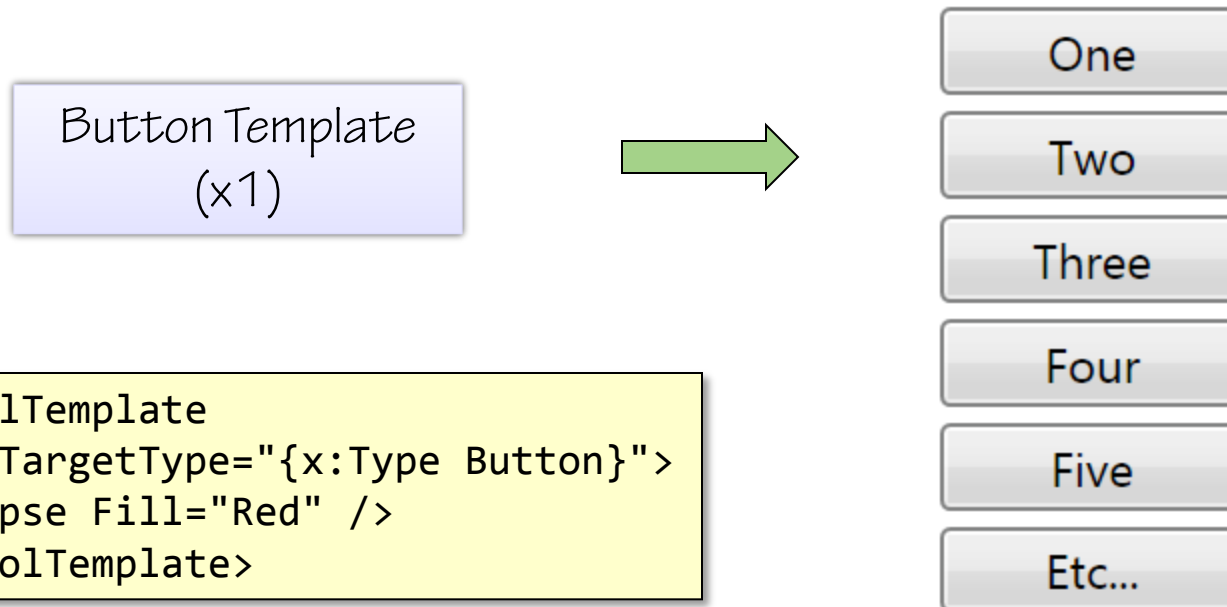


HeaderedItemsControl

```
<ControlTemplate TargetType="{x:Type TreeViewItem}">
<Grid>
    ...
    <CheckBox HorizontalAlignment="Center" VerticalAlignment="Center"
        IsChecked="{Binding Path=IsExpanded,
            RelativeSource={RelativeSource TemplatedParent}}" />
    <Grid Grid.Row="0" Grid.Column="1" HorizontalAlignment="Left">
        <Rectangle Fill="LightGreen" RadiusX="5" RadiusY="5" />
        <ContentPresenter Margin="2,0" Content="{TemplateBinding Header}" />
    </Grid>
    <StackPanel IsItemsHost="True" x:Name="host" Margin="0,1"
        Grid.Row="1" Grid.Column="1" >
        <StackPanel.LayoutTransform> <RotateTransform Angle="-8" />
    </StackPanel.LayoutTransform>
    </StackPanel>
</Grid>
<ControlTemplate.Triggers>
    <Trigger Property="IsExpanded" Value="false">
        <Setter TargetName="host" Property="Visibility"
            Value="Collapsed"/>
    </Trigger>
</ControlTemplate.Triggers>
</ControlTemplate>
```



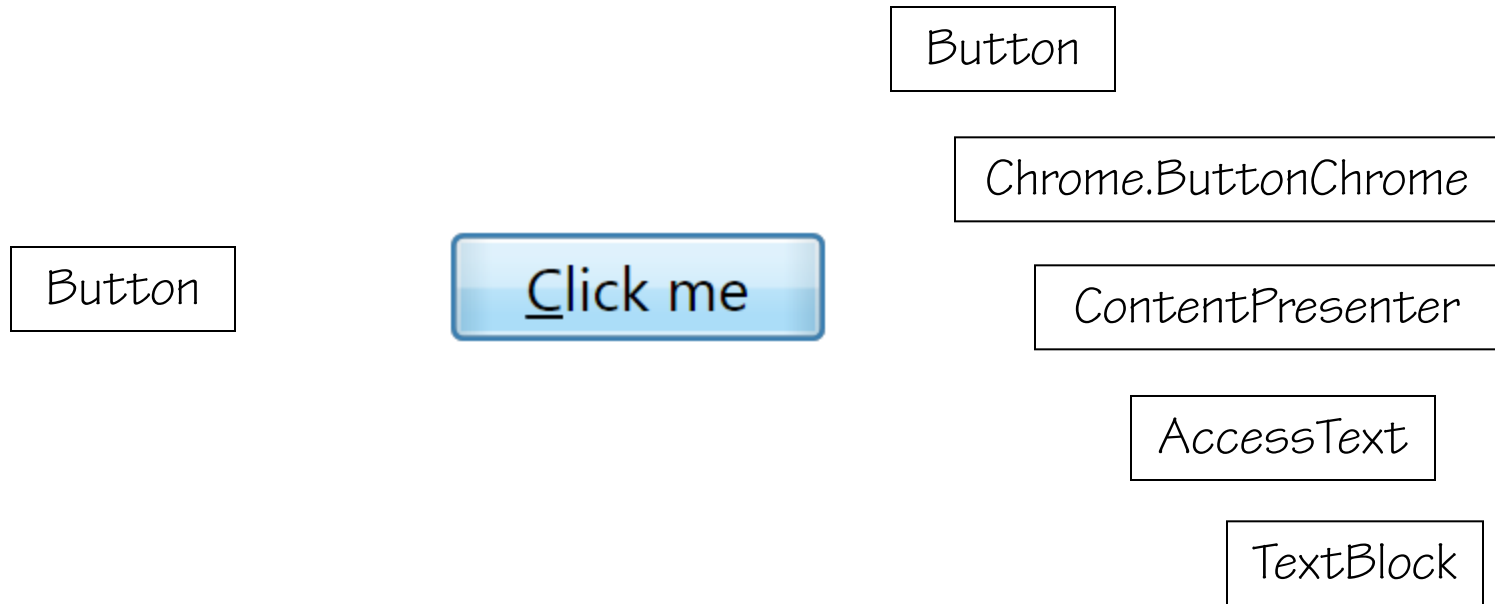
Templates are Factories



```
<ControlTemplate
    TargetType="{x:Type Button}">
    <Ellipse Fill="Red" />
</ControlTemplate>
```

```
ControlTemplate bt = new ControlTemplate(typeof(Button));
FrameworkElementFactory fact =
    new FrameworkElementFactory(typeof(Ellipse));
fact.SetValue(Ellipse.FillProperty, Brushes.Red);
bt.VisualTree = fact;
```


Logical and Visual Trees



Summary

- Customization continuum
- 'Lookless' controls
- ControlTemplate and TemplateBinding
- Triggers
- Contract between control and template
- Visual and logical trees