

Deep Dive on DataTemplates

Brian Noyes
CTO, Solliance (www.solliance.net)
brian.noyes@solliance.net, @briannoyes



pluralsight 
hardcore developer training

Outline

DataTemplates
Overview

Explicit
DataTemplates

Implicit
DataTemplates

DataTemplate
Selectors

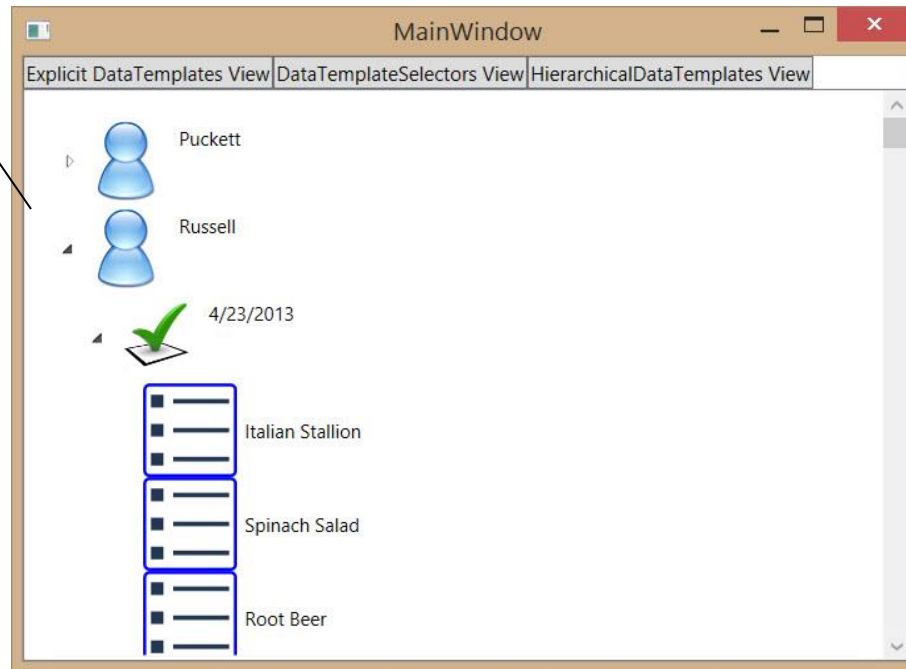
Hierarchical
DataTemplates

DataTemplates Overview

```
<ContentControl Grid.Row="1" Content="{Binding CurrentViewModel}"/>
```

DataContext=CurrentViewModel

```
<DataTemplate DataType="{x:Type local:HierachicalDataTemplatesViewModel}">  
  <local:HierarchicalDataTemplatesView/>  
</DataTemplate>
```



DataTemplates Overview

```
<ListBox x:Name="ProductsListBox" ItemsSource="{Binding Products}"
  ItemTemplate="{StaticResource ProductTemplate}">
</ListBox>
```

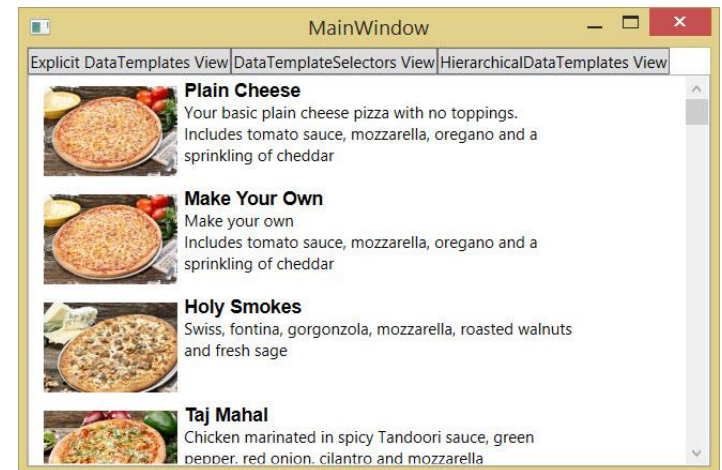
```
<DataTemplate x:Key="ProductTemplate">
  <Grid>
    <Grid.ColumnDefinitions>
      <ColumnDefinition Width="Auto"/>
      <ColumnDefinition Width="Auto"/>
    </Grid.ColumnDefinitions>
    <Image x:Name="ProductImage" Source="{Binding Image}" Margin="5" Width="100"/>
    <Grid Grid.Column="1" Height="{Binding ElementName=ProductImage, Path=Height}">
      <Grid.RowDefinitions>
        <RowDefinition Height="Auto"/>
        <RowDefinition Height="Auto"/>
      </Grid.RowDefinitions>
      <TextBlock Text="{Binding Name}" FontFamily="Arial" FontSize="14" FontWeight="Bold"/>
      <TextBlock Grid.Row="1" Text="{Binding Description}" TextWrapping="Wrap" Width="300"/>
    </Grid>
  </Grid>
</DataTemplate>
```

DataContext=Product



Make Your Own

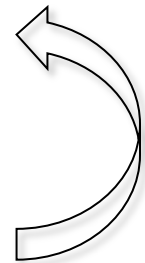
Make your own
Includes tomato sauce, mozzarella, oregano and a
sprinkling of cheddar



Explicit DataTemplates

- Define DataTemplate as a Resource with an x:Key
- Point to that template (StaticResource or DynamicResource)
 - ContentControl.ContentTemplate
 - ItemsControl.ItemTemplate
- It will be used to render the data-bound object(s)

```
<UserControl ...>  
  <UserControl.Resources>  
    <DataTemplate x:Key="ProductTemplate">  
      <Grid> ... </Grid>  
    </DataTemplate>  
  </UserControl.Resources>  
  <Grid>  
    <ListBox ItemTemplate="{StaticResource ProductTemplate}"  
             ItemsSource="{Binding Products}" />  
  </Grid>  
</UserControl>
```



Implicit DataTemplates

- Associated with a data-bound object by type
- DataTemplate must:
 - Be defined in Resource dictionary that is in scope where the data-bound control (ItemsControl / ContentControl) is used
 - Have DataType that matches the concrete type of the data bound object
 - Using {x:Type} markup extension

```
<Window>  
  <Window.Resources>  
    <DataTemplate DataType="{x:Type local:ExplicitDataTemplatesViewModel}">  
      <local:ExplicitDataTemplatesView/>  
    </DataTemplate>  
    ...  
  </Window.Resources>  
</Grid>  
  <ContentControl Grid.Row="1" Content="{Binding CurrentViewModel}"/>  
</Grid>  
</Window>
```

- Allows the use of heterogeneous collections of DataTemplates in a single control

DataTemplateSelectors

- **May need to dynamically select DataTemplate**
 - Based on something other than the type of data-bound object
 - i.e. Property value on data bound object
- **Derive class from DataTemplateSelector**
- **Override SelectTemplate method**

```
public class ProductTemplateSelector : DataTemplateSelector
{
    public override DataTemplate SelectTemplate(object item, DependencyObject container)
    {
        DataTemplate template = LocateTemplate(item, container);
        return template;
    }
}
```

- **Use whatever logic is needed to create / select DataTemplate based on data bound item and/or containing element**
- **Set the ItemTemplateSelector / ContentTemplateSelector property on data bound control**

HierarchicalDataTemplates

- **Used to address hierarchical data models presented in a TreeView control**
- **Similar to normal DataTemplate**
 - Contents will be rendered for current tree node data object
- **HierarchicalDataTemplate.ItemsSource property**
 - Bound to collection property exposed on current node's data object
 - Results in rendering a set of child nodes for those child objects
- **HierarchicalDataTemplate.ItemTemplate property**
 - Identify the DataTemplate to use when rendering the node's ItemsSource collection items
- **HierarchicalDataTemplate.ItemTemplateSelector**
 - Dynamic selection of child node DataTemplate
- **Can work with Implicit HierarchicalDataTemplates**
 - Set HierarchicalDataTemplate.DataType to {x:Type} for node data object

Summary

DataTemplates supply
dynamic content structure
based on data-bound objects

Work with ContentControls
or ItemsControls

Explicit DataTemplates wire
up named templates

Implicit DataTemplates
auto-associate template
with bound object type

DataTemplateSelectors
let you write logic to select
DataTemplate

HierarchicalDataTemplates
apply same capabilities to
tree of objects