

Data-binding Part 2




Copyright © 2008
www.wintellect.com

Composable UI model

- A core ideal in WPF is the Composable UI model
- UI can be composed of any valid WPF visual element
- Provided you understand the content rules
 - Some elements only allow single element
 - Some elements allow many children
 - Panels
 - ItemsControls
- This ideal shows up in many places
 - Replaceable UI for standard elements (example: Button)
 - Centralized declaration of UI (Templates)

Template

- Two types of Templates
 - Control templates
 - Data templates
- Control Template



Template

- What is a data template?
 - XAML markup that describes the UI for a bound object
 - A tree of elements that are expanded when showing data source
- Scope
 - Create at element level (applies to single element)
 - Create in *.Resources with DataType property
 - Applies to all rendering of data source
- Content controls
 - Render a single piece of content
 - Change the UI for content with the ContentTemplate property
- ItemsControl controls
 - Render a list of items
 - Change UI for each item with the ItemTemplate property
 - Change UI for header with the HeaderTemplate property

Template

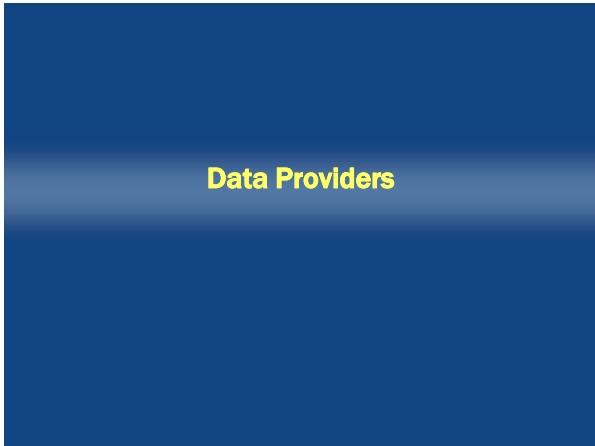
- Selecting templates
 - Default is one template for the entire list
- Can you apply different templates to common list?
 - Out of stock part
 - Premium Customers
- Template selectors
 - Data trigger:
 - Similar to property triggers
 - Change the template based on value in bound object
 - Template selector
 - Apply selection criteria to determine which template to apply

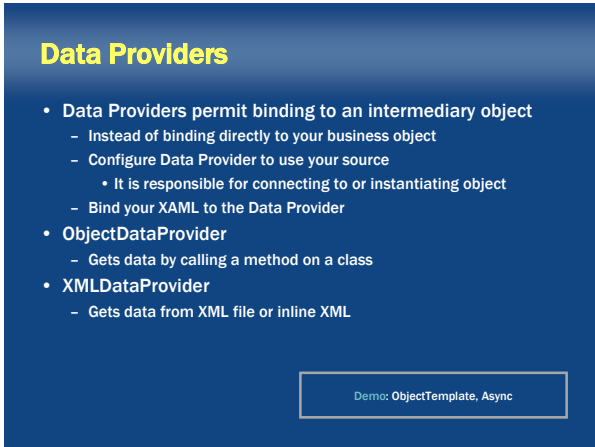
Template Selectors

- Choose DataTemplateSelector
 - When Type of each item in list is same
 - Template is selected according to property value of item
- Implicit Data Templates
 - Items in bound list are different types
 - Template is selected according to Type of item
- DataTriggers
 - Set properties while a Data Condition is true

Demo: Templates









XML Data Templates

- XML is a useful datasource for hierarchical controls
 - Treeview
 - Menus
- Specify a hierarchical data template

```
12 <HierarchicalDataTemplate DataType="{x:Type} ItemSource" Binding="{Binding XPath}"/>  
13 <!-- what elements to use in tree -->  
14 <TextBlock>Fleet Info</TextBlock>  
15 </HierarchicalDataTemplate>
```

Demo: XML Templates

Views

Working with lists

- When you bind to a list of data you need additional services
 - Selecting an item from list
 - Navigating through items
 - Filter the list
 - Organize list into groups
 - Order the list
- Accomplished in WPF via CollectionView
 - Known as DataView in other parts of .NET
 - Often just called 'views'



Data Views

- List binding requires 'broker' between target and source
 - Your target is not bound directly to source
 - The broker is the `CollectionView`
- Framework inserts a view between target and source
 - This is automatic and required for all list binding
- `CollectionView`
 - Create your own `CollectionView`
 - Get reference to automatic `CollectionView`

```
46  
47 // get the DataView that currently used by framework  
48 // pass reference to datasource (datatable, collection etc. )  
49 // you may need to cast ICollectionView to specific class  
50 // BindingListCollectionView or ListCollectionView for example  
51 ICollectionView view = CollectionViewSource.GetDefaultView(yourDataSource);  
52
```

CollectionView

- Sorting
- Grouping
- Filtering
- Navigation

Demo: Data Views

DataProviders

- WPF ships with several Data Providers
- Allow you to define a data provider in resources
- Use the resource in you XAML to specify source



Questions