Module 3

"Data Binding Collections"

TEKNOLOGISK
INSTITUT

Agenda

- **Data Binding to Collections**
- Data Templates, Grouping, and Sorting
- Change Notification

Binding Item Controls

- Item controls can be bound to a collection of objects

- `ItemsControl.`
  - `ItemsSource`
  - `DisplayMemberPath`
  - `IsSynchronizedWithCurrentItem`
    - Is `ItemControl.CurrentItem` synchronized with `ICollectionView.CurrentItem`?
  - `ItemTemplate`

- Can also just bind a single property to a list

Collection Views

- A collection view manages data currency for collection
  - Is automatically generated behind the scenes
  - Retrieve via `CollectionViewSource.GetDefaultView()`

- `ICollectionView`
  - `CurrentPosition`, `CurrentItem`
  - `MoveCurrentTo`, `MoveCurrentToFirst`, `MoveCurrentToLast`, `MoveCurrentToNext`, `MoveCurrentToPrevious`, `MoveCurrentToPosition`
  - `IsCurrentBeforeFirst`, `IsCurrentAfterLast`

- `ICollectionView`
  - `IList`
    - `IBindingList`

`CollectionView`
`ListCollectionView`
`BindingListCollectionView`

Master-Detail Binding

- Master-Details view can be created by
  - Binding primary `ItemsControl` to master collection
  - Binding secondary `ItemsControl` to property (or relation) between master and details collection
  - Synchronizing on current items

- Remember to set `IsSynchronizedWithCurrentItem = true` on primary `ItemsControl`

Data Provider Classes

- ObjectDataProvider
  - ObjectType
  - ConstructorParameters
  - MethodName
  - MethodParameters
  - IsAsynchronous

- XmlDataProvider
  - Source, Document
  - XPath

DataGrid

- `DataGrid`
  - Designed for quick and easy data binding
  - Columns can be auto-generated from `ItemsSource`
  - `IsReadOnly`
- Column types

`DataGridtextColumn`	`TextBlock`	`TextBox`
`DataGridHyperlinkColumn`	`HyperLink`	
`DataGridCheckBoxColumn`	`CheckBox`	`CheckBox`
`DataGridComboBoxColumn`	`TextBlock`	`ComboBox`
`DataGridTemplateColumn`	*`CellTemplate`*	*`CellEditingTemplate`*

- Extra row details
- Column Freezing
- …

Design-time Data

- XAML allows setting specific properties applying only to design-time

```
<Window ...
   xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
   xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
   mc:Ignorable="d">
   <d:Window.DataContext>
      <clr:Participants>
         <clr:Participant FirstName="Jesper"
                          LastName="Gulmann Henriksen"
                          Company="Wincubate ApS" />
       </clr:Participants>
    </d:Window.DataContext>
   ...
</Window>
```

Agenda

- Data Binding to Collections
- **Data Templates, Grouping, and Sorting**
- Change Notification

Data Templates

- Data Templates are XAML-fragments determining how bound data is displayed

- `DataTemplate`
  - Can be set via `ContentControl.ContentTemplate` property
  - Can be set via `ItemsControl.ItemTemplate` property
    - Mutually exclusive with `DisplayMemberPath` property!
  - Can be defined
    - Directly in bound control
    - In `Resources`

- Data template can be automatically applied to all elements of a certain type
  - Set `DataTemplate.DataType`
  - Don't specify an `x:Key` for data template!

Hierarchical Data Templates

- `HierarchicalDataTemplate`
  - Excellent for hierarchical data
    - e.g. file system etc.
- Use
  - `HierarchicalDataTemplate` for internal nodes
  - "regular" DataTemplate for leaf nodes
- Note:
  - A "non-recursive" example is supplied in Lab 3.3

Sorting

- Bound data can be sorted through **`ICollectionView`**
- **`ICollectionView`**
  - **`SortDescriptions`**
    - **`SortDescription`** collection
- Specifically for **`ListCollectionView`**
  - **`CustomSort`**
    - **`IComparer`** implementation
- Note
  - The **`SortDescription`** class is defined in the **`System.ComponentModel`** namespace in **`WindowsBase.dll`**.

Grouping

- Bound data can be grouped through `ICollectionView`
- `ICollectionView`
  - `GroupDescriptions`
    - `PropertyGroupDescription` collection
- `ItemsControl`
  - `GroupStyle`
    - `HeaderTemplate`, `HeaderTemplateSelector`
    - `ContainerStyle`, `ContainerStyleSelector`
    - `Panel`
- Custom Grouping
  - Can be performed by implementing `IValueConverter` appropriately

Filtering

- Bound data can be filtered though **`ICollectionView`**

- **`ICollectionView`**
  - **`Filter`**
    - should be set to filtering predicate (of **`object`**!) in code

- This approach does not work for ADO.NET objects
  - These views are **`BindingListCollectionView`**
    - `CustomFilter = "ColumnName Operator Value"`

CollectionViewSource

- Collection views can similarly be created in XAML
  - Define a **CollectionViewSource** instance bound to data
  - Bind ItemsControl to the **CollectionViewSource** instance

```
<Window.Resources>
    <clr:Participants x:Key="participants" />
    <CollectionViewSource x:Key="cvs"
       Source="{Binding Source={StaticResource participants}}" />
</Window.Resources>
```

```
<ListBox ItemsSource="{Binding Source={StaticResource cvs}}"
         DisplayMemberPath="FullName"/>
```

- Sorting can also be applied in XAML

Agenda

- Data Binding to Collections
- Data Templates, Grouping, and Sorting
- **Change Notification**

`INotifyPropertyChanged`

- Implement **`INotifyPropertyChanged`** to propagate modifications to a single element through data binding
  - **`PropertyChanged`** event
    - Raise event with CLR property name whenever it is changed

`ObservableCollection<T>`

- Implement collections by inheriting **`ObservableCollection<T>`**
  - Located in **`WindowsBase.dll`**!
- Automatically propagates adding and removal of elements to collection
- Handling change notifications overall
  - Implement **`INotifyPropertyChanged`** on single elements of type **`T`**
  - Inherit collection storage class from **`ObservableCollection<T>`**

Summary

- Complex Data Binding
- Data Templates, Grouping, and Sorting
- Change Notification


WINCUBATE
Jesper Gulmann Henriksen
PhD, MCT, MCSD, MCPD
Phone : +45 22 12 36 31
Email : jgh@wincubate.net
WWW : http://www.wincubate.net
Ringgårdsvej 4A
8270 Højbjerg
Denmark