Data Views

Navigation,
Filtering,
Sorting and
Grouping

Agenda

- Overview
- Navigation
- Sorting
- Filtering

The View Object

- When you bind a collection to an ItemsControl (e.g. a ListBox) then a data view is automatically created and used by the binding class.
- This view sits between your data source and the bound control.
- This view tracks the current item, and it supports features such as sorting, filtering, and grouping.
- The actual kind of view that is created depends on which interface the bound data source implements:

– IBindingList	→	BindingListCollectionView
– IList	→	ListCollectionView
– IEnumerable	→	CollectionView

CollectionViewSource

- Is a framework helper class that is used to:
  - retrieve a view by calling GetDefaultView()
  - Create a new view object
    - and add your collection to the Source property, and
    - get the collection view from the View property.

- A source collection can have multiple views associated with it
  - Because a view does not change the underlying source collection.

Retrieving a View Object

- Use the static GetDefaultView() method of the CollectionViewSource class:

```
ICollectionView view =
CollectionViewSource.GetDefaultView(lstProducts.ItemsSource);
```

- Note:
  GetDefaultView() method always returns an ICollectionView reference.
  It's up to you to cast the view object to the appropriate class, such as a ListCollectionView or BindingListCollectionView, depending on the data source
  - if you need the extra functionality by the specialized view.

Navigation

- The view keeps track of the current item and it has a handful of methods to move within the list, such as:
  - MoveCurrentToFirst()
  - MoveCurrentToLast()
  - MoveCurrentToNext()
  - MoveCurrentToPrevious()
  - MoveCurrentToPosition()

Sorting with CollectionViewSource

- A CollectionViewSource has a collection of SortDescriptions
  - a SortDescription defines how to sort the data
    - the name of the property to sort by,
    - and the direction (Ascending or Descending).
- Configuring sorting in XAML:

```
xmlns:scm="clr-namespace:System.ComponentModel;assembly=WindowsBase"
…
<CollectionViewSource x:Key="processesView"
                      Source="{StaticResource processes}" >
    <CollectionViewSource.SortDescriptions>
        <scm:SortDescription PropertyName="ProcessName" />
    </CollectionViewSource.SortDescriptions>
</CollectionViewSource>
…
<ListView Name="listView1"
          ItemsSource="{Binding Source={StaticResource
                              processesView}}" >
```

Sorting with CollectionViewSource in C#

- To sort a CollectionViewSource you must:
    1. Clear the SortDescriptions collection for any existing SortDescribtions.
    2. just add a SortDescribtion to the SortDescribtions collection on the CollectionViewSource.

```
private void sortOrderCombo_SelectionChanged(object sender,
                                            SelectionChangedEventArgs e)
{
    string newSortOrder =
       ((ComboBoxItem)sortOrderCombo.SelectedItem).Content.ToString();
    SortDescription sortDesc = new SortDescription(newSortOrder,
                                            ListSortDirection.Ascending);
    CollectionViewSource src = (CollectionViewSource)FindResource
                                                      ("processesView");
    src.SortDescriptions.Clear();
    src.SortDescriptions.Add(sortDesc);
}
```

Contains the name of a property on the elements in the collection

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING

Filtering with CollectionViewSource

1. Add a handler for the Filter event on the CollectionViewSource.

```
<CollectionViewSource x:Key="processesView"
                              Source="{StaticResource processes}"
                              Filter="CollectionViewSource_Filter"/>
```

2. Implement the event handler.
   - The event handler will be called once for each row in the data source.
   - FilterEventArgs contain the current item (e.Item) and an Accepted property

```
private void CollectionViewSource_Filter(object sender,
                                                   FilterEventArgs e)
{
    Process p = e.Item as Process;
    e.Accepted = (p.BasePriority >= 8);
}
```

AARHUS UNIVERSITY
SCHOOL OF ENGINEERING