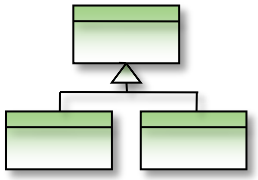
Data in WPF Applications

pluralsight
see what you can learn

Outline

- Data binding
- Data contexts
- Data templates
- Collections
- XML

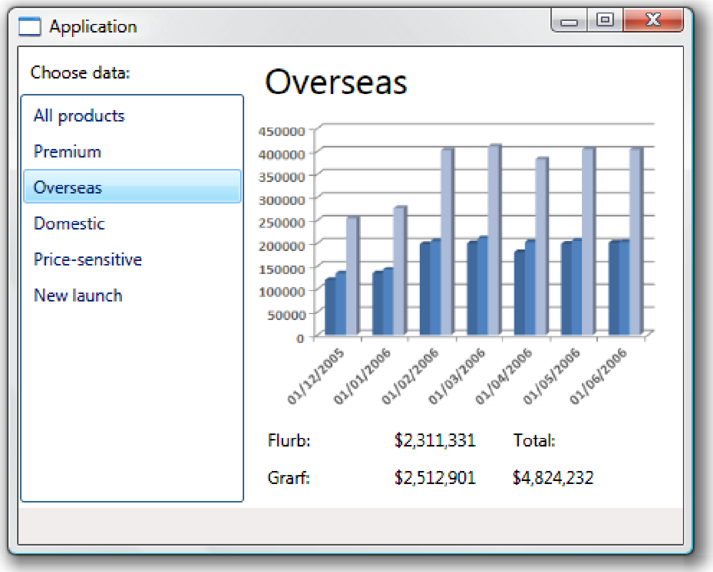
Data Binding



Data Sources

```
<order>
  <item>
    <sku>410821</sku>
    <quantity>100</quantity>
    <price currency="USD">49</price>
  </item>
 <item>
    <sku>8118055</sku>
    <quantity>4</quantity>
    <price currency="USD“>79</price>
  </item>
</order>
```

UI



Flurb:	$2,311,331	Total:
Grarf:	$2,512,901	$4,824,232

Binding Targets

```
<TextBox Text="{Binding Path=Property}" />
```

Target

- Any FrameworkElement
- Any DependencyProperty

Binding Expressions

```
<TextBox Text="{Binding Path=Property}" />
```

Property	Usage
Path	Source property
XPath	XML source node
Source	Source object
Mode	OneTime/OneWay/ OneWayToSource/TwoWay

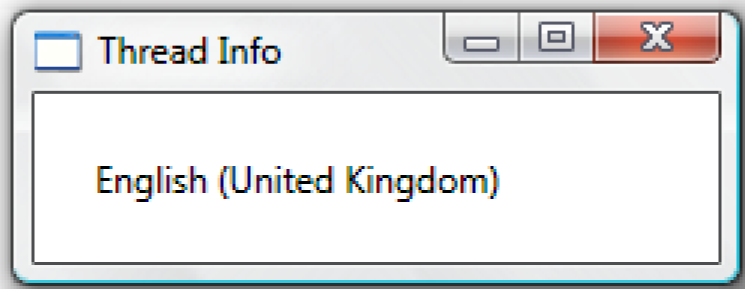
Creating Bindings in Code

```
// Equivalent to:
// Text="{Binding Path=CurrentCulture,
//                    Source={x:Static t:Thread.CurrentThread}}"

Binding binding = new Binding("CurrentCulture");
binding.Source = Thread.CurrentThread;
myTextBlock.SetBinding(TextBlock.TextProperty, binding);
```

- **FrameworkElement.SetBinding**
- **Alternative: BindingOperations**

Explicit Data Source

```
<Grid xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:t="clr-namespace:System.Threading;assembly=mscorlib">

  <TextBlock Text="{Binding Path=CurrentCulture,
                            Source={x:Static t:Thread.CurrentThread}}" />

 </Grid>
```



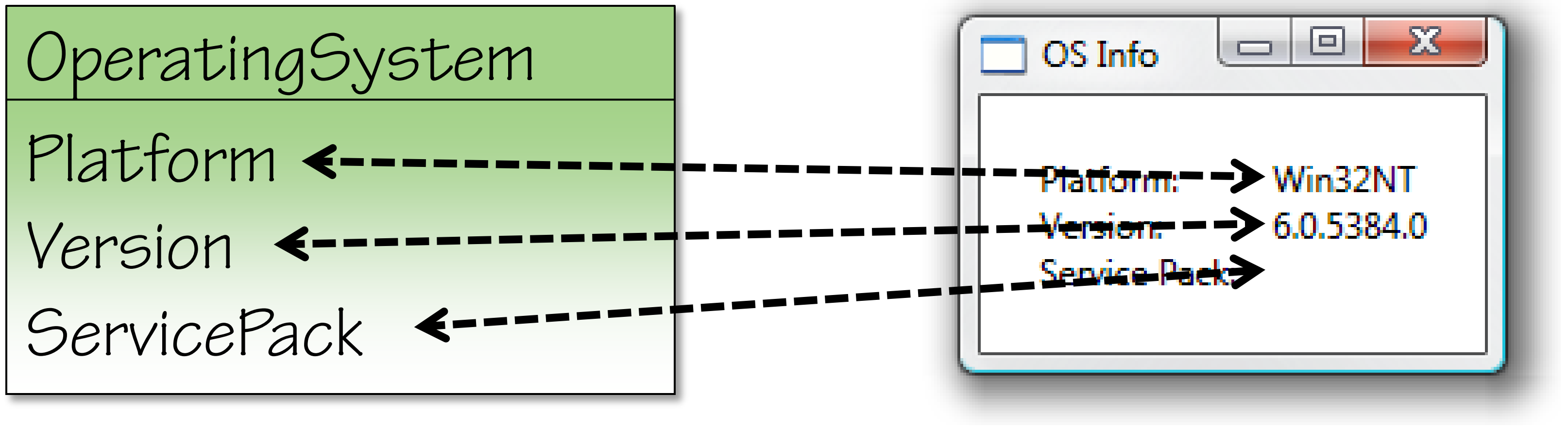
English (United Kingdom)

Explicit Data Source as Resource

```
<Grid xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:s="clr-namespace:System;assembly=mscorlib">

  <Grid.Resources>
    <s:String x:Key="val">Foo</s:String>
  </Grid.Resources>

  <StackPanel>
    <TextBlock Text="{Binding Source={StaticResource val}}" />
    <TextBlock Text="{Binding Path=Length,
                              Source={StaticResource val}}" />
  </StackPanel>

</Grid>
```

Data Contexts

- **Multiple targets, one source**



Without Data Context

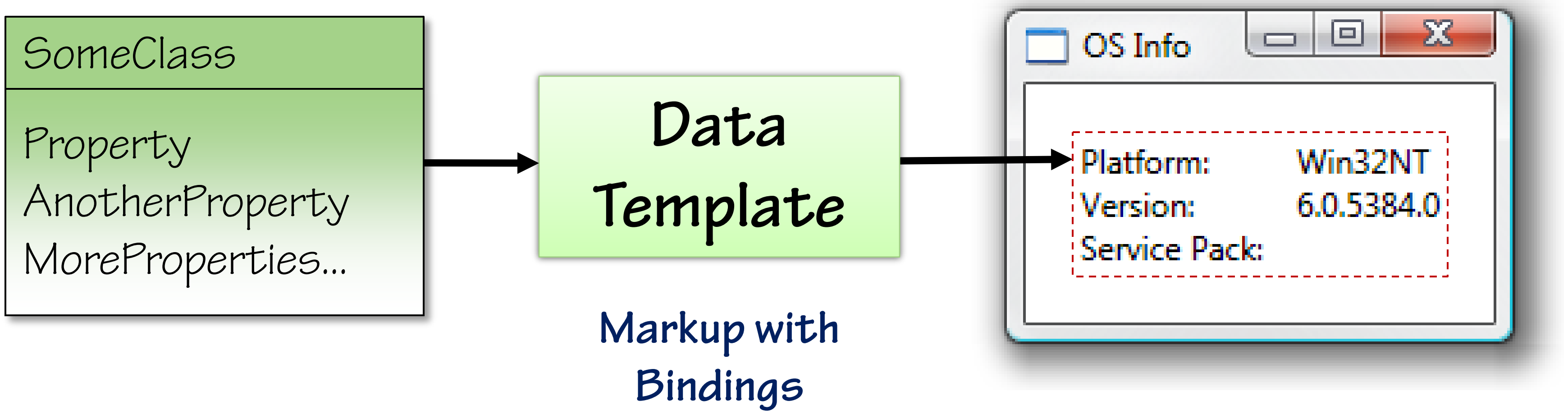
```
<Grid xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:s="clr-namespace:System;assembly=mscorlib"
      >

  <Grid.RowDefinitions>
    <RowDefinition />
    <RowDefinition />
    <RowDefinition />
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="80" />
    <ColumnDefinition />
  </Grid.ColumnDefinitions>

  <TextBlock Grid.Row="0" Grid.Column="0" Text="Platform:" />
  <TextBlock Grid.Row="0" Grid.Column="1"
    Text="{Binding Platform, Source={x:Static s:Environment.OSVersion}}" />
  <TextBlock Grid.Row="1" Grid.Column="0" Text="Version:" />
  <TextBlock Grid.Row="1" Grid.Column="1"
    Text="{Binding Version, Source={x:Static s:Environment.OSVersion}}" />
  <TextBlock Grid.Row="2" Grid.Column="0" Text="Service Pack:" />
  <TextBlock Grid.Row="2" Grid.Column="1"
    Text="{Binding ServicePack, Source={x:Static s:Environment.OSVersion}}" />

</Grid>
```

With Data Context

```
<Grid xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
      xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:s="clr-namespace:System;assembly=mscorlib"
      DataContext="{x:Static s:Environment.OSVersion}"
      >
  <Grid.RowDefinitions>
    <RowDefinition />
    <RowDefinition />
    <RowDefinition />
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition Width="80" />
    <ColumnDefinition />
  </Grid.ColumnDefinitions>

  <TextBlock Grid.Row="0" Grid.Column="0" Text="Platform:" />
  <TextBlock Grid.Row="0" Grid.Column="1"
    Text="{Binding Platform}" />
  <TextBlock Grid.Row="1" Grid.Column="0" Text="Version:" />
  <TextBlock Grid.Row="1" Grid.Column="1"
    Text="{Binding Version}" />
  <TextBlock Grid.Row="2" Grid.Column="0" Text="Service Pack:" />
  <TextBlock Grid.Row="2" Grid.Column="1"
    Text="{Binding ServicePack}" />

</Grid>
```

Data Templates

- Template defines how a type is presented



Platform: Win32NT
Version: 6.0.5384.0
Service Pack:

DataTemplate Instantiation Options

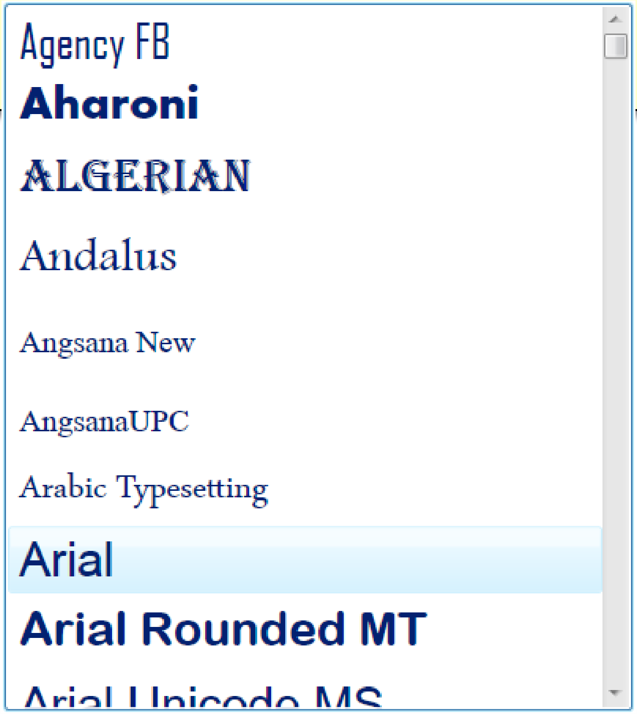
- Keyed on Type
- Inline
- Named resource

DataTemplate Triggers

```
<DataTemplate DataType="{x:Type c:Whatever}">
  ...
  <TextBlock x:Name="valueText" Text="{Binding Value}" />

  <DataTemplate.Triggers>
    <DataTrigger Binding="{Binding Alert}" Value="True">
      <Setter TargetName="valueText" Property="Foreground"
              Value="Red" />
    </DataTrigger>
  </DataTemplate.Triggers>
</DataTemplate>
```

Binding to Collections

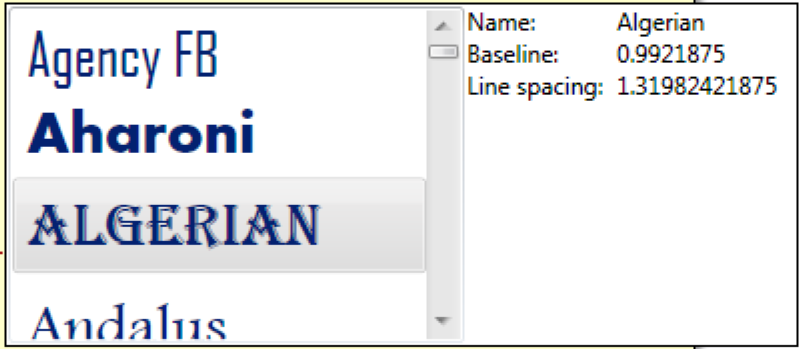
```
<ListBox ItemsSource="{x:Static Fonts.SystemFontFamilies}" Width="400">
  <ListBox.ItemTemplate>
    <DataTemplate DataType="{x:Type FontFamily}">
      <TextBlock Text="{Binding}" FontFamily="{Binding}" FontSize="30" />
    </DataTemplate>
  </ListBox.ItemTemplate>
</ListBox>
```


Agency FB
Aharoni
ALGERIAN
Andalus
Angsana New
AngsanaUPC
Arabic Typesetting
Arial
Arial Rounded MT
Arial Unicode MS

- **ItemsControl base class**
  - ListBox, ComboBox, ListView, TabControl, TreeView, Menu etc.

Master/Details

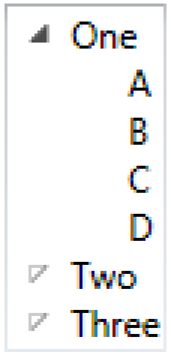
```
<Grid DataContext="{x:Static Fonts.SystemFontFamilies}">
  <Grid.RowDefinitions>
    <RowDefinition Height="Auto"/> <RowDefinition
  </Grid.RowDefinitions>
  <Grid.ColumnDefinitions>
    <ColumnDefinition/>
    <ColumnDefinition Width="75"/> <ColumnDefiniti
  </Grid.ColumnDefinitions>

  <ListBox ItemsSource="{Binding}" Grid.RowSpan="3"
           IsSynchronizedWithCurrentItem="True">
    <ListBox.ItemTemplate>
      <DataTemplate DataType="{x:Type FontFamily}">
        <TextBlock Text="{Binding}" FontFamily="{Binding}" FontSize="30" />
      </DataTemplate>
    </ListBox.ItemTemplate>
  </ListBox>

  <TextBlock Grid.Column="1" Grid.Row="0" Text="Name:" />
  <TextBlock Grid.Column="2" Grid.Row="0" Text="{Binding FamilyNames[en-US]}" />
  <TextBlock Grid.Column="1" Grid.Row="1" Text="Baseline:" />
  <TextBlock Grid.Column="2" Grid.Row="1" Text="{Binding Baseline}" />
  <TextBlock Grid.Column="1" Grid.Row="2" Text="Line spacing:" />
  <TextBlock Grid.Column="2" Grid.Row="2" Text="{Binding LineSpacing}" />
</Grid>
```

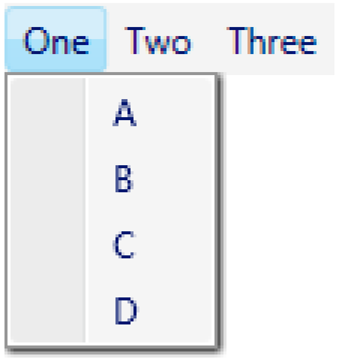

Agency FB
Aharoni
ALGERIAN
Andalus

Name:	Algerian
Baseline:	0.9921875
Line spacing:	1.31982421875

Hierarchical Binding

- TreeView, Menu, custom controls
- HierarchicalDataTemplate





```
<HierarchicalDataTemplate
            ItemsSource="{Binding Path=Children}">

  <TextBlock Text="{Binding Path=Label}" />

</HierarchicalDataTemplate>
```

Data Source Providers

- **ObjectDataProvider**
  - Construction parameters
  - Method invocation
  - Asynchronous creation

- **XmlDataProvider**
  - <x:XData> islands

XML Binding

- Use XPath instead of Path

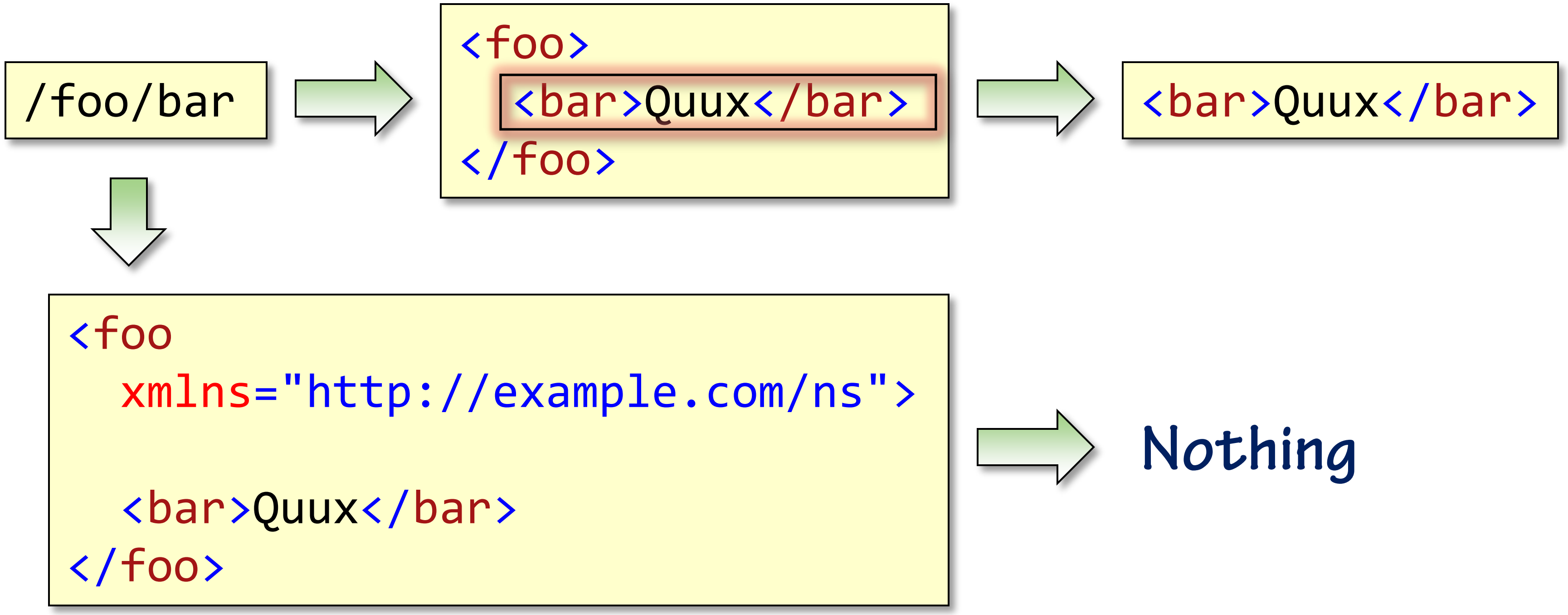
```
<HierarchicalDataTemplate
             ItemsSource="{Binding XPath=./*}">

  <TextBlock Text="{Binding XPath=@Label}" />

</HierarchicalDataTemplate>
```

XML Binding Without XmlDataProvider

```
XmlNode node = GetNodeFromSomewhere();
this.DataContext = node;
```

XML Namespaces

- XPath is namespace-aware



```
/foo/bar
```

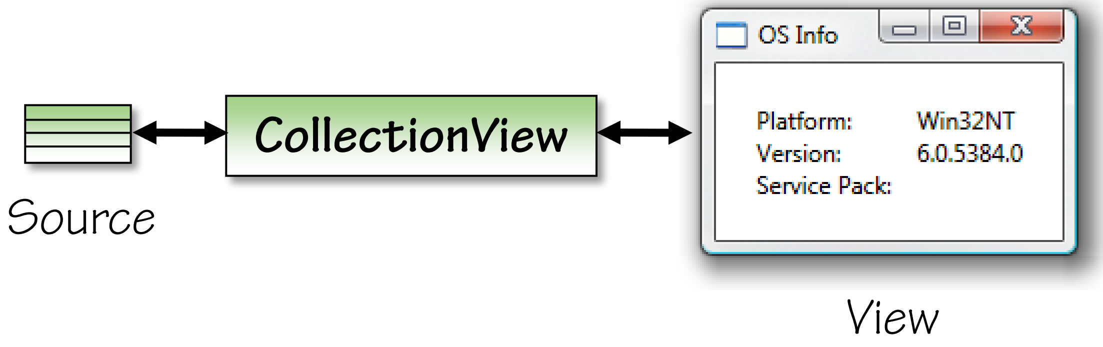
```
<foo>
  <bar>Quux</bar>
</foo>
```

```
<bar>Quux</bar>
```

```
<foo
  xmlns="http://example.com/ns">

  <bar>Quux</bar>
</foo>
```

Collection Views

- Sorting
- Filtering
- Grouping

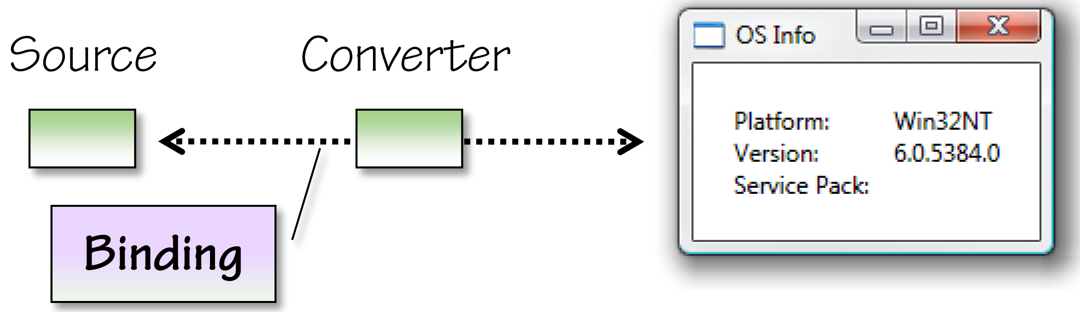


```
<CollectionViewSource x:Key="cvs"
     Source="{StaticResource origSrc}">
  <CollectionViewSource.SortDescriptions>
    <scm:SortDescription PropertyName="CityName"/>
  </CollectionViewSource.SortDescriptions>
  <CollectionViewSource.GroupDescriptions>
    <dat:PropertyGroupDescription
PropertyName="State"/>
  </CollectionViewSource.GroupDescriptions>
</CollectionViewSource>
```

Writing Data Source Classes

- **INotifyPropertyChanged**
- **INotifyCollectionChanged**
  - ObservableCollection<T>

Converters



Platform: Win32NT
Version: 6.0.5384.0
Service Pack:

```
<Grid.Resources>
  <c:MyConverter x:Name="myConverter" />
</Grid.Resources>

...

<TextBlock Text="{Binding Path=Prop,
     Converter={StaticResource myConverter}}" />
```

Validation

```
<TextBox>
  <TextBox.Text>
    <Binding Path="Foo">
      <Binding.ValidationRules>
        <ExceptionValidationRule />
        <my:MyRule />
      </Binding.ValidationRules>
    </Binding>
  </TextBox.Text>
</TextBox>
```

```
public class MyRule : ValidationRule {
  public override ValidationResult Validate(object v,
                                            CultureInfo ci) {
    if (!DoMyCheck(v))
         return new ValidationResult(false, "Oops");
    else return ValidationResult.ValidResult;
  }
}
```

Showing Validation Errors

- **Failure sets attached properties:**
  - Validation.HasErrors – Boolean
  - Validation.Errors – Array of ValidationError objects
- **Visualization options:**
  - Validation.ErrorTemplate attached property
  - Set ToolTip using Style with triggers
- **Validation.AddErrorHandler (or Validation.Error event)**
  - Binding must have NotifyOnValidationError=True

Other Binding Types

- **MultiBinding feeds multiple properties to a converter**
  - IMultiValueConverter

- **PriorityBinding selects first viable binding from list**

Summary

- Data binding
- Data contexts
- Data templates
- Collections
- XML