MODULE 16

FORMATTING BOUND DATA

MODULE TOPICS

Formatting Bound Data

StringFormat Property

Value Converters

Data Templates

STRINGFORMAT PROPERTY

- StringFormat Property is the perfect tool for formatting numbers that need to be displayed as text
- The Binding.StringFormat property can be used to convert the raw text to its display value just before it appears in the control
- When setting the StringFormat property, you use standard .NET format strings

STRINGFORMAT PROPERTY

```
<TextBlock Text="{Binding Path=UnitCost,
    StringFormat={}{0:C}}" />
<TextBlock Text="{Binding Path=Date,
    StringFormat={}{0:MM/dd/yyyy}}" />
<TextBlock Text="{Binding Source={x:Static system:DateTime.Now},
    StringFormat=Time: {0:HH:mm}}" />
```

VALUE CONVERTERS

- A value converter is responsible for converting the source data just before its displayed in the target and converting the new target value just before it's applied back to the source
- Create a class that implements IValueConverter
  - Add the ValueConversion attribute to the class
  - Implement a Convert( ) method
  - Implement a ConvertBack( ) method

VALUE CONVERTERS

```
[ValueConversion(typeof(decimal), typeof(string))]
public class PriceConverter : IValueConverter
{
    public object Convert(object value, Type targetType, object parameter, C
    {
        decimal price = (decimal)value;
        return price.ToString("c", culture);
    }

    public object ConvertBack(object value, Type targetType, object paramete
    {
        string price = value.ToString();
        decimal result;
        if (Decimal.TryParse(price, System.Globalization.NumberStyles.Any, c
```

VALUE CONVERTERS

- Define an instance of the converter as a resource
- Set the Converter property of the target element binding

```
<Window.Resources>
    <local:PriceConverter x:Key="PriceConverter"/>
</Window.Resources>

<TextBox Text="{Binding Path=UnitCost,
    Converter={StaticResource PriceConverter}}">
```

DATA TEMPLATES

- A data template is a chunk of XAML that defines how a bound data object should be displayed

```
<ListBox Name="CustomersListBox" MouseDoubleClick="CustomersListBox_MouseDoubleC
    <ListBox.ItemTemplate>
        <DataTemplate>
            <Border Margin="5" BorderThickness="1" BorderBrush="SteelBlue" Corne
                <StackPanel>
                    <TextBlock Text="{Binding Path=ContactName} FontWeight="Bold
                    <TextBlock Text="{Binding Path=CompanyName}"/>
                </StackPanel>
            </Border>
        </DataTemplate>
    </ListBox.ItemTemplate>
</ListBox>
```

DATA TEMPLATES

- Data Templates can also be stored as a reusable resource

```
<Application.Resources>
    <DataTemplate x:Key="CustomerDataTemplate">
</Application.Resources>

<ListBox ItemTemplate="{StaticResource CustomerDataTemplate}">
```

ANY QUESTIONS?