

How to debug data binding problems on Windows Phone

This article explains how to debug data binding issues by inspecting the elements before they are due to be displayed by the control.

Introduction



Data binding allows you to link a data source defined in code to a UI description in XAML (target). While defining a binding is quite straightforward, if something goes wrong it is useful to be able to tell whether the problem is in your source data or in the binding definition.

The easiest way to do this is to create a custom value converter (*IValueConverter*), which will be called when the data from the binding source arrives - before it gets delivered to the control. By setting a breakpoint in the converter you can confirm that the source is sending the expected data and contains the correct properties.



Tip: If the converter doesn't get called then this is a sign that the binding has failed earlier than expected, perhaps in the definition of the source property or an incorrect DataContext. If a wrong property name in the binding declaration is given a message in the debug output window of VS appears.

Steps to do

First, consider the case where you have a `ListBox` definition in XAML as shown below:

```
<ListBox x:Name="MainListBox" Margin="0,0,-12,0" ItemsSource="{Binding Items}" SelectionChanged="MainListBox_SelectionChanged">
  <ListBox.ItemTemplate>
    <DataTemplate>
      <StackPanel Margin="0,0,0,17" Width="432" Height="78">
        <TextBlock Text="{Binding LineOne}" TextWrapping="Wrap" Style="{StaticResource PhoneTextExtraLargeStyle}"/>
        <TextBlock Text="{Binding LineTwo}" TextWrapping="Wrap" Margin="12,-6,12,0" Style="{StaticResource PhoneTextSubtleStyle}"/>
      </StackPanel>
    </DataTemplate>
  </ListBox.ItemTemplate>
</ListBox>
```

To implement the converter you first have to derive a class from *IValueConverter* and implement it's methods:

```
public class DatabindingDebugConverter : IValueConverter
{
    public object Convert(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)
    {
        return value;
    }

    public object ConvertBack(object value, Type targetType, object parameter, System.Globalization.CultureInfo culture)
    {
        return value;
    }
}
```

Then you have to declare the converter as a static resource in your page's XAML:

```
<phone:PhoneApplicationPage.Resources>
  <local:DatabindingDebugConverter x:Key="DatabindingDebugConverter"/>
</phone:PhoneApplicationPage.Resources>
```

You also have to change the XAML data binding statement to call the converter for a particular binding:

```
<TextBlock Text="{Binding LineOne, Converter={StaticResource DatabindingDebugConverter}}"
  TextWrapping="Wrap" Style="{StaticResource PhoneTextExtraLargeStyle}"/>
```

Note that this assumes you have declared the local namespace in the page tag:

```
xmlns:local="clr-namespace:EncryptionSample"
```

Next you set a breakpoint in the `Convert()` method and voilà, you can see the values that arrive from binding, if any.

Sample code

The sample code for the [How to encrypt your application data in Windows Phone](#) article uses this technique. Or you can load the source code directly from [here](#).

