

数据处理

郑贵锋

课程目标



通过这一节课，对数据绑定基本概念有更深入的理解，掌握如何进行数据绑定。

数据绑定

SqlDataSource

AccessDataSource

ObjectDataSource

.....



数据绑定引擎

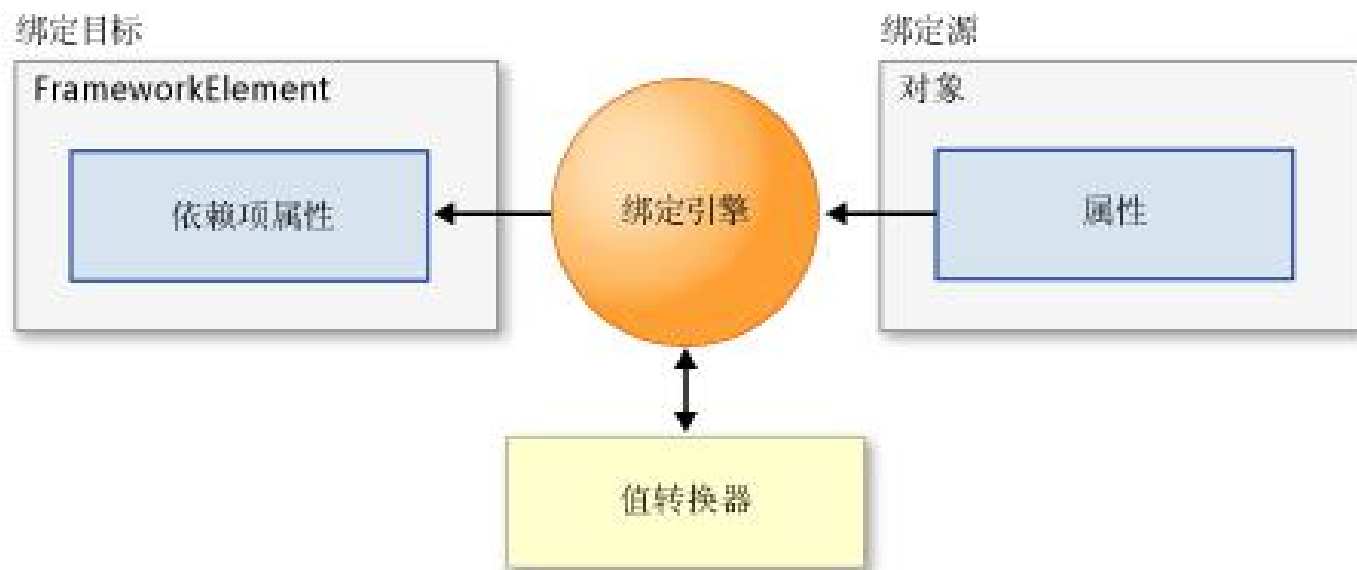
- 数据绑定引擎功能

- 实现UI元素到CLR源数据对象之间的关联
- 创建并维护二者之间的数据通道
- 将CLR源数据对象包含的数据反映到UI元素
- 将UI中对数据的更改反馈回源数据对象

- 绑定引擎基本组成要素

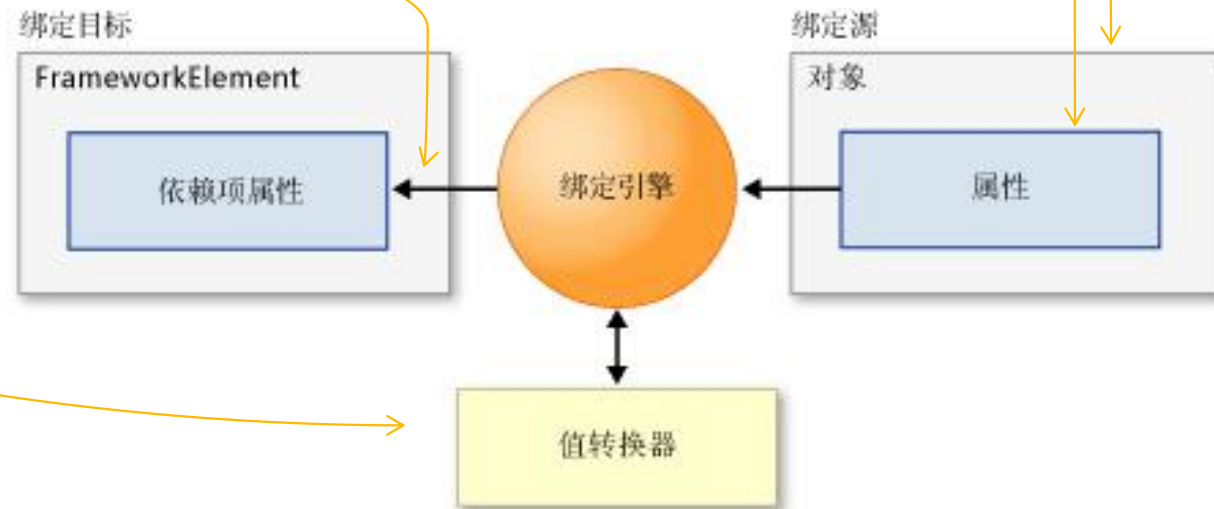
- **绑定源**：即数据的来源，可以是任意的CLR对象。实际通常需要的是该对象的某一个特定属性
- **绑定目标**：显示数据的UI元素，可以是任何FrameworkElement类型的对象。实际绑定到的也是该对象的某一个特定属性
- **绑定模式**：主要决定数据流的方向，可以是单向或双向
- **值转换器**：需要时用以进行数据类型或格式的转换

数据绑定引擎



Binding对象

- 命名空间: System.Windows.Data
- 重要属性
 - Source (ElementName / RelativeSource)
 - Path
 - Mode
 - Converter



DataContext

- DataContext: 数据上下文
- 所有UI元素都有一个DataContext属性
- 一旦为一个UI元素指定了数据上下文, 则其所有子元素都将共享该数据上下文
- 与其子元素关联的所有数据绑定在没有另行指定Source和DataContext的情况下, 都将默认使用当前数据上下文作为绑定源

Converter

- Converter: 数值转换器
- 实现绑定源属性与目标属性间的数据类型适配
- 一般用于实现简单的类型转换、格式转换
- 也可以实现一些复杂的逻辑转换（但是这并不推荐）
- Converter的类型为一个实现IValueConverter接口的类

使用Converter

//出生年份&年龄的数值转换器

```
public class BirthYearToAgeConverter:IValueConverter
{
    public object Convert(object value, Type targetType, object parameter,
System.Globalization.CultureInfo culture)
    {
        int birthYear = (int)value;
        int age = DateTime.Today.Year - birthYear;
        return age;
    }
    public object ConvertBack(object value, Type targetType, object parameter,
System.Globalization.CultureInfo culture)
    {
        int age = (int)value;
        int birthYear = DateTime.Today.Year - age;
        return birthYear;
    }
}
```

Mode

- Mode: 数值绑定模式
- Mode可选值:
 - OneTime
 - OneWay
 - TwoWay
- INotifyPropertyChanged: 属性更改通知接口
- INotifyCollectionChanged: 集合更改通知接口
 - ObservableCollection(T) 类

实现属性更改通知接

//实现了INotifyPropertyChanged接口的Employee类

```
public class Employee: INotifyPropertyChanged
{
    public event PropertyChangedEventHandler PropertyChanged;

    private string name;
    public string Name
    {
        get { return this.name; }
        set
        {
            this.name = value;
            NotifyPropertyChanged("Name");
        }
    }

    public void NotifyPropertyChanged(string propertyName)
    {
        if (PropertyChanged != null)
        {
            PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
        }
    }
}
```

XAML绑定语法

<UI标记

```
    绑定属性="{ Binding      Source=*, Path=*,  
                      Mode=*,      Converter=*,  
                      }"
```

... />

e.g.

```
<TextBox Name="txbName3"  
    Text="{Binding Path=Name,Mode=OneWay}"  
    Height="71" Width="308" />
```

XAML绑定语法

XAML Code

```
<TextBox Name="txbName3"  
    Text="{Binding Path=Name,Mode=OneWay}"  
    Height="71" Width="308" />
```

C# Code

```
canvas2.DataContext = employee;
```

集合对象数据绑定

- ListBox控件
 - ItemsControl控件
- ItemsSource
 - 实现枚举接口的CLR对象
 - ObservableCollection (可观察集合)
- DataTemplate

绑定集合对象

//ListBox数据模板

```
<ListBox Name="listEmployees" ItemsSource="{Binding}" Grid.Row="1"
VerticalAlignment="Top" Height="400" >
    <ListBox. ItemTemplate>
        <DataTemplate >
            <StackPanel Height="50" HorizontalAlignment="Center" Width="480"
VerticalAlignment="Top" Orientation="Horizontal">
                <TextBlock Height="50" HorizontalAlignment="Left"
Text="{Binding Number}" VerticalAlignment="Top" Width="120" />
                <TextBlock Height="50" HorizontalAlignment="Left"
Text="{Binding Name}" VerticalAlignment="Top" Width="120" />
                <TextBlock Height="50" HorizontalAlignment="Left"
Text="{Binding Sex}" VerticalAlignment="Top" Width="120" />
                <TextBlock Height="50" HorizontalAlignment="Center"
Text="{Binding BirthYear}" VerticalAlignment="Top" Width="120" />
            </StackPanel>
        </DataTemplate>
    </ListBox. ItemTemplate>
</ListBox>
```


//设置数据源

```
ObservableCollection<Employee> employees = new  
ObservableCollection<Employee>  
{  
    new Employee{Number=2012,Name="MagicBoy",Sex="男",BirthYear=2000},  
    new Employee{Number=1001,Name="Newpeilan",Sex="男",BirthYear=2000},  
    new Employee{Number=1002,Name="Bear",Sex="男",BirthYear=2000}  
};  
listEmployees.DataContext = employees;
```

工号	姓名	性别	出生年份
2012	MagicBoy	男	2000
1001	Newpeilan	男	2000
1002	Bear	男	2000

The Microsoft logo is centered on a solid blue background. The logo itself is white and features the word "Microsoft" in a bold, italicized sans-serif font, followed by a registered trademark symbol (®). Scattered across the blue background are several white squares of various sizes, some of which are slightly offset from each other, creating a sense of depth and movement.

Microsoft®