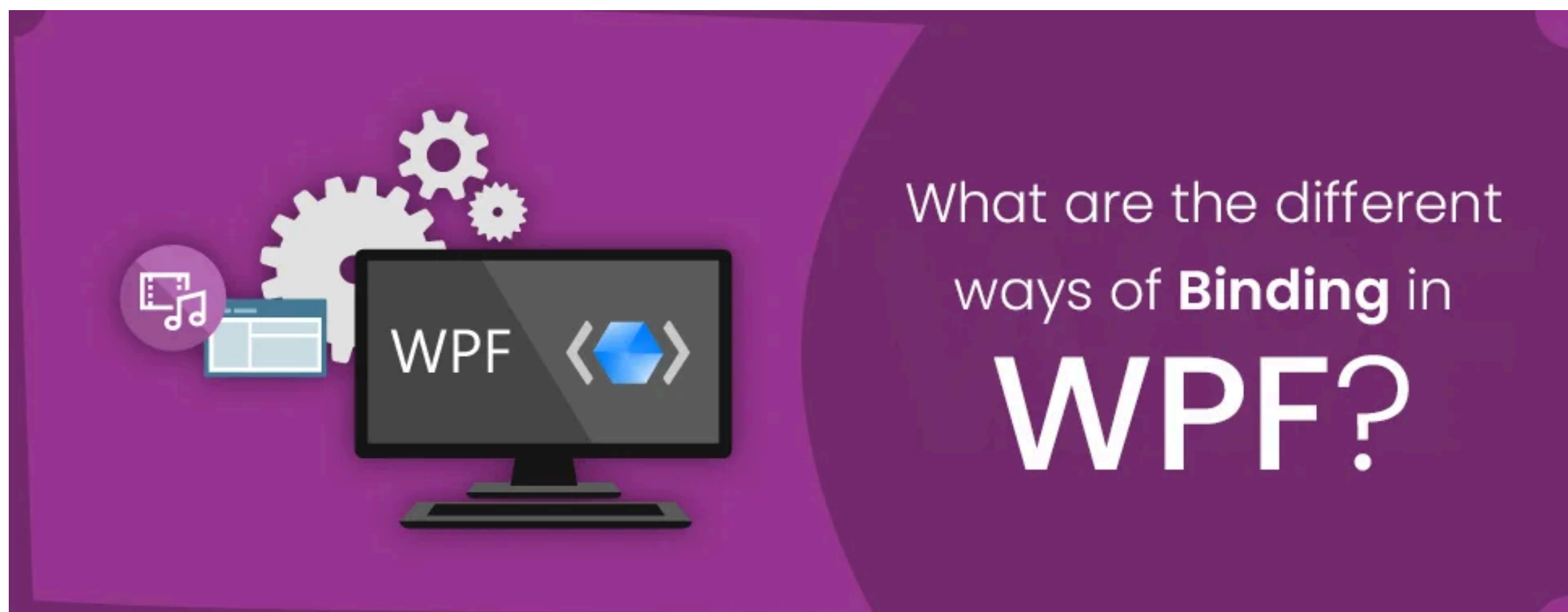


## 深入WPF数据绑定：ConverterParameter的妙用与实战

作者：微信公众号：【架构师老卢】

12-19 18:42

~ 9



**概述：**在WPF中，ConverterParameter是数据绑定的重要组成部分，用于向IValueConverter传递额外信息。通过详细实例，本文演示了ConverterParameter的基本概念、常见用法（如文本格式化、图片地址转换）以及高级技巧（多参数、动态参数绑定），为开发者提供清晰的段落和中文注释的源代码，帮助更好地理解和应用该功能。

在WPF (Windows Presentation Foundation) 中，**ConverterParameter** 是在使用数据绑定时，用于传递额外信息给 **IValueConverter** 的参数。**IValueConverter** 是一个接口，用于在数据绑定过程中转换数据的方式。

## 1. 基本概念

**ConverterParameter** 是在 **Binding** 中使用的一个属性，用于传递额外的信息给 **IValueConverter**。

```
1 | <TextBlock Text="{Binding Path=SomeProperty, Converter={StaticResource MyConverter}, ConverterParameter=SomeParameter}" />
```

在这个例子中，**ConverterParameter** 被设置为 "SomeParameter"，并且在 **IValueConverter** 的 **Convert** 或 **ConvertBack** 方法中可以通过参数获取这个值。

## 2. 常见使用方法

### 2.1 显示文本的格式化

假设有一个 **DateTime** 属性，我们希望在界面上以特定的格式显示时间。

```
1 | <TextBlock Text="{Binding Path=Date, Converter={StaticResource DateTimeConverter}, ConverterParameter=yyyy-MM-dd}" />
```

```
1 public class DateTimeConverter : IValueConverter
2 {
3     public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
4     {
5         if (value is DateTime dateTime && parameter is string format)
6         {
7             return dateTime.ToString(format);
8         }
9         return value;
10    }
11
12    public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
13    {
14        throw new NotImplementedException();
15    }
16 }
```

### 2.2 图片控件地址转换显示

假设有一个字符串属性，表示图片文件的路径，我们希望在界面上显示图片。

```
1 | <Image Source="{Binding Path=ImagePath, Converter={StaticResource ImagePathConverter}, ConverterParameter=100}" />
```

```
1 public class ImagePathConverter : IValueConverter
2 {
3     public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
4     {
5         if (value is string imagePath && int.TryParse(parameter?.ToString(), out int size))
6         {
7             // 在这里根据路径加载图片，并设置大小
8             BitmapImage bitmapImage = new BitmapImage(new Uri(imagePath));
9             bitmapImage.DecodePixelWidth = size;
10            return bitmapImage;
11        }
12        return value;
13    }
14
15    public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
16    {
17        throw new NotImplementedException();
18    }
19 }
```

## 3. 高级使用技巧

### 3.1 多个参数

如果需要传递多个参数，可以使用逗号或其他符号分隔它们，然后在转换器中解析。

```
1 | <TextBlock Text="{Binding Path=SomeValue, Converter={StaticResource MyConverter}, ConverterParameter=Param1,Param2}" />
```

```
1 public class MyConverter : IValueConverter
2 {
3     public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
4     {
5         if (parameter is string parameters)
6         {
7             string[] paramArray = parameters.Split(',');
8             // 处理参数...
9         }
10        return value;
11    }
12
13    public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
14    {
15        throw new NotImplementedException();
16    }
17 }
```

### 3.2 动态参数

有时候，可能需要根据数据的不同动态地设置 **ConverterParameter**。可以通过使用 **{Binding}** 在运行时动态绑定参数。

```
1 | <TextBlock Text="{Binding Path=SomeValue, Converter={StaticResource MyConverter}, ConverterParameter={Binding Path=Parameter}}" />
```

在这种情况下，确保上下文有一个名为 **Parameter** 的属性。

```
1 public class MyViewModel : INotifyPropertyChanged
2 {
3     private string _parameter;
4     public string Parameter
5     {
6         get { return _parameter; }
7         set
8         {
9             if (_parameter != value)
10            {
11                _parameter = value;
12                OnPropertyChanged(nameof(Parameter));
13            }
14        }
15    }
16
17    // 实现 INotifyPropertyChanged 接口的代码...
18 }
```



运行效果：

2023-12-19 18:40



希望这个详细实例源代码能够帮助你更好地理解和使用 **ConverterParameter** 在 WPF 中的作用和应用。





源代码获取：公众号回复消息【code: 64076】

相关代码下载地址



**重要提示!**：取消关注公众号后将无法再启用回复功能，不支持解封!

**第一步:** 微信扫码关注公众号“架构师老卢”

**第二步:** 在公众号聊天框发送 **code: 64076**，如：  **code: 64076**   获取下载地址

**第三步:** 恭喜你，快去下载你想要的资源吧