



# Silverlight

## Converters

# Converter

Os conversores permitem interceptar e modificar o valor propagado da propriedade fonte para a propriedade alvo (e vice-versa). Estas intercapções são necessárias quando, por exemplo, a propriedade do tipo alvo não é compatível com a do tipo destino. Na prática, um conversor é uma classe que implementa a interface ***IValueConverter***.

# Converter

Esta interface introduz dois métodos: ***Convert*** e ***ConvertBack***. O primeiro é usado para influenciar o valor propagado da propriedade fonte para a propriedade alvo. O segundo é apenas utilizado quando estamos perante um *Binding* bidirecional e permite a adaptação do valor propagado da propriedade alvo para a propriedade fonte..

# Converter

Conversores de valor são extremamente útil no Data Binding do Silverlight. Você pode usá-los de várias maneiras:

- ✓ **Para formatar dados para uma representação string:** Por exemplo, você pode converter um número em uma string de moeda.
- ✓ **Para criar um tipo específico de objeto Silverlight:** Por exemplo, você pode ler um bloco de dados comum e criar um objeto BitmapImage que pode ser ligado a um elemento da imagem.
- ✓ **Para condicionalmente alterar uma propriedade em um elemento com base nos dados vinculados:** Por exemplo, você pode criar um conversor de valor que muda a cor de fundo de um elemento para destacar valores em uma faixa específica.

# Converter

## **Formatação de Strings com um conversor de valor.**

Conversor de valor é a ferramenta perfeita para formatar números que precisam ser exibido como texto. Por exemplo, considere a propriedade `UnitCost` da classe `Product`. É armazenado como um `double`; e quando é exibido em uma caixa de texto, você verá valores como 3,9900. Não só este formato de exibição mostrar mais casas decimais do que você provavelmente gostaria, mas também deixa de fora o símbolo da moeda. A representação mais intuitiva é o valor da moeda-formatado \$ 3,99.

# Converter

Para criar um conversor de valor, é necessário realizar três passos:

1. Crie uma classe que implementa `IValueConverter` (do namespace `System.Windows.Data`). Você coloca essa classe em seu projeto Silverlight, que é onde a conversão ocorre e, não no serviço web.
2. Implemente um método `Convert ()` que altera os dados de seu formato original para o seu formato de exibição.
3. Implemente um método `ConvertBack ()` que faz o inverso e altera um valor de formato de exibição para seu formato nativo.

A seguir você verá o código completo do conversor de valor que lida com valores de preços como a propriedade `Product.UnitCost`:

# Converter

```
namespace Capitulo16ProSilverlight3
{
    public class PriceConverter : IValueConverter
    {
        public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
        {
            double price = (double)value;
            return price.ToString("C",culture);
        }

        public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
        {
            string price = value.ToString();

            double result;

            if (Double.TryParse(price, NumberStyles.Any, culture, out result))
            {
                return result;
            }

            return value;
        }
    }
}
```

# Converter

Agora, você precisa criar uma instância da classe PriceConveter na coleção da sua página de recursos, como mostramos aqui:

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

# Converter

Depois, você pode apontar para ele na sua ligação usando uma referência StaticResource:

```
<TextBox Margin="5" Grid.Row="2" Grid.Column="1"  
    Text="{Binding UnitCost, Mode=TwoWay,  
        Converter={StaticResource PriceConverter}}">  
</TextBox>
```

Veja o exemplo completo no Visual Studio.