

# Data Binding in WPF

---

# Conversie van Data

## Overzicht

---

- Gebruik een conversie klasse om data om te zetten die anders moet worden getoond in UI
- bv
  - Formatteringen, vertalingen
  - incompatibele data types tussen target en source

# Conversie van Data

## ingebouwde Converters

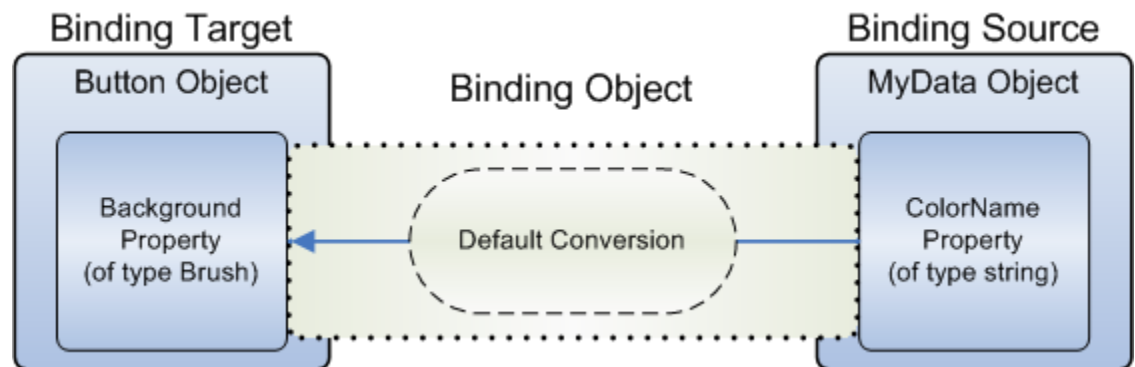
---

- In System.Windows.Controls namespace
- **BooleanToVisibilityConverter**
  - Converteert True/False naar Visible, Hidden of Collapsed
- **BorderGapMaskConverter**
  - Converteert dimensies van GroupBox control in een VisualBrush
- **MenuScrollingVisibilityConverter**
  - Regelt visibility van repeat buttons in scrolling menu's

# Conversie van Data

## Hoe werkt het

- **IValueConverter**
  - Convert
  - ConvertBack
- **IMultiValueConverter**
  - Convert
  - ConvertBack



# Conversie van Data

## IValueConverter voorbeeld

```
//[ValueConversion(source type, target type)]  
[ValueConversion(typeof(DateTime), typeof(String))]  
public class DateConverter : IValueConverter  
{  
  
    public object Convert(object value, Type targetType,  
        object parameter,  
        System.Globalization.CultureInfo culture)  
    {  
        string returnValue = string.Empty;  
        ...  
        return returnValue;  
    }  
  
    public object ConvertBack(object value, Type targetType,  
        object parameter,  
        System.Globalization.CultureInfo culture)  
    {  
        DateTime resultDateTime;  
        ...  
        return resultDateTime;  
    }  
}
```

# Conversie van Data

## IMultiValueConverter voorbeeld

```
public class NameConverter : IMultiValueConverter
{
    public object Convert(object[] values, Type targetType,
        object parameter, CultureInfo culture)
    {
        string name;
        switch ((string)parameter)
        {
            case "FormatLastFirst":
                name = values[1] + ", " + values[0];
                break;
            ...
        }
        return name;
    }
    public object[] ConvertBack(object value, Type[] targetTypes,
        object parameter, CultureInfo culture)
    {
        string[] splitValues = ((string)value).Split(' ');
        return splitValues;
    }
}
```

*demo*

---

# Data Validatie

## Overview

---

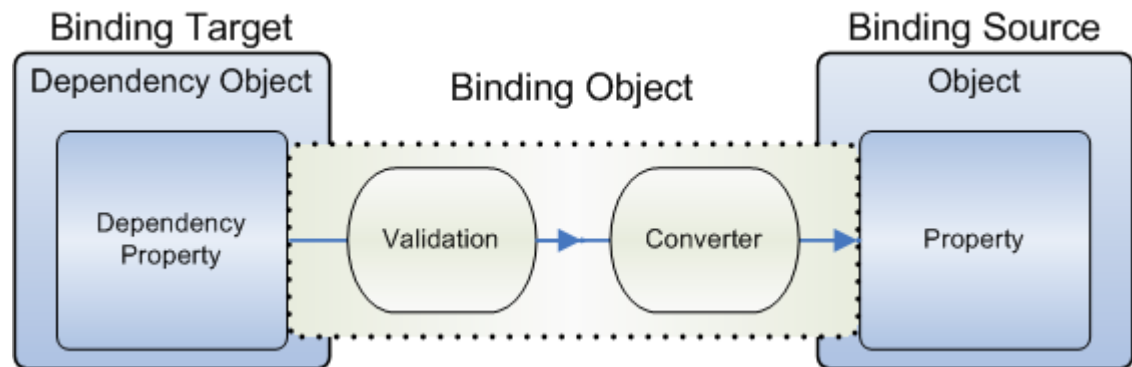
- Gebaseerd op ValidationRule
- Meerdere ValidationRules op één control mogelijk
- ErrorTemplate property en een ControlTemplate voor visuele voorstelling



# Data Validatie

## Process

- Wordt aangeroepen voor de converter
- Enkel voor 2-Way en OneWayToSource binding modes
- UpdateSourceTrigger-afhankelijk
  - LostFocus
  - PropertyChanged
  - Explicit



# Data Validatie

## ValidationRule

---

- Abstracte Klasse
- Override Validate methode
- Geeft een ValidationResult object terug

```
public abstract class ValidationRule
{
    // Methods
    protected ValidationRule();
    public abstract ValidationResult validate(object value,
                                           CultureInfo cultureInfo);
}
public class ValidationResult
{
    static ValidationResult();
    public ValidationResult(bool isValid, object errorContent);

    public object ErrorContent { get; }
    public bool IsValid { get; }
    public static ValidationResult validResult { get; }
}
```

# Enkele boeken

