

Extend `IValueConverter` to create your own converter



```
//File:Window.xaml.cs
```

```
using System;  
using System.Collections.Generic;  
using System.Text;
```

```
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using System.ComponentModel;
using System.Collections.ObjectModel;

namespace WpfApplication1 {

public class Employee : INotifyPropertyChanged {
public event PropertyChangedEventHandler PropertyChanged;
protected void Notify(string propName) {
if( this.PropertyChanged != null ) {
PropertyChanged(this, new PropertyChangedEventArgs(propName));
}
}

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

int age;
public int Age {
get { return this.age; }
set {
if( this.age == value ) { return; }
this.age = value;
Notify("Age");
}
}
```

```

}
}

public Employee() { }
public Employee(string name, int age) {
this.name = name;
this.age = age;
}
}

class People : ObservableCollection { }

public partial class Window1 : System.Windows.Window {
public Window1() {
InitializeComponent();
}

}

public class AgeToRangeConverter : IValueConverter {

public object Convert(object value, Type targetType, object
parameter, System.Globalization.CultureInfo culture) {
return (int)value < 25 ? "Under 25" : "Over 25"; } public
object ConvertBack(object value, Type targetType, object
parameter, System.Globalization.CultureInfo culture) { throw
new NotImplementedException(); } } } [/csharp]

```