

فرض کنید قصد دارید برای انتخاب بین چند گزینه‌ی محدود، از RadioButton ها بجای سایر کنترل‌های موجود استفاده کنید. این گزینه‌ها نیز توسط یک Enum تعریف شده‌اند. اکنون نیاز است گزینه‌های مختلف این Enum را به RadioButton های تعریف شده Bind کنیم.

تعریف Enum برنامه به صورت زیر است:

```
namespace WpfBindRadioButtonToEnum.Models
{
    public enum Gender
    {
        Female,
        Male
    }
}
```

در ادامه با توجه به اینکه RadioButton ها با خاصیت IsChecked از نوع bool کار می‌کنند، نیاز است بتوانیم گزینه‌های Enum را به bool و یا برعکس [تبدیل کنیم](#) . برای این منظور از تبدیگر EnumBooleanConverter ذیل می‌توان استفاده کرد:

```
using System;
using System.Globalization;
using System.Windows;
using System.Windows.Data;

namespace WpfBindRadioButtonToEnum.Converters
{
    public class EnumBooleanConverter : IValueConverter
    {
        public object Convert(object value, Type targetType, object parameter, CultureInfo culture)
        {
            if (Enum.IsDefined(value.GetType(), value) == false)
                return DependencyProperty.UnsetValue;

            return Enum.Parse(value.GetType(), parameter.ToString()).Equals(value);
        }

        public object ConvertBack(object value, Type targetType, object parameter, CultureInfo culture)
        {
            return Enum.Parse(targetType, parameter.ToString());
        }
    }
}
```

پیش‌فرض تبدیگر تهیه شده بر این است که مقدار ثابت Enum را از طریق سومین پارامتر، یعنی ConverterParameter تنظیم شده در حین عملیات Binding، دریافت می‌کند. پارامتر value مقداری است که از طریق Binding خاصیت IsChecked دریافت خواهد شد.

اکنون اگر ViewModel برنامه به شکل زیر باشد که GenderValue را در اختیار View قرار می‌دهد:

```
using System.ComponentModel;
using WpfBindRadioButtonToEnum.Models;

namespace WpfBindRadioButtonToEnum.ViewModels
{
    public class MainWindowViewModel : INotifyPropertyChanged
    {
        Gender _genderValue;
        public Gender GenderValue
        {
            get { return _genderValue; }
            set
            {
                _genderValue = value;
            }
        }
    }
}
```

```

        notifyPropertyChanged("GenderValue");
    }
}

#region INotifyPropertyChanged Members
public event PropertyChangedEventHandler PropertyChanged;
private void notifyPropertyChanged(string propertyName)
{
    if (PropertyChanged != null)
    {
        PropertyChanged(this, new PropertyChangedEventArgs(propertyName));
    }
}
#endregion
}
}

```

View متناظری که از آن و همچنین Enum و تبدیگر تهیه شده استفاده می‌کند، به شرح ذیل خواهد بود:

```

<Window x:Class="WpfBindRadioButtonToEnum.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:VM="clr-namespace:WpfBindRadioButtonToEnum.ViewModels"
        xmlns:C="clr-namespace:WpfBindRadioButtonToEnum.Converters"
        xmlns:Models="clr-namespace:WpfBindRadioButtonToEnum.Models"
        Title="MainWindow" Height="350" Width="525">
    <Window.Resources>
        <VM:MainWindowViewModel x:Key="VMainWindowViewModel" />
        <C:EnumBooleanConverter x:Key="CEnumBooleanConverter" />
    </Window.Resources>
    <StackPanel DataContext="{Binding Source={StaticResource VMainWindowViewModel}}">
        <TextBlock Text="Gender" Margin="3" />
        <RadioButton Content="{x:Static Models:Gender.Male}"
                    IsChecked="{Binding GenderValue, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged,
                    Converter={StaticResource CEnumBooleanConverter},
                    ConverterParameter={x:Static Models:Gender.Male}}"
                    Margin="3" GroupName="G1" />
        <RadioButton Content="{x:Static Models:Gender.Female}"
                    IsChecked="{Binding GenderValue, Mode=TwoWay, UpdateSourceTrigger=PropertyChanged,
                    Converter={StaticResource CEnumBooleanConverter},
                    ConverterParameter={x:Static Models:Gender.Female}}"
                    Margin="3" GroupName="G1" />
    </StackPanel>
</Window>

```

در این View از یک markup extension به نام [x:Static](#) برای دسترسی به فیلدهای ثابت برنامه کمک گرفته شده‌است. از [x:Static](#) در ConverterParameter و همچنین Content می‌توان استفاده کرد. برای دسترسی به Enum تعریف شده در برنامه، فضای نام آن توسط [xmlns:Models](#) در ابتدای کار تعریف گردیده‌است. در اینجا EnumBooleanConverter تهیه شده، کار تبدیل مقدار true و false دریافتی از IsChecked را به معادل Enum آن و برعکس، انجام می‌دهد.

به صورت خلاصه: ابتدا تبدیگر EnumBooleanConverter باید اضافه شود. سپس به ازای هر گزینه‌ی Enum، یک RadioButton در صفحه قرار می‌گیرد که ConverterParameter خاصیت IsChecked آن مساوی است با یکی از گزینه‌های Enum متناظر.