

# WPF Create Custom Checkbox List With ListBox Element

In order to use **CheckBox ListBox** in a WPF application, we should use custom **ListBox** list.

There are three steps to accomplish this:

1. Create a class which will represent items of the **ListBox**.
2. Create a **List** object consisting of the custom item class defined in step 1.
3. Add **ListBox.ItemTemplate** into **<ListBox>** element to create **CheckBox ListBox**.
4. Create a custom **MultiValueConverter** if you want to initialize the **CheckBox** items to be selected or not.

## Step 1: Create Country class for ListBox items

```
1 internal class Country
2 {
3     public string Name { get; set; }
4     public bool Active { get; set; }
5 }
```

## Step 2: Initialize ListBox

```
1 private static void initialize()
2 {
3     List<Country> countryList = new List<Country>();
4     Country country = new Country();
5     country.Name = "Turkey";
6     country.Active = true;
7     Country countryUS = new Country();
8     countryUS.Name = "America";
9     countryUS.Active = false;
10    countryList.Add(country);
11    countryList.Add(countryUS);
12    //countryTr is active, so add this object to SelectedItems collection
13    lstCountries.SelectedItems.Add(country);
14    lstCountries.ItemsSource = countryList;
15 }
```

## Step 3: Custom ListBox element in XAML file

```
1 <UserControl.Resources>
2     <converters:CollectionToBoolConverter x:Key="CollectionToBoolConverter" />
3 </UserControl.Resources>
4 <ListBox Grid.Row="2" SelectionMode="Multiple"
5         x:Name="lstCountries" Grid.Column="1" Margin="5" Width="600"
6         ScrollViewer.HorizontalScrollBarVisibility="Disabled">
7     <ListBox.ItemsPanel>
8         <ItemsPanelTemplate>
9             <WrapPanel Orientation="Horizontal" />
10            </ItemsPanelTemplate>
11        </ListBox.ItemsPanel>
12        <ListBox.ItemTemplate>
13            <DataTemplate>
14                <CheckBox Content="{Binding Name}"
15                    Margin="5 5 0 0" Width="80">
16                    <CheckBox.IsChecked>
17                        <MultiBinding Converter="{StaticResource CollectionToBoolConverter}">
18                            Mode="TwoWay">
19                                <Binding Path="IsSelected" RelativeSource="{RelativeSource
20 AncestorType=ListBoxItem}" />
21                                <Binding Path="Active" />
22                </CheckBox.IsChecked>
23            </DataTemplate>
24        </ListBox.ItemTemplate>
25    </ListBox>
26 </UserControl>
```

```

22         </MultiBinding>
23             </CheckBox.IsChecked>
24         </CheckBox>
25     </DataTemplate>
26 </ListBox.ItemTemplate>
27 </ListBox>

```

Line 4: We specified SelectionMode as Multiple, so we can choose multiple checkboxes.

Line 5, 6, 9: These lines are used to make **ListBox** style as WrapPanel.

Line 18: It is necessary to get selected checkboxes. To get SelectedItems: var `selectedCheckBoxList = 1stCountries.SelectedItems;`

Line 19: `Path="Active"` is necessary when initialize **CheckBox** items.

#### Step 4: CollectionToBoolConverter Class

```

1  public class CollectionToBoolConverter : IMultiValueConverter
2  {
3      /// <summary>
4      /// Runs when initialization
5      /// </summary>
6      /// <param name="values">Binding values</param>
7      /// <param name="targetType">Merged value</param>
8      /// <param name="parameter"></param>
9      /// <param name="culture"></param>
10     /// <returns></returns>
11     public object Convert(object[] values, Type targetType, object parameter,
12                           CultureInfo culture)
13     {
14         bool listBoxValue = (bool) values[0];
15         bool initializedValue = (bool) values[1];
16         return listBoxValue || initializedValue;
17     }
18     /// <summary>
19     /// Runs when any ListBox element thants CheckBox element is clicked.
20     /// This method only runs when Mode is "TwoWay"
21     /// </summary>
22     /// <param name="value"></param>
23     /// <param name="targetTypes"></param>
24     /// <param name="parameter"></param>
25     /// <param name="culture"></param>
26     /// <returns></returns>
27     public object[] ConvertBack(object value, Type[] targetTypes, object parameter,
28                                 CultureInfo culture)
29     {
30         /*bool isChecked = (bool) value;
31         object[] resultObjects=new object[2];
32         resultObjects[0] = isChecked;
33         resultObjects[1] = isChecked;
34         return resultObjects;
35         */
36         return null;
37     }
38 }

```