Border less and Draggable Window in WPF

http://www.c-sharpcorner.com/UploadFile/631fc0/border-less-and-draggable-window-in-wpf/

Introduction

In this article I am going to change a window appearance by creating a border less and Drag-gable window in WPF using the C# and XAML .The primary purpose of the windows is to host content that display information and enables user to interact with information .

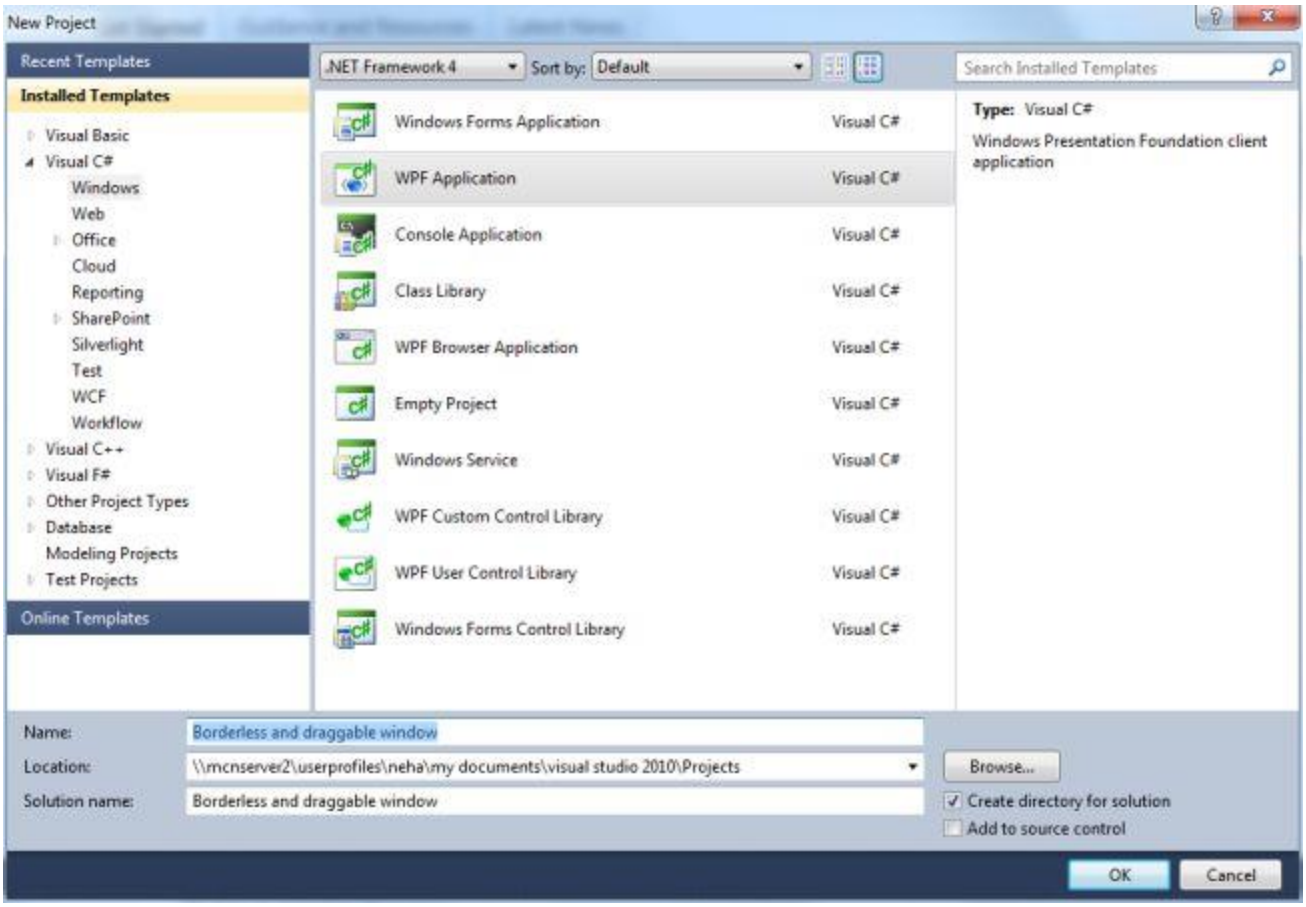
Here I use a handler called MouseDown event for calling a DragMove method to make window drag-gable.

```
private void rectangle2_MouseDown(object sender, MouseButtonEventArgs e)
{
   this.DragMove();
 }
```

Step 1

To create a WPF application using the following procedure :

- Open the Visual Studio.
- Select the C# language and "WPF" Application.
- Name the project as "Borderless and draggable window" .
- Click on the "OK" button .



Step 2

To get a border-less window ,you need to set the following attributes .

- WindowStyle = "None" Which make window border-less .
- AllowTransparency ="False".

Step 3

Now we start designing the window .

- Go to "view" -> "Toolbox".
- Drag and Drop the 2 rectangle on window named as "rectangle1" and "rectangle2"
- Now go to the MainWindow.xaml and write the following code between the rectangle1 tag.

```
   <Rectangle Height="323" HorizontalAlignment="Left" Name="rectangle1" Stroke="Black"
VerticalAlignment="Top" Width="510" RadiusX="9" RadiusY="9"
MouseDown="rectangle1_MouseDown">
        <Rectangle.OpacityMask>
          <LinearGradientBrush EndPoint="1,0.5" StartPoint="0,0.5">
            <GradientStop Color="Black" Offset="0" />
            <GradientStop Color="#CDFFFF00" Offset="1" />
          </LinearGradientBrush>
        </Rectangle.OpacityMask>
        <Rectangle.Fill>
          <LinearGradientBrush EndPoint="1,0.5" StartPoint="0,0.5">
            <GradientStop Color="Black" Offset="0" />
            <GradientStop Color="#FFD8D8B0" Offset="0.008" />
          </LinearGradientBrush>
        </Rectangle.Fill>
      </Rectangle>
```

- Similarly write the following code between the rectangle2 tag .

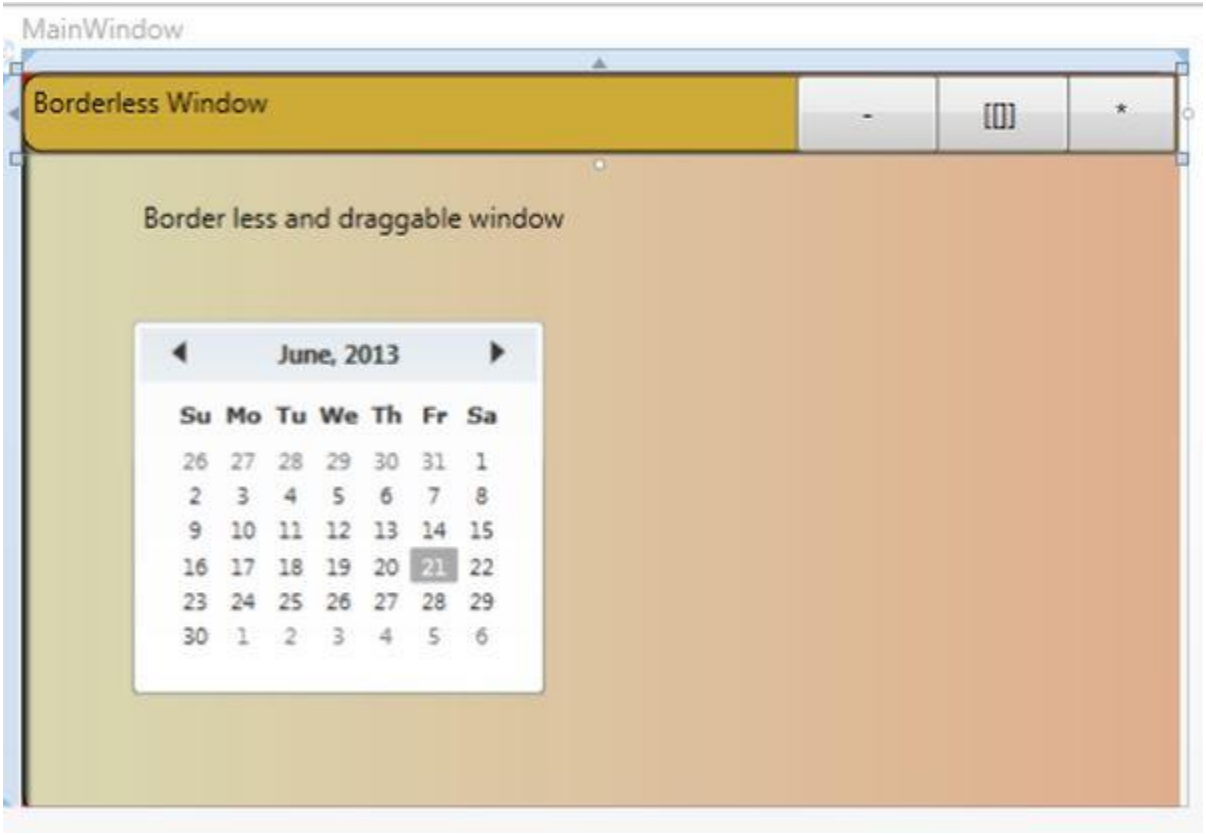
```
<Rectangle Height="35" HorizontalAlignment="Left" Name="rectangle2" Stroke="Black"
VerticalAlignment="Top" Width="503" Fill="#FFCEAB37" MouseDown="rectangle2_MouseDown"
RadiusX="9" RadiusY="9"></Rectangle>
```

- Then Drag and Drop some other control on the window such as calendar,label and 3 button . here is final source code of the MainWindow.xaml .

```
<Window x:Class="Dragable_window_in_wpf.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="361" Width="526" WindowStyle="None" AllowsTransparency="False"
Loaded="Window_Loaded" IsHitTestVisible="True" ResizeMode="NoResize" >
    <Window.Background>
        <SolidColorBrush />
    </Window.Background>
    <Grid Background="Red" Width="508" Height="319">
        <Rectangle Height="323" HorizontalAlignment="Left" Name="rectangle1" Stroke="Black"
VerticalAlignment="Top" Width="510" RadiusX="9" RadiusY="9"
MouseDown="rectangle1_MouseDown">
            <Rectangle.OpacityMask>
                <LinearGradientBrush EndPoint="1,0.5" StartPoint="0,0.5">
                    <GradientStop Color="Black" Offset="0" />
                    <GradientStop Color="#CDFFFF00" Offset="1" />
                </LinearGradientBrush>
            </Rectangle.OpacityMask>
            <Rectangle.Fill>
                <LinearGradientBrush EndPoint="1,0.5" StartPoint="0,0.5">
                    <GradientStop Color="Black" Offset="0" />
                    <GradientStop Color="#FFD8D8B0" Offset="0.008" />
                </LinearGradientBrush>
            </Rectangle.Fill>
        </Rectangle>
        <Rectangle Height="35" HorizontalAlignment="Left" Name="rectangle2" Stroke="Black"
VerticalAlignment="Top" Width="503" Fill="#FFCEAB37" MouseDown="rectangle2_MouseDown"
RadiusX="9" RadiusY="9"></Rectangle>
        <Calendar Height="170" HorizontalAlignment="Left" Margin="48,105,0,0" Name="calendar1"
VerticalAlignment="Top" Width="180" />
        <Label Content="Border less and draggable window" Height="28" HorizontalAlignment="Left"
Margin="48,50,0,0" Name="label1" VerticalAlignment="Top" Width="199" />
        <Button Content="*" Height="35" HorizontalAlignment="Left" Margin="454,0,0,0" Name="button1"
VerticalAlignment="Top" Width="49" Click="button1_Click" />
        <Button Content="[[]]" Height="35" HorizontalAlignment="Left" Margin="395,0,0,0"
Name="button2" VerticalAlignment="Top" Width="61" Click="button2_Click"  />
        <Button Content="-" Height="35" HorizontalAlignment="Left" Margin="337,0,0,0" Name="button3"
VerticalAlignment="Top" Width="62" Click="button3_Click" />
        <Label Content="Borderless Window" Height="35" HorizontalAlignment="Left" Name="label2"
VerticalAlignment="Top" Width="116" />
```

```
    </Grid>
</Window>
```

After writing the code the window looks as



Step 4

- Go to the MainWindows.xaml.cs and write the code given below .

```
using System;
using System.Collections.Generic;
using System.Linq;
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.Navigation;
using System.Windows.Shapes;

namespace Dragable_window_in_wpf
```

```
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public MainWindow()
        {
            InitializeComponent();
            SizeToContent = System.Windows.SizeToContent.Manual;
        }

        private void Window_Loaded(object sender, RoutedEventArgs e)
        {

        }


        private void rectangle2_MouseDown(object sender, MouseButtonEventArgs e)
        {
            this.DragMove();
        }
        private void rectangle1_MouseDown(object sender, MouseButtonEventArgs e)
        {
            this.DragMove();
        }

        private void button1_Click(object sender, RoutedEventArgs e)
        {
            this.Close();
        }

        private void button3_Click(object sender, RoutedEventArgs e)
        {
            this.WindowState = WindowState.Minimized;

        }

        private void button2_Click(object sender, RoutedEventArgs e)
        {
            this.WindowState = WindowState.Maximized;
        }

      }
         }
```

Output

Now press F5 :

When we click on "minimize [-] " icon it minimize the window ,similarly when click on the "cross [*]" icon it shutdown the window and if we click on the "maximize [[]] icon it maximize window which looks as.


Borderless Window
-
[[]]
*
Border less and draggable window
June, 2013
Su Mo Tu We Th Fr Sa
26 27 28 29 30 31 1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 1 2 3 4 5 6