# Create a WPF user control

### http://msdn.microsoft.com/en-us/library/cc294992.aspx

#### Blend 2

If you want to design a reusable component that can be added to the artboard just like a system control, you can create a user control in Microsoft Expression Blend. User controls can contain other controls, resources, and animation timelines, just like a Windows Presentation Foundation (WPF) application. The only difference is that the root element is a **UserControl** instead of a **Window** or a **Page**The following procedures show you how to create a user control that contains some animation, and instantiate it in another document. (For an example of creating a custom control in code that can be based on an existing system control, see <u>Try it: Create a custom WPF control.</u>)

You can find more examples of user controls in the samples that come with Expression Blend. In the **Help** menu, click **Welcome Screen**, select the **Samples**tab, and then click the name of a sample, such as **SmoothBlends**. For more information about user controls, including XAML and code examples, see the <u>Control Authoring Overview</u> in the <u>Windows Presentation Foundation</u> section on <u>MSDN</u>.



User controls are not supported in Microsoft Silverlight 1.0. However, you can create custom JavaScript class instantiate programmatically. For an example, see the Button class in the **ButtonGallery** sample that is availathe **Samples** tab in the **Welcome Screen** (on the **Help** menu).

User controls are supported in Silverlight 2. For more information, see Create a new user control in your Silverlight

# To define the user control

- 1. In Expression Blend, do one of the following:
  - To create a user control in a .dll file, click **New Project** on the **File** menu, select the **WPF Control Library** project type, give the project a name, select the language for the codebehind file of the main document, and then click **OK**. Use this option if you want to hide
    the implementation of your user control when you give it to someone else, or if you want
    to be able to change the appearance of the user control by creating a template when the
    user control is drawn in another application.
  - To create a user control in a .xaml file in an existing project, click **New Item** (CTRL+N) on the **File** menu, select the **UserControl** template, give the file a name, and then click **OK**. This option is easier to change because your user control is in the same project in which you use it. Therefore, you can skip the step of updating a reference to a .dll.

Expression Blend opens the user control for editing.

and then clicking the name of the panel.

- Decide what type of panel you want for the root element. By default,
   a Grid named LayoutRoot is used, which allows any animations to resize when the user control is
   drawn into another document. You could change this to a Canvas or other panel control by right clicking theLayoutRoot element under Objects and Timeline, pointing to Change Layout Type,
- 3. From the Toolbox, select whichever controls and drawing tools that you want in your user control and draw them on the artboard. Modify them using the properties in the **Properties** panel. User controls can contain anything that a WPF application can contain.

4. Under **Objects and Timeline**, create any animation timelines that you want. For an example, see <u>Create simple animation</u>.



When you set keyframes, consider the timing of all the animations in your application, and when the a control will run. For example, consider an application that animates a splash screen first and then anim fading into view second. You can put each animation into its own user control, but leave enough time second animation for the length of time that it takes for the splash screen animation to finish.

# **V**Tip:

Set a keyframe at the 0-second mark if you want the animation to be able to start over multiple times. create an animation timeline that moves an object from left to right and is triggered by a mouse double set a keyframe at the 0-second mark. The animation will not run more than once on subsequent mouse it is a handoff animation. For information about handoff animations, see "Using multiple and overlapp timelines" in the Animation overview.

5. Under **Triggers**, configure any property or event triggers that will make your application respond to user interactions. For an example, see <u>Add or remove a trigger</u>.



When you decide on the triggers that you want to set in your user control, consider all the properties at want to make available to your user control. For example, consider an application that contains a buttor that is contained in a user control. In Expression Blend, you cannot add a trigger to the user control that animation timeline when the button is clicked unless the button is also a part of the user control. Addit bind only between property values if both properties are in the same user control. You can work aroun programmatically in code-behind files, or by creating a template with triggers and animation timelines after it is added to a document. For an example of a user control with a code-behind file, see the Control Overview in the Windows Presentation Foundation section on MSDN. For information about how to make the control of the contro

- 6. When you finish creating your user control, make sure that you change the size of the root element of your document so that it is only as big as it has to be. Under **Objects and Timeline**, select your document root, and then, by using the **Selection** tool, adjust the size of your document window using the blue adorners on the artboard.
- 7. If your user control depends on mouse clicks or interaction with the empty area in the user control, you need to set the background of the root element to a solid brush so that your user control uses space on the artboard when it is added to another document. In the **Brushes** category of the **Properties** panel, change the **Background** property of your root element to a **Solid Color Brush**. If you want to keep the background invisible, change the Alpha subproperty to 0.

- 8. If your user control depends on a specific height or width, set the **MinHeight** and **MinWidth** properties in the **Advanced** section of the **Layout** category in the **Properties** panel.
- 9. If you want your user control to be able to be resized when it is drawn into a document, make sure that the **Width** and **Height** properties of all objects in the user control are reset to **Auto**.
- 10. Save your files and project by clicking **Save All** on the **File** menu.
- 11. If your project is a control library, build your project to create the .dll file by clicking **Build Project** (CTRL+SHIFT+B) on the **Project** menu. The .dll file is built and saved to the \bin\Debug folder in the same location as your project.

### ↑ Back to top

### To instantiate the user control in another document

- 1. If you created a **Control Library**, add a reference to the .dll in the project in which you want to use your user control. On the **Project** menu, click**Add Reference**, browse to the .dll file for your user control in the **Add Reference** window, and then click **Open**.
- 2. Build your current project (CTRL+SHIFT+B) to make the user control available for selection in the **Asset Library**.
- 3. From the Custom Controls tab of the **Asset Library** , select your user control. The icon for your user control appears selected above the **Asset Library** button.
- 4. Use your mouse to draw your user control on the artboard.
- 5. Test your project (F5) to see your user control in action.

### ↑ Back to top

# To edit a user control after it is added to another document

If you created a user control document instead of a control library (.dll), you can enter the editing mode for your control using the shortcut menu.

- 1. Right-click the user control on the artboard or under **Objects and Timeline**, and then click **Edit Control**. Expression Blend opens the user control document for editing.
- 2. After you make your changes, save your file, and then switch back to the main document. If you created a user control in a control library project, you need to edit the original control library project, and then rebuild the .dll.

### User Control in XAML and WPF

http://www.c-sharpcorner.com/uploadfile/mahesh/user-control-in-wpf/

This tutorial shows how to create a User Control in XAML and WPF using Visual Studioand how to use it in a WPF application. The first part of this article creates a Login user control and second part shows how to use it in an application.

### **The Login Control**

Create a XAML application using Visual Studio and add a new item by right clicking on the project, select Add >> New Item from the menu and select User Control from the available

templates. See in Figure 1. Before clicking the Add button, change the name of XAML file in the below text box to the name of your control. I change my file name to LoginControl.

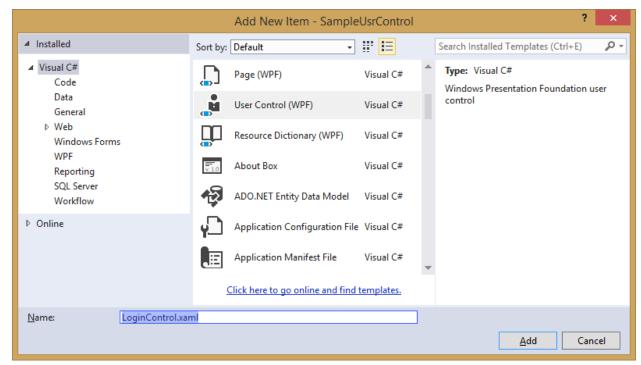


Figure 1. Adding a User Control (WPF)

This action adds a LoginControl.xaml file to your project. The

```
<UserControl
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    x:Class="XamlFileBrowser.UserControl1"
    x:Name="UserControl1"
    d:DesignWidth="397" d:DesignHeight="39">
        <Grid x:Name="LayoutRoot">
        </Grid>
</UserControl>
```

### **Designing the LoginControl UserControl**

Now I add a button and a TextBox control to the user control and change names of the TextBox and the Button controls to IblUserName and txtUserName respectively. The design of the control looks like Figure 2.



Figure 2. LoginControl UserControl

As you can see from the below code, I also add a Button click event handler and TextBox's text changed event handler.

```
<TextBox x:Name="FBCTextBox" Margin="4,10.313,137,4.001" Text="TextBox"TextWr apping="Wrap" Grid.ColumnSpan="2" TextChanged="FBCTextBox_TextChanged" /> <Button x:Name="FBCButton" HorizontalAlignment="Right" Margin="0,8,13,4"Width = "111" Content="Browse" Grid.Column="1" Click="FBCButton_Click" />
```

I also change the x:Class and x:Name values of the user control and ad some formatting to the Grid. The final code looks like following:

```
<UserControl
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    x:Class="McXamlLib.XamlFileBrowser"
    x:Name="XAMLFileBrowserControl"
    d:DesignWidth="397" d:DesignHeight="39">

<pre
```

### Adding code to the XamlFileBrowser Control

Now, we are going to add the code to the control.

First, I create a public property called FileName and sets and gets text of the FBCTextBox control. The property code looks like following:

```
public string FileName
{
         get { return FBCTextBox.Text; }
         set { FBCTextBox.Text = value; }
}
```

Now, on the FBCButton click event handler, I write code to browse files on the system using the OpenFileDialog control and set FileName property to the selected file name in the OpenFileDialog.

In the end, I add an event called FileNameChanged and call it on the textchanged event of the TextBox as following:

That's it. Our control is ready.

### The XAMLFileBrowser Control Host Application

Using a user control in XAML is pretty simple. I use a WPF Application to test the control.

Create a WPF application project and copy the control code files to your project.

After that, you need to add namespace of the library in which the user control is defined. In our case, the library was McXamlLib. Add the following namespace within the Window or Page tag of your application.

```
xmlns:local="clr-namespace:McXamlLib"
```

After that, create the control using the following syntax:

The complete code for the host application looks like following:

Now if you run the application, the output looks like Figure 3. If you click on the Browse control, it opens the OpenFileDialog control and let you select a file. The selected file name is added as text of the TextBox.



Figure 3. The XamlFileBrowser control in action

### **Forthcoming**

Using the same approach, I am creating an interactive XamlFileBrowser control that will have styles and interactive UI as well as new features such as MRUs. After that, I will use this control in one of my Graphics Designer applications I am working on. Stay tuned. As always, all comments and critics are most welcome. Please post them at the bottom of this article using Post Comment button.

### Summary

In this article, we saw how easy it is to create a user control in XAML and WPF application programming model. First, we created a user control and later we created a host application to test our user control