Create a WPF user control

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 Try it: Create a custom WPF control.)

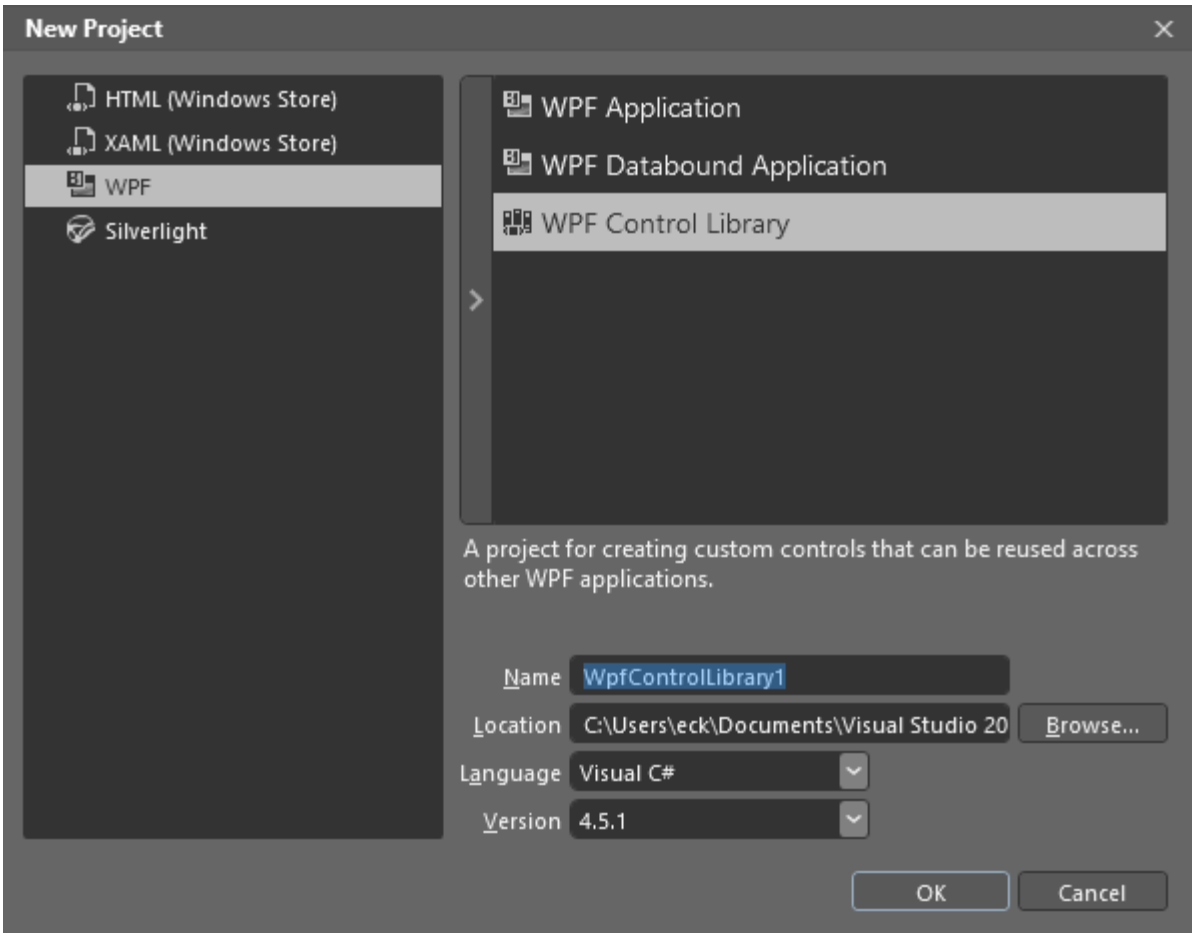
You can find more examples of user controls in the samples that come with Expression Blend. In the **Help** menu, click **Welcome Screen**, select the **Samples**tab, and then click the name of a sample, such as **SmoothBlends**. For more information about user controls, including XAML and code examples,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

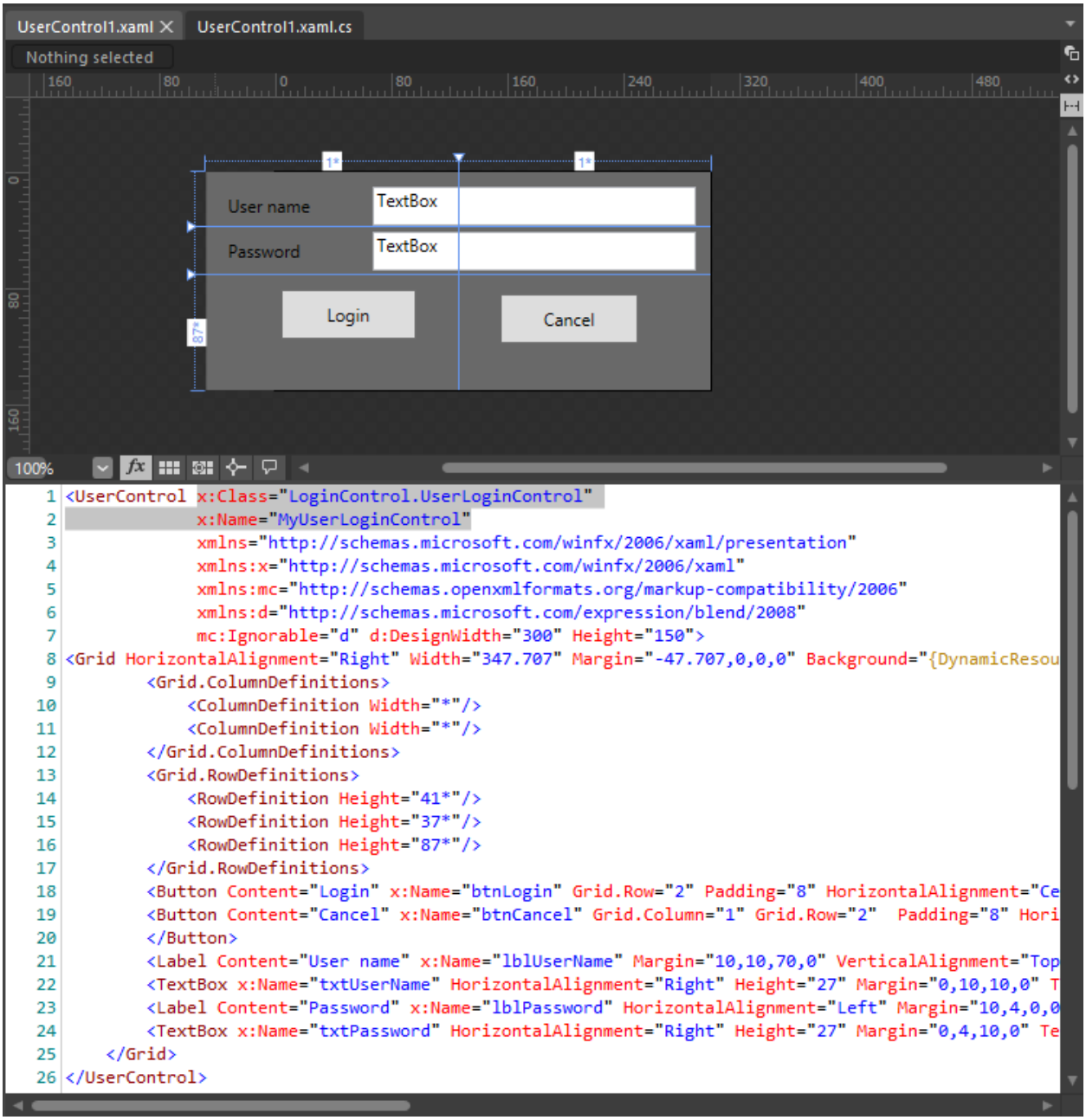
To define the user control

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 code-behind 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.


New Project
HTML (Windows Store)
XAML (Windows Store)
WPF
Silverlight
WPF Application
WPF Databound Application
WPF Control Library
A project for creating custom controls that can be reused across other WPF applications.
Name
WpfControlLibrary1
Location
C:\Users\eck\Documents\Visual Studio 20
Browse...
Language
Visual C#
Version
4.5.1
OK
Cancel




```
<UserControl x:Class="LoginControl.UserLoginControl"
             x:Name="MyUserLoginControl"
             xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
             xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
             xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
             mc:Ignorable="d" d:DesignWidth="300" Height="150">
<Grid HorizontalAlignment="Right" Width="347.707" Margin="-47.707,0,0,0" Background="{DynamicResou
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="*"/>
            <ColumnDefinition Width="*"/>
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="41*"/>
            <RowDefinition Height="37*"/>
            <RowDefinition Height="87*"/>
        </Grid.RowDefinitions>
        <Button Content="Login" x:Name="btnLogin" Grid.Row="2" Padding="8" HorizontalAlignment="Ce
        <Button Content="Cancel" x:Name="btnCancel" Grid.Column="1" Grid.Row="2"  Padding="8" Hori
        </Button>
        <Label Content="User name" x:Name="lblUserName" Margin="10,10,70,0" VerticalAlignment="Top
        <TextBox x:Name="txtUserName" HorizontalAlignment="Right" Height="27" Margin="0,10,10,0" T
        <Label Content="Password" x:Name="lblPassword" HorizontalAlignment="Left" Margin="10,4,0,0
        <TextBox x:Name="txtPassword" HorizontalAlignment="Right" Height="27" Margin="0,4,10,0" Te
    </Grid>
</UserControl>
```

2. Modify the Class and the Name attributes.

```
x:Class="LoginControl.UserLoginControl"
x:Name="MyUserLoginControl"
```

where for instance `LoginControl` is the namespace of the UserControl and `UserLoginControl` is the name of the usercontrol class `name` in that namespace. The Name attribute is used to refer to this control in code of the host application

3. 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 the**LayoutRoot** element under **Objects and Timeline**, pointing to **Change Layout Type**, and then clicking the name of the panel.

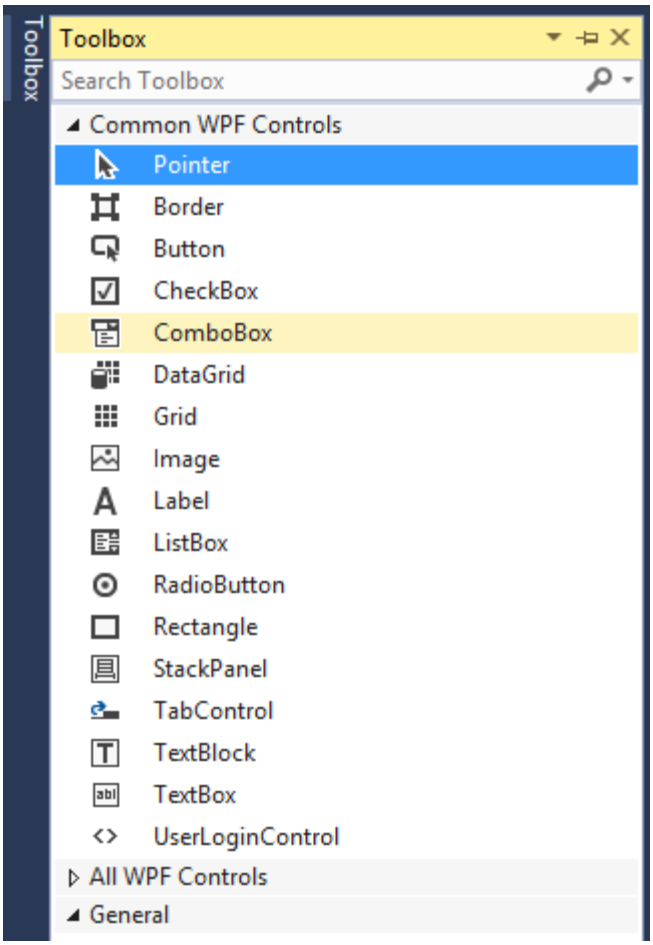
4. From the Toolbox, select whichever controls and drawing tools that you want in your user control and draw them on the artboard. Modify them using the properties in the **Properties** panel. User controls can contain anything that a WPF application can contain.
5. Under **Triggers**, configure any property or event triggers that will make your application respond to user interactions. For an example, see Add or remove a trigger.
6. 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.
7. 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**.
8. Save your files and project by clicking **Save All** on the **File** menu.
9. 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.

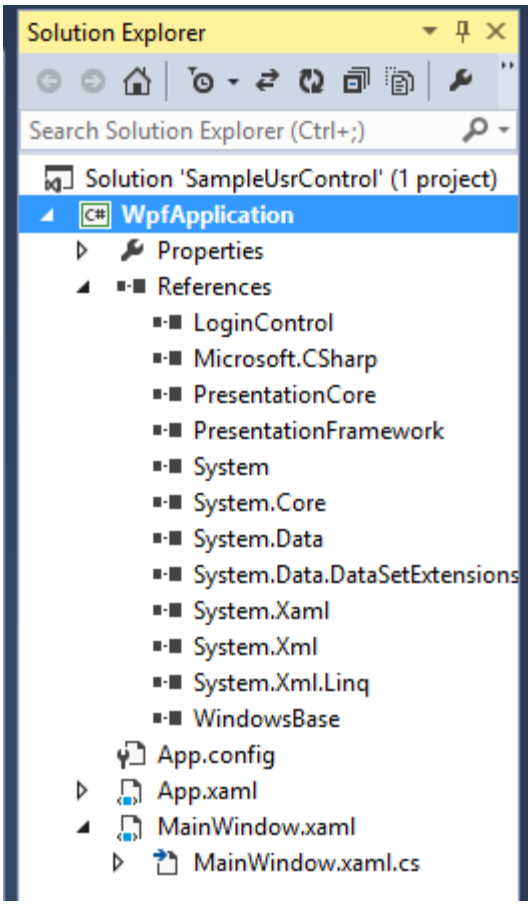
To instantiate the user control in another document

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.

To instantiate the user control in Visual Studio

1. Create a WPF application. Right click ion the Toolbox and select Choose Items. Browse to select the DLL file of the UserControl and you will get it among the rest of the controls in the Toolbox.

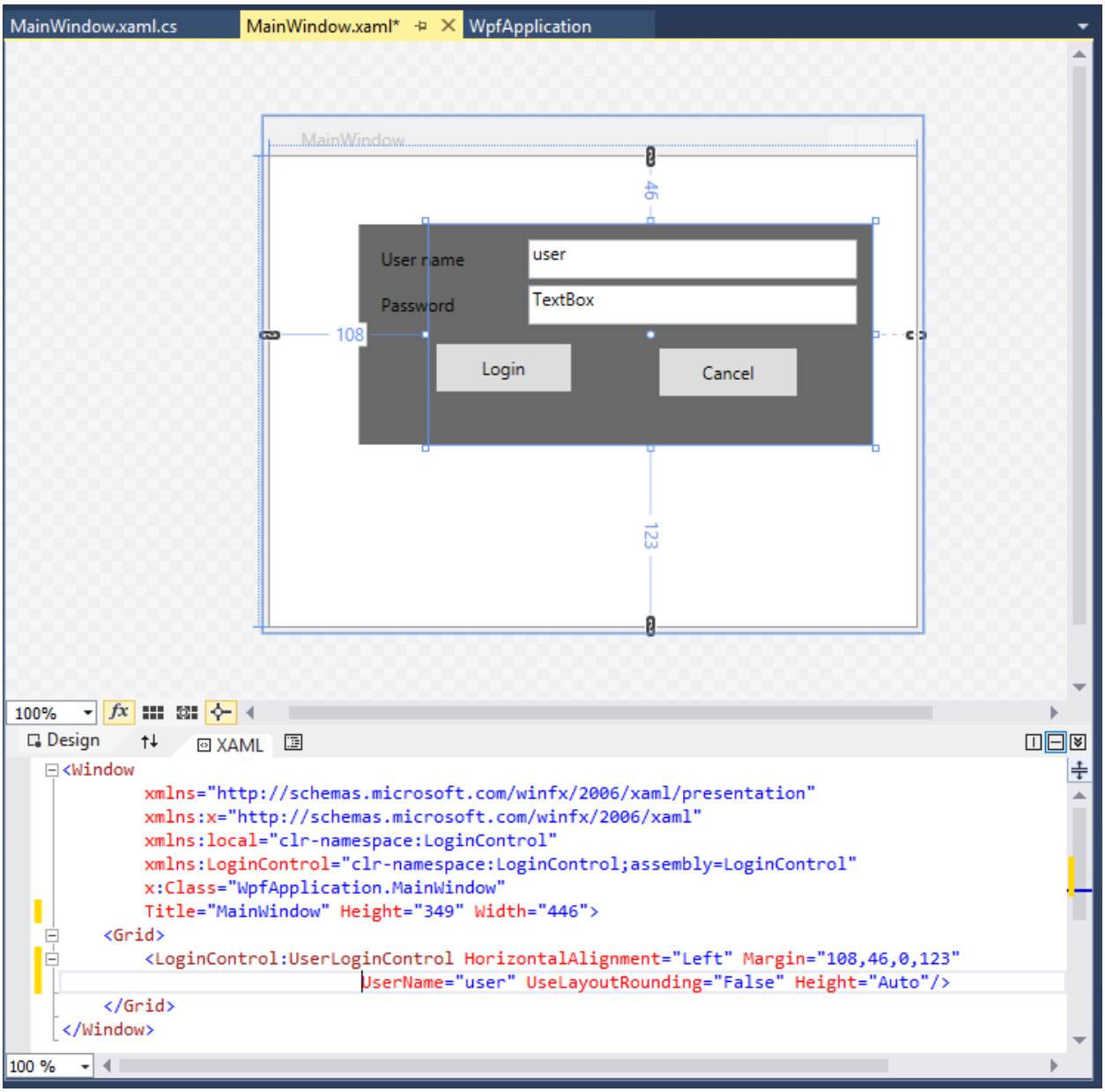

Toolbox
Toolbox
Search Toolbox
Common WPF Controls
Pointer
Border
Button
CheckBox
ComboBox
DataGrid
Grid
Image
Label
ListBox
RadioButton
Rectangle
StackPanel
TabControl
TextBlock
TextBox
UserLoginControl
All WPF Controls
General



2. Drag and drop the usercontrol in the WPF design area.
3. Make sure the Window root attribute uses the namespace of the use control. For instance, there should be a reference

   ```
   xmlns: ="clr-namespace:LoginControl;assembly=LoginControl"
   ```

   while the `xmln` and `assembly` values match the namespace (`LoginControl`) of the usercontrol


MainWindow.xaml.cs
MainWindow.xaml*
WpfApplication
MainWindow
46
User name
user
Password
TextBox
108
Login
Cancel
123
100%
Design
XAML
<Window
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:local="clr-namespace:LoginControl"
xmlns:LoginControl="clr-namespace:LoginControl;assembly=LoginControl"
x:Class="WpfApplication.MainWindow"
Title="MainWindow" Height="349" Width="446">
<Grid>
<LoginControl:UserLoginControl HorizontalAlignment="Left" Margin="108,46,0,123"
UserName="user" UseLayoutRounding="False" Height="Auto"/>
</Grid>
</Window>
100 %