

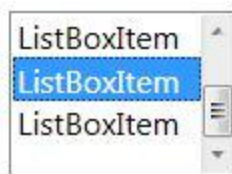
WPF tutorial: HOW TO style and change WPF ListBox (ComboBox, TreeView...) highlighting colors



How to change color of selected (highlighted) item in your WPF ListBox (or ComboBox, ListView or TreeView) in Expression Blend? WPF allows you to do this and much more by using Styles and Templates. In this tutorial I will show you how to achieve that. **(added source code)**

Often, you need to restyle default style of your WPF controls. In this tutorial, I will show you how to change highlighting color of your WPF List Box in Expression Blend. That way you can customize your controls and provide better user experience.

Basically we will go from this:

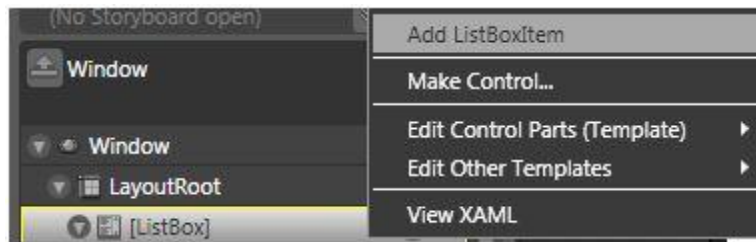


...to something like this:

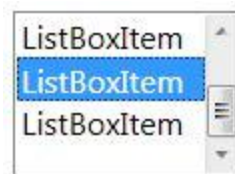


ListBox (and everything I will show you in this tutorial is applicable to ComboBox, TreeView and ListView WPF controls) is a common control used widely across all different kinds of applications. Customization is, however, rarely seen and I hope to show that changing something like highlight color of your ListBox control is very simple.

1. Start your Expression Blend and draw ListBox control on workspace.
2. Right click on your ListBox in **Objects and Timeline** pane and click on **Add ListBoxItem**. Repeat that procedure several times – just enough to populate your ListBox with 5 or 6 items.



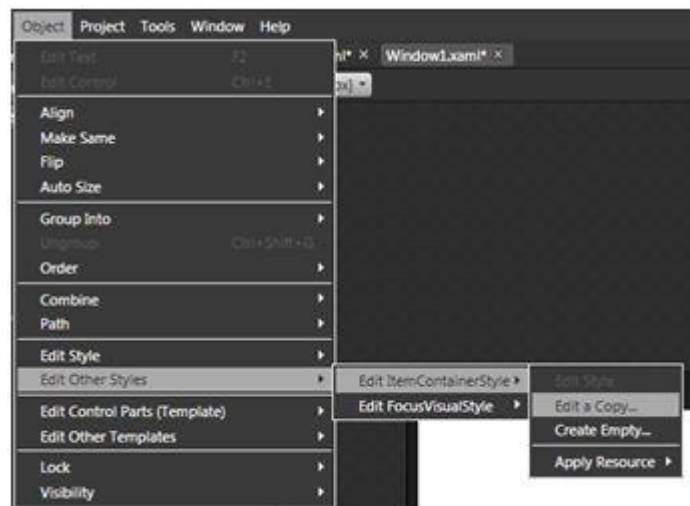
3. Press **F5** to start your project. Click on some item in your list box and it should look something like this:



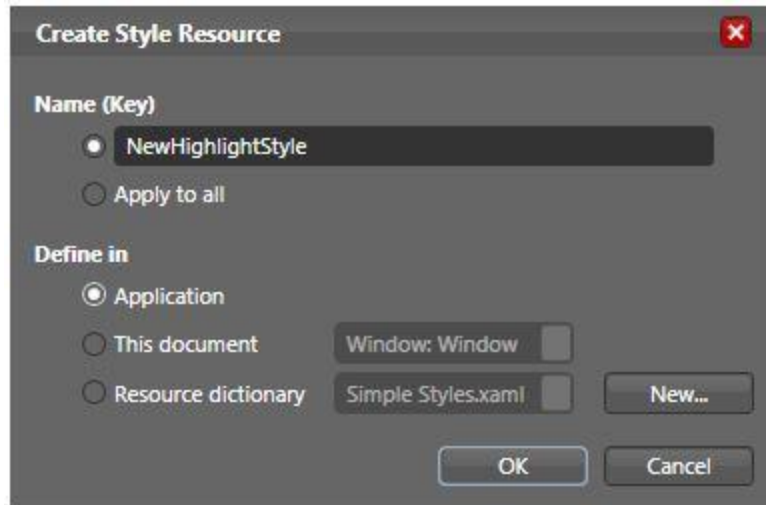
That is the default highlight for selected item in your ListBox control. And we are going to change it right now.

4. Close your window and return to Blend.

5. Select your ListBox and from **Object** menu choose **Edit Other Styles** then **Edit ItemContainerStyle** and finally Edit Copy...

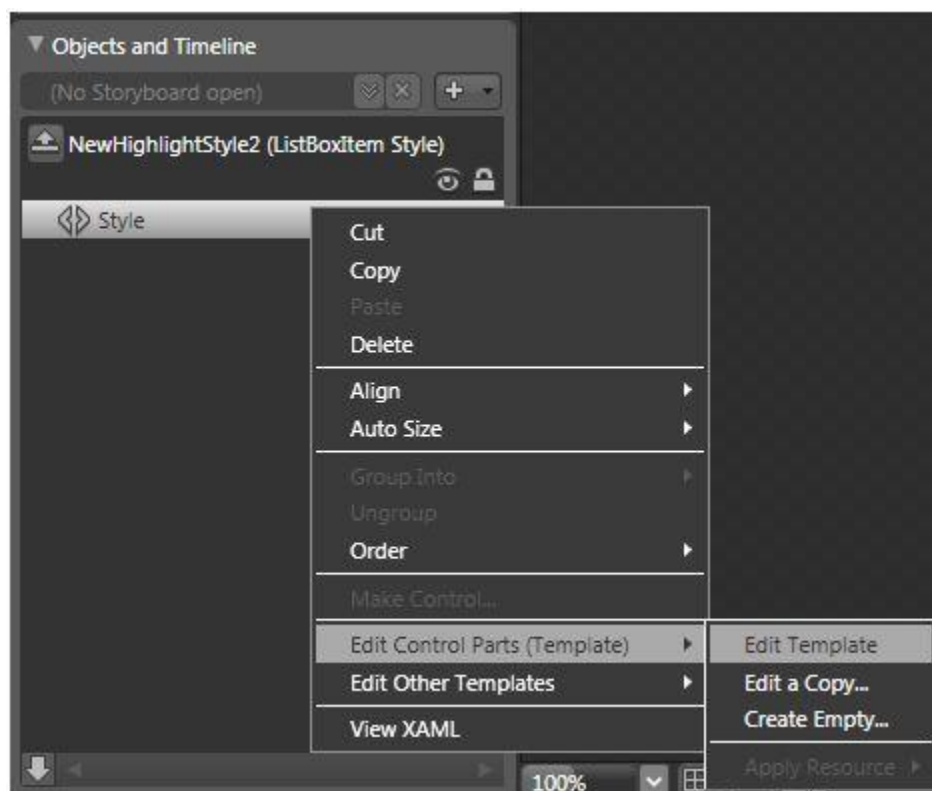


6. Create Style Resource dialog shows. Enter name for your style – for example – **NewHighlightStyle** and select **Define in Application** (this basically just tells if this style will be able to all objects in application or this document only or in resource dictionary).

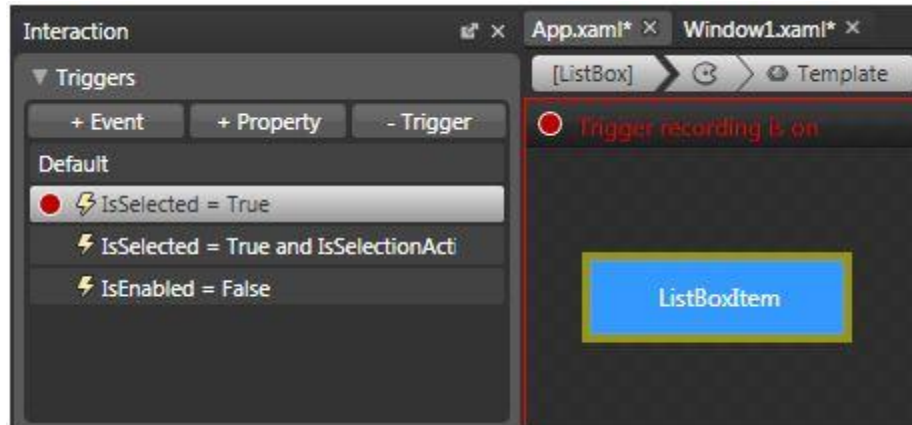


Click on **OK** when you are done.

7. Right-click on **Style** in **Objects and Timeline** pane and then select **Edit Control Parts (Template)** and **Edit Template** after that.

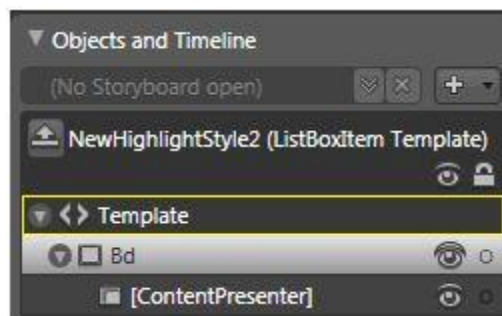


8. Now we are ready for some serious job. Take a look at **Trigger** section in **Interaction** pane. You can notice that there are several defined triggers and that they are connected with some specific actions.

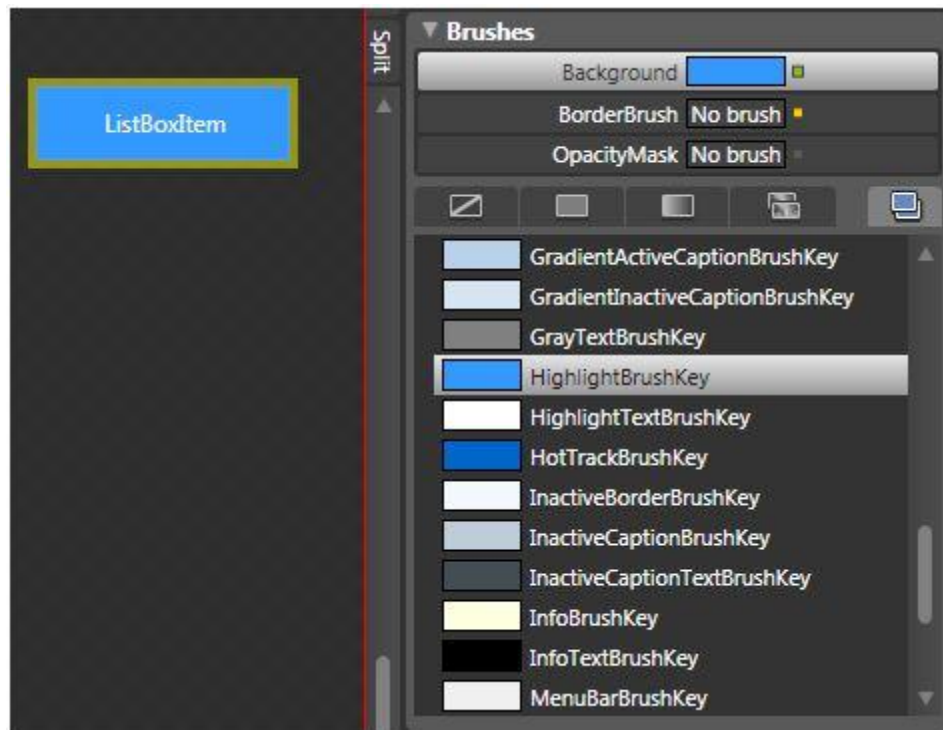


Click on **IsSelected = True** and you will notice that your ListBoxItem changes its color to blue – color that it usually has when it is highlighted.

8. Select the border control called **Bd** in your **Objects and Timeline** pane.

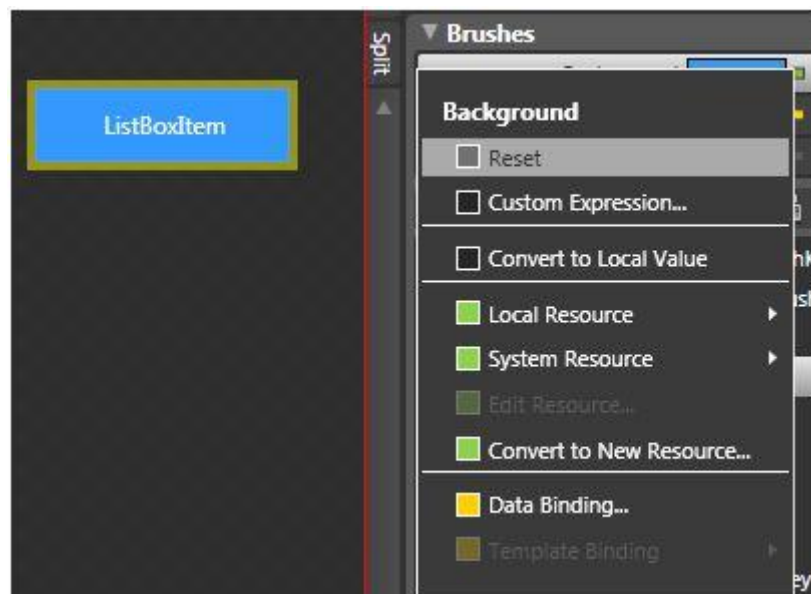


9. Take a look at **Properties** pane.



You can see under **Brushes** section that **Background** is set to **HighlightBrushKey** - which is one of predefined resources.

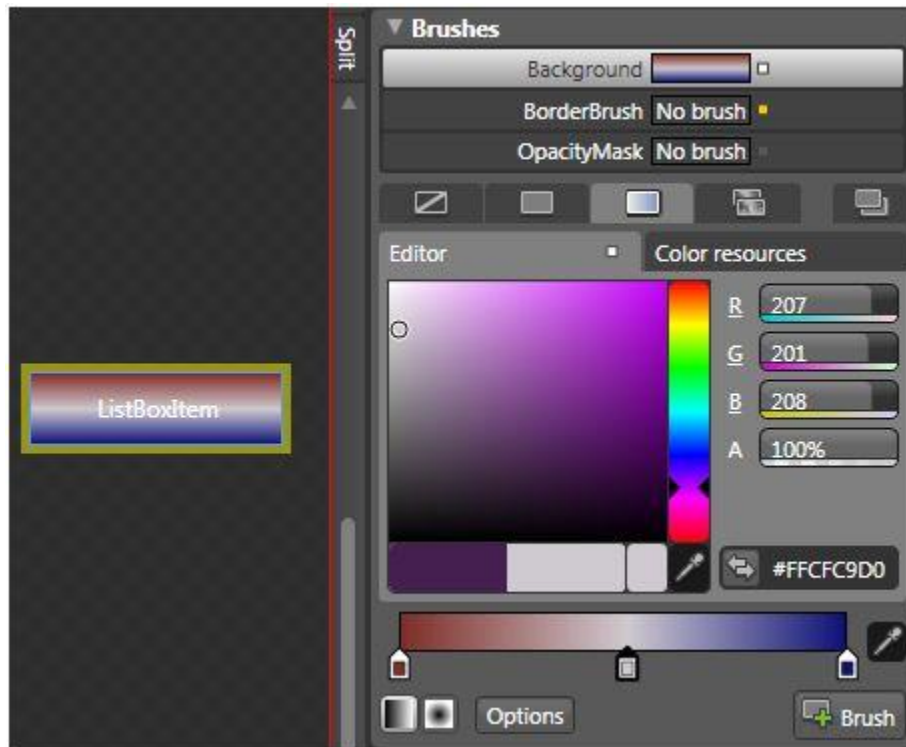
10. Click on **Advanced property options** (little green square) and after that click on **Reset**.



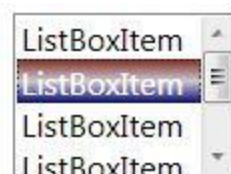
You will now notice that your `ListBoxItem` looks transparent.



11. Now we will add some colors and make that ListBoxItem look nicer. Click on **Gradient Brush**, add few stops and create whatever background you want. (You can tell I am from Croatia just by looking on my choice of colors) 😊



12. If you go now again and press **F5** to start your project and click on ListBoxItem you should see that it now sports highlight style that you've just defined.



In this way you can easily change look of *de facto* all WPF controls without worrying if you will change default behaviors.

In future tutorials on UXPassion.com I will go and show you numerous other possibilities for creating and customizing controls.

Please note that this process can be used for styling your Silverlight controls as well. If you have some questions, leave them in comments, I'd be glad to try to help you out.