

## PictureBox Opacity

You can find the completed version of this program on GitHub:

<https://github.com/AldworthClass/Opacity-Changer-VB>

This tutorial will cover how to change the **opacity** of an image in a PictureBox. You may use your own image files, or download the ones that will be used in the example program.

Create a new project. Give it an appropriate name.

Import *slimer.png* and *haunted\_background.png* into the project resources.

Set the **Background Image** property of your Form to the haunted\_background image from resources.

Add a PictureBox and:

- Give it an appropriate name (we will use `imgSlimer` for this tutorial)
- Set the image of this PictureBox to `slimer` from Resources
- Make the `Background_Color` Transparent

Add a Button and name it `btnFadeOut` with Text "Fade Out"

Add another Button named `btnFadeIn` with Text "Fade In"

Add another Button named `btnAppear` with Text "Appear"

In the code editor, at the very top, add the following import statement:

```
Imports System.Drawing.Imaging
```

Inside your class, add the following function:

```
Public Shared Function ChangeOpacity(ByVal img As Image, ByVal opacityvalue As Single) As Bitmap
    Dim bmp As New Bitmap(img.Width, img.Height)
    Dim graphics__1 As Graphics = Graphics.FromImage(bmp)
    Dim colormatrix As New Imaging.ColorMatrix
    colormatrix.Matrix33 = opacityvalue
    Dim imgAttribute As New ImageAttributes
    imgAttribute.SetColorMatrix(colormatrix, ColorMatrixFlag.Default,
    ColorAdjustType.Bitmap)
    graphics__1.DrawImage(img, New Rectangle(0, 0, bmp.Width, bmp.Height), 0, 0,
    img.Width, img.Height,
    GraphicsUnit.Pixel, imgAttribute)
    graphics__1.Dispose()
    Return bmp
End Function
```

We can use this function to set the opacity of an image in a PictureBox.



## btnFadeOut

Add the following line of code to its **Click** event:

```
imgSlimer.Image = ChangeOpacity(imgSlimer.Image, 0.8)
```

Try some different values for the **opacityvalue**. Keep the values between 0 and 1.

*What happens as you make the number smaller/larger?*

*What happens if you click on the Button repeatedly?*

## btnFadeIn

Add the following line of code to its **Click** event:

```
imgSlimer.Image = ChangeOpacity(imgSlimer.Image, 1.2)
```

Try some different values for the **opacityvalue**. Keep the values greater than 1 (but still small).

*What happens as you make the number smaller/larger?*

*What happens if you click on the button repeatedly?*

## btnAppear

To make slimer re-appear fully, the best way is to dispose of the current image and reset it with the following code:

```
imgSlimer.Image.Dispose()  
imgSlimer.Image = My.Resources.slimer
```

## Things to try on your own:

- Add a **TextBox** so that the user can type an opacity value and click a Button to apply it.
  - You will need a Double variable for the **opacityvalue**.
  - Use the code from **btnAppear** to start with a fully opaque image each time.
  - Can you make it so that the user can type a number between 1 and 10 instead?  
1 → 0.1  
5 → 0.5  
10 → 1
- See if you can add a **Timer(s)** to make slimer slowly fade in, and/or out.
  - You will need a Double variable for the **opacityvalue**.
- Try to use a **TrackBar** to allow the user to select its opacity.
  - You will need a Double variable for the **opacityvalue**.
  - Set the max/min properties to 0 and 10.
  - Use the Scroll event to read in the value, and reset the image each time.

