

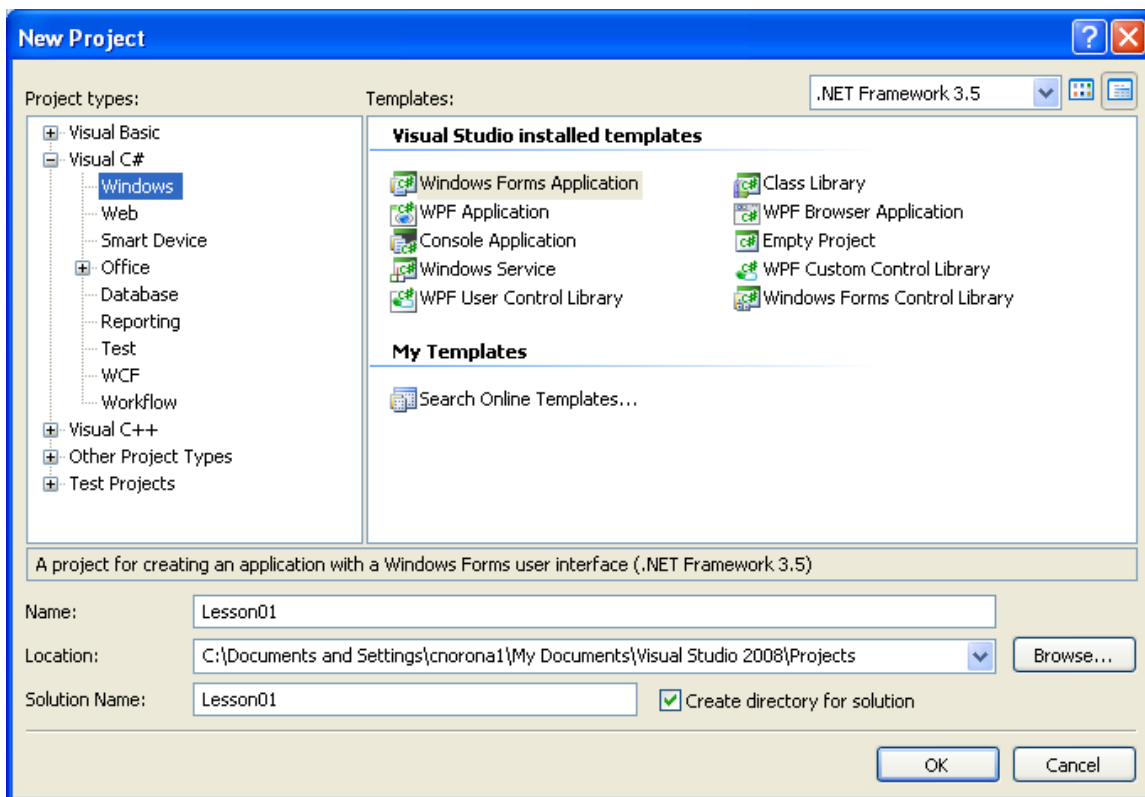
## Creating Windows User Interfaces with .NET

Written by Charles Norona  
Cnorona1@fau.edu  
29<sup>th</sup> of January, 2009

Note: This tutorial was created with the use of Microsoft Visual Studio 2008.

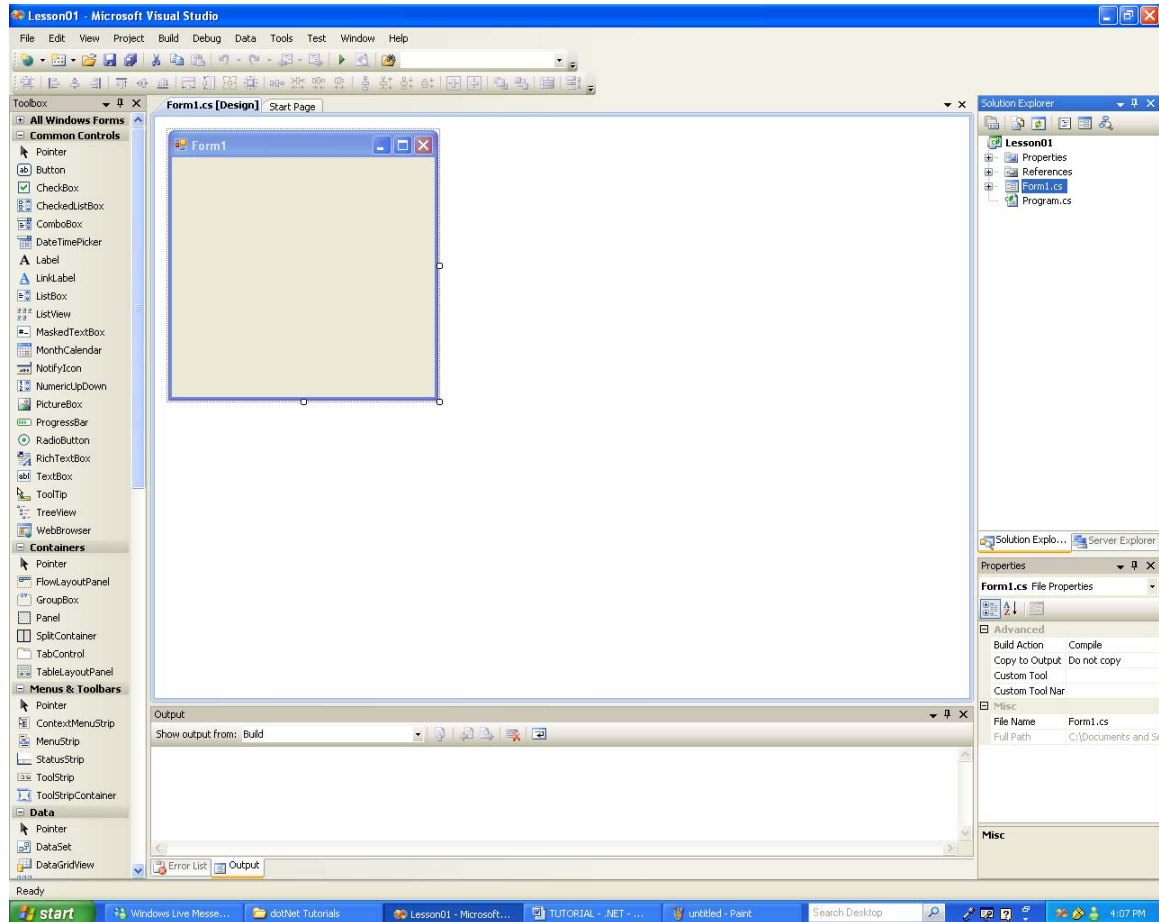
Visual studio has an Integrated Development Environment which automates the creation of software, especially user interfaces (Stellman, 3). This tutorial will focus on the creation of user interfaces as windows form.

Open the new project window and select **Windows Form Application** under the Windows section of C#.



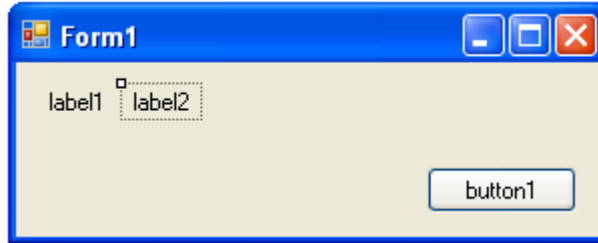
Enter a name for the project. For the sake of this tutorial you can call it "Lesson01" then select **OK**.

Visual Studio will present you with the next window which will serve as a blank canvas for creating user interfaces.



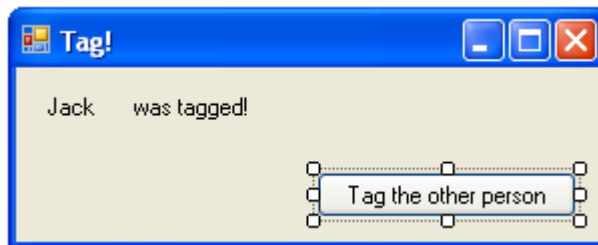
The panel on the left is known as the **Toolbox** which contains all of the user interface objects such as buttons, labels, checkboxes, etc. The panels on the right are the **Solution Explorer** and the **Properties** panels. The solution explorer helps keep track of files associated with your project. The properties panel shows you the properties of various objects in the interfaces you create. In this tutorial we will make a simple UI that displays a text to simulate a game of tag between two children.

Resize Form1 then drag and drop the **label** and **button** objects from the toolbox and resize them as necessary. Try to mimic the one depicted below.



The next step is to configure the properties of these objects. First click on Form1 then look for the **Text** field of the appearance section in the properties panel and enter “Tag!” Click on **label1** and do the same but type in “Jack” into the text field of the appearance section of properties. For **label2**, enter “was tagged!” Click the **button1** object and change its text property to “Tag the other person” and stretch the push button so that the text shows.

Your user interface should now look like this.



We now have a simple user interface but for sake of completion, double-click the push button and you should get a view of the form’s code along with a newly placed skeleton function of what happens when the push button is pressed.

```
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace Lesson01
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            //Code for button behavior goes here.
        }
    }
}
```

Enter the following code into the button1\_Click() function to make label1 change text.

```
if (label1.Text == "Jack")  
    label1.Text = "Jill";  
else  
    label1.Text = "Jack";
```

When finished, build the project and run it. This is just about as simple as it can get without doing nothing! Typically, user interfaces have a lot more going on. To learn more about the other tools you can use to create windows user interfaces visit [http://msdn.microsoft.com/en-us/library/ms173080\(VS.80\).aspx](http://msdn.microsoft.com/en-us/library/ms173080(VS.80).aspx).

Using Visual Studio makes it easy to create programs that involve user interfaces. Normally, there is a lot more code going on but Visual Studio takes care of that for you. In this tutorial, you were only introduced to the the presentation aspect of the code. Most of the time there is a business logic layer beneath what goes on in the presentation layer which usually involves coding algorithms with the use of classes (Hamza-Lup).

## References

Hamza-Lup, Dr. Georgian. Consultation. January 2008.  
Stellman, Andrew and Greene, Jennifer. Head First C#. O'Reilly Media, Inc. Sebastpool, CA. November, 2007. First ed.