

Tutorial: .NET Windows Forms

By

Adnan Zejnilovic

Thursday, April 9, 2015

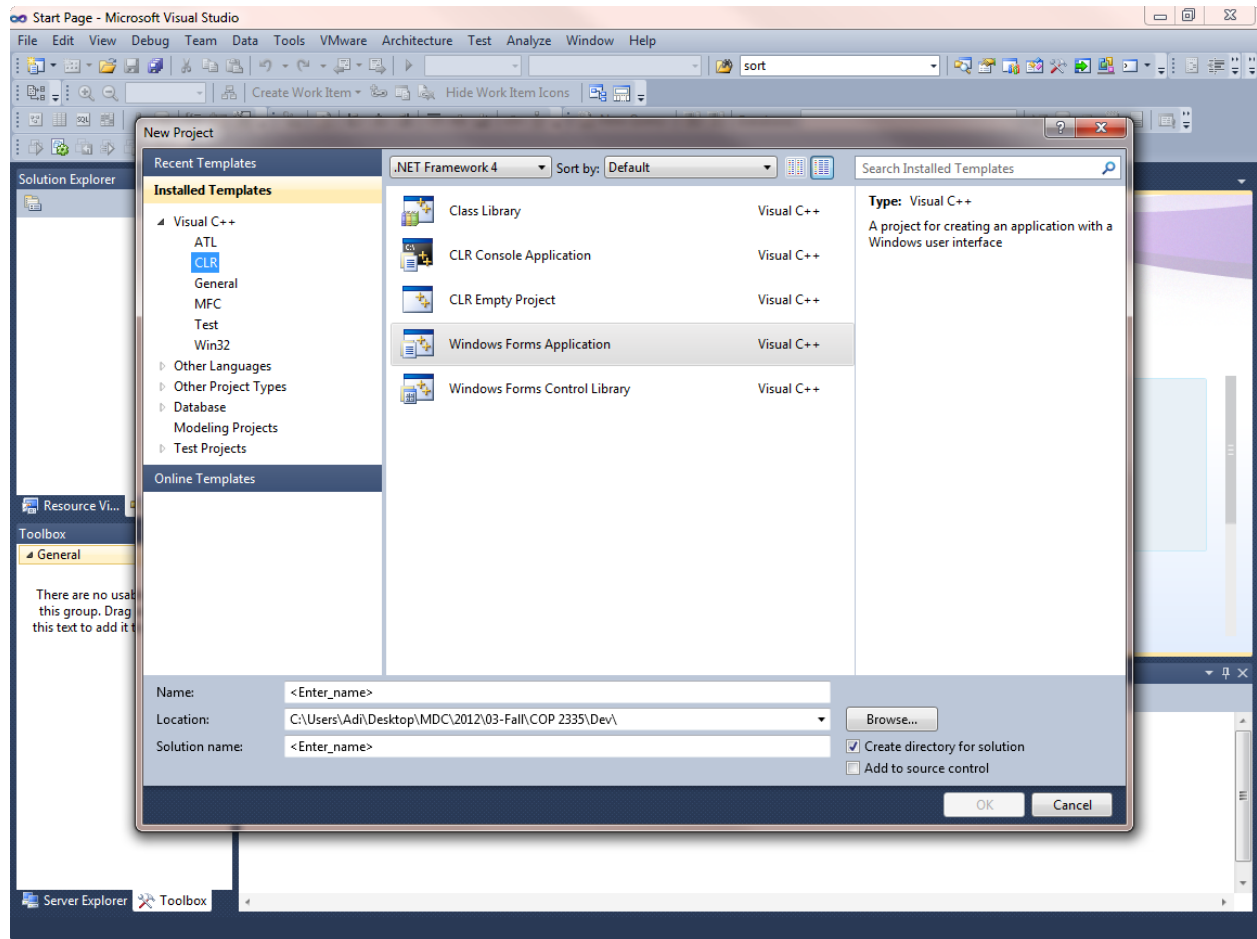
COP 2335

Spring 2015



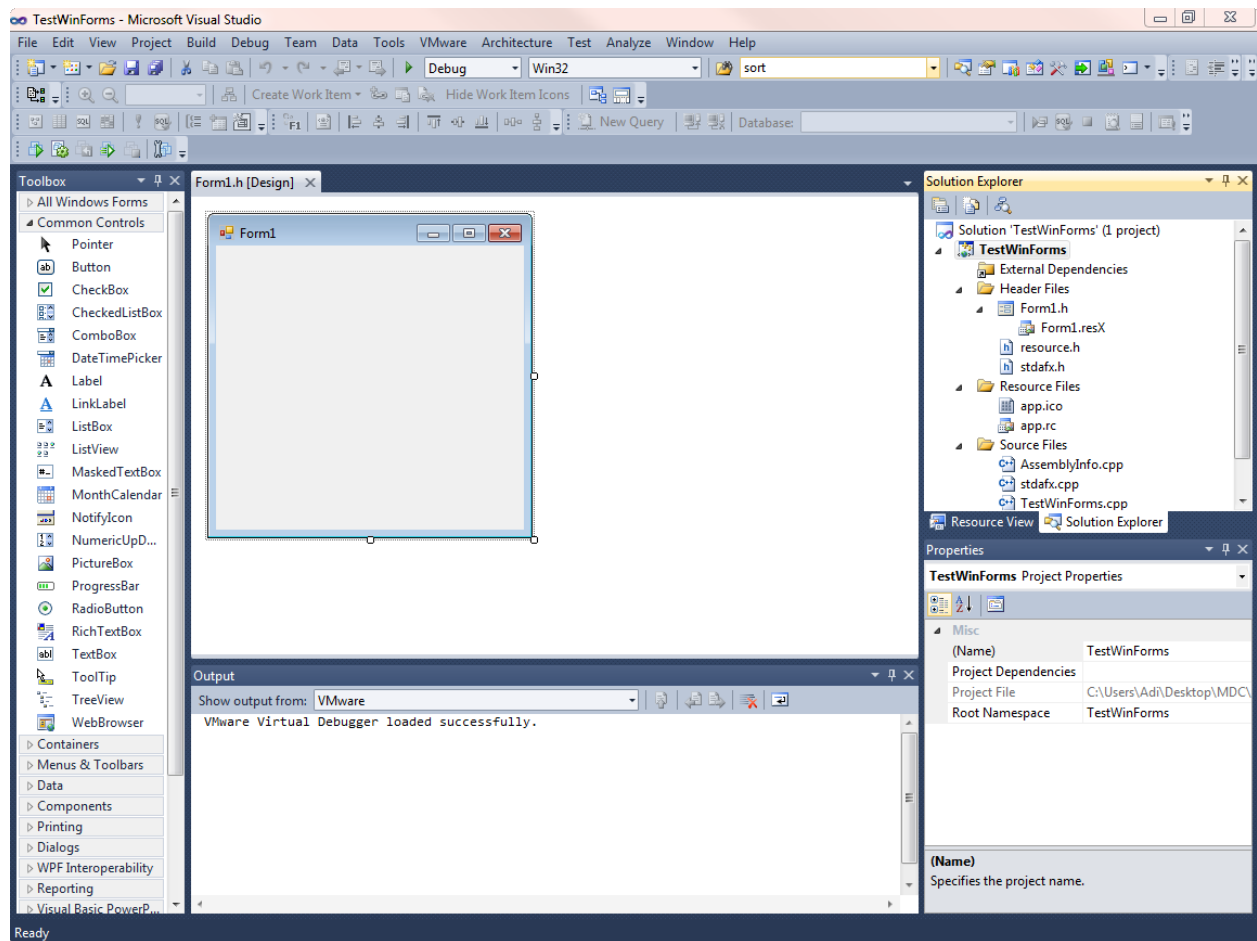
Wolfson Campus

This tutorial briefly describes how to use Microsoft .NET to create Windows Forms application. We'll start by creating a new CLR project (Windows Forms):



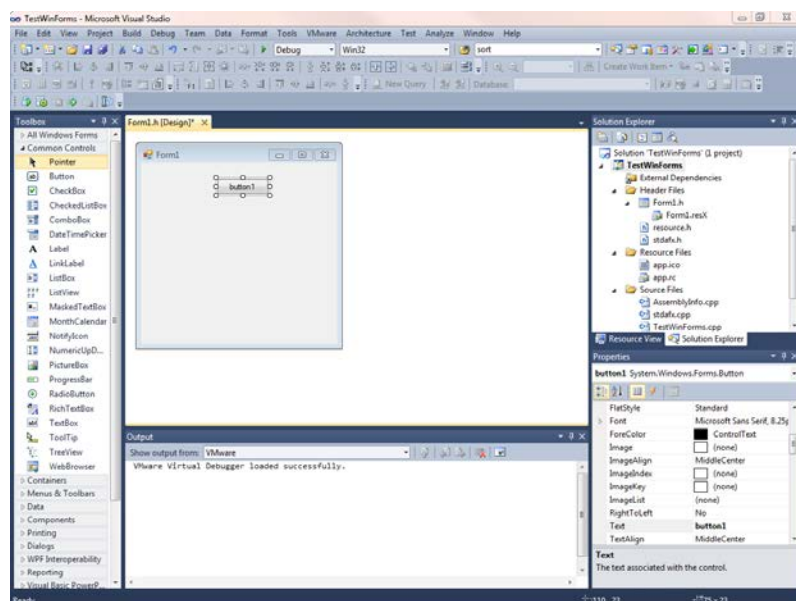
Choose the location of your project and also choose the name of your application. If you leave the “Create directory for solution” check box checked, Visual Studio will create a directory for your project (solution).

You will be presented with the following screen containing a blank form, a Toolbox which contains all the screen elements that you can drag and drop onto the form, and Properties pane. You can arrange them as you please by dragging and dropping (“docking”) panes. I like to have my Toolbox on the upper left hand side and the Solution Explorer pane in the upper right hand side of the screen with the Properties pane below it:

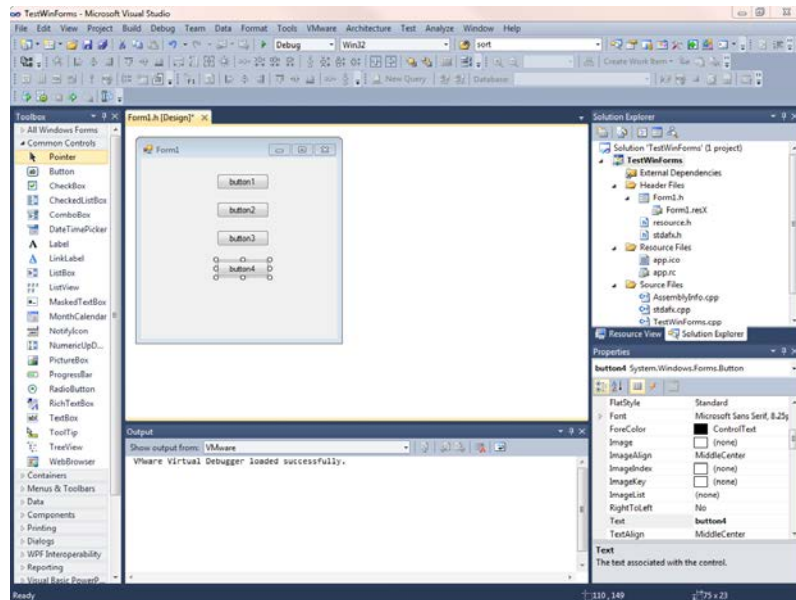


Add Visual Components to the Form

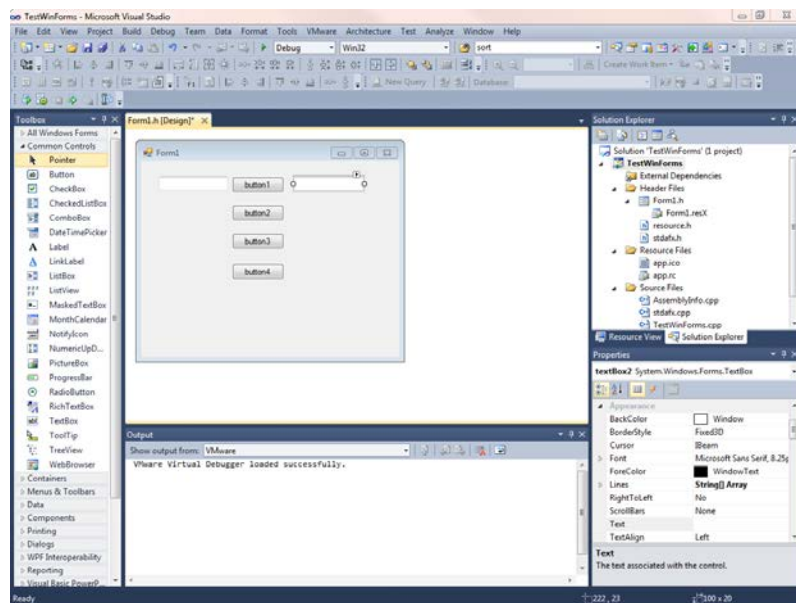
Select a button from the toolbox and drag it onto the Form:



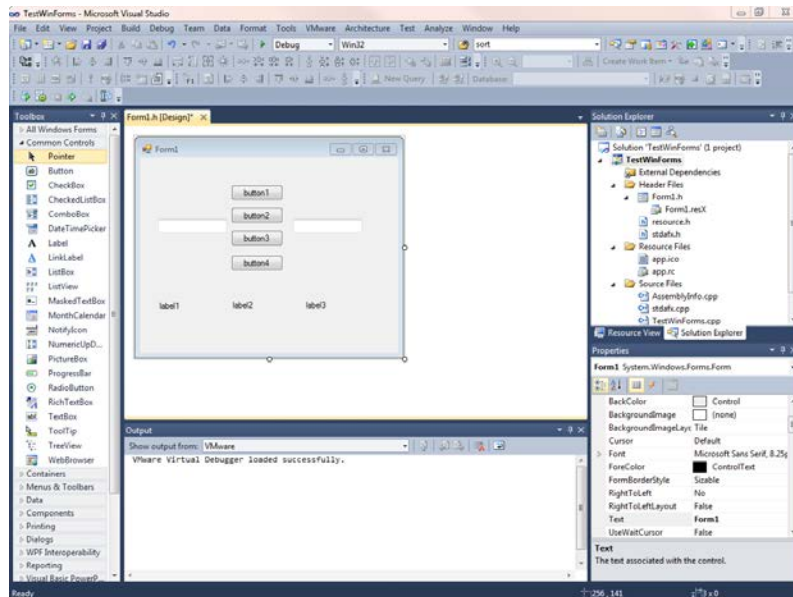
Repeat with 3 more buttons:



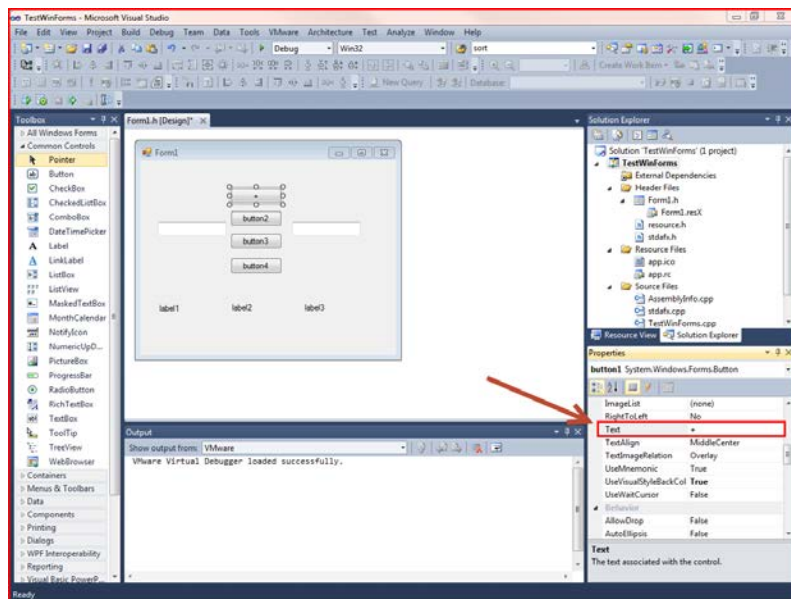
Next, add a couple TextBox controls:



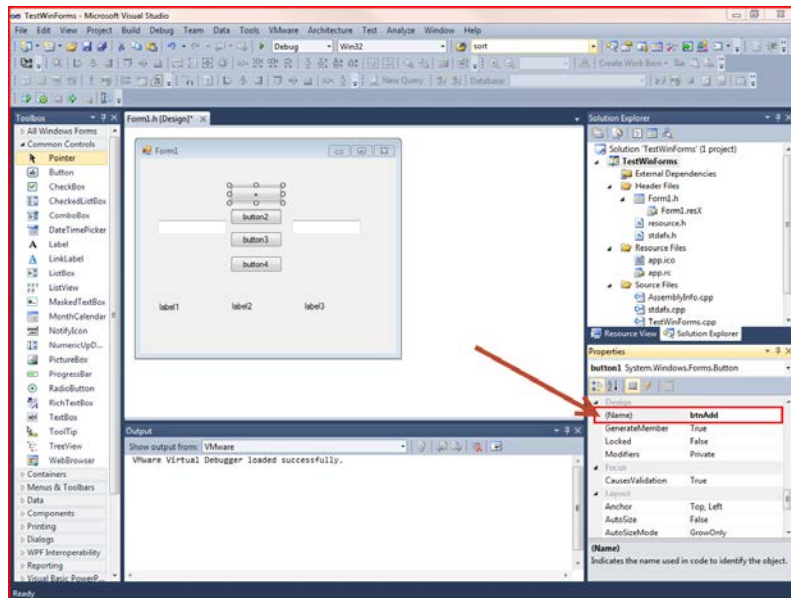
Add a few labels and rearrange textboxes and buttons a little bit. You may need to make your form a bit bigger:



Next we are going to start changing properties of the controls we added to the form. We'll start with buttons. Select "button1" and then in the Properties pane find the Text property. It should read "button1". Change it to "+".



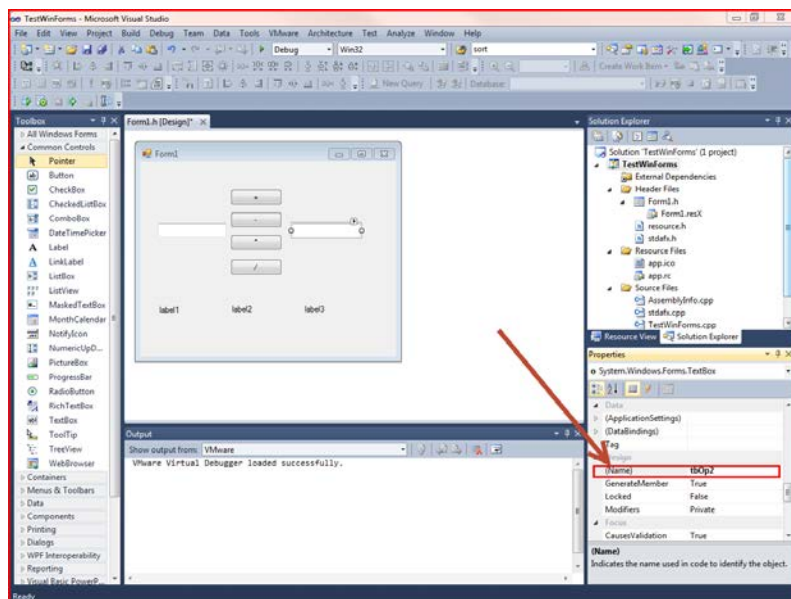
Next, find the **(Name) Property**. It should also read **"button1"**. Change it to **btnAdd**:



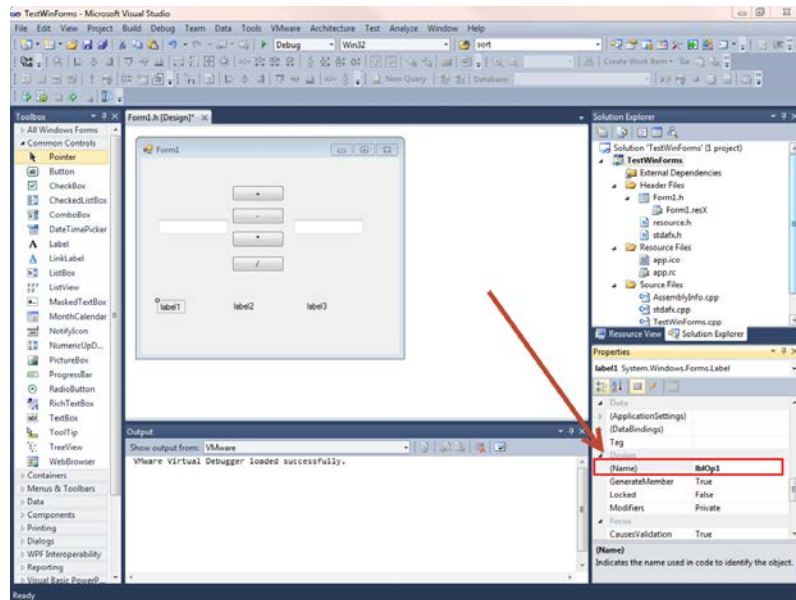
Repeat this process for button2, button3, and button4:

- button2: Text: "-" and (Name) **btnSub**
- button3: Text: "*" and (Name) **btnMult**
- button4: Text: "/" and (Name) **btnDiv**

Next, let's change the properties (Name) for the TextBox controls. We are going to call them **tbOp1** and **tbOp2**:

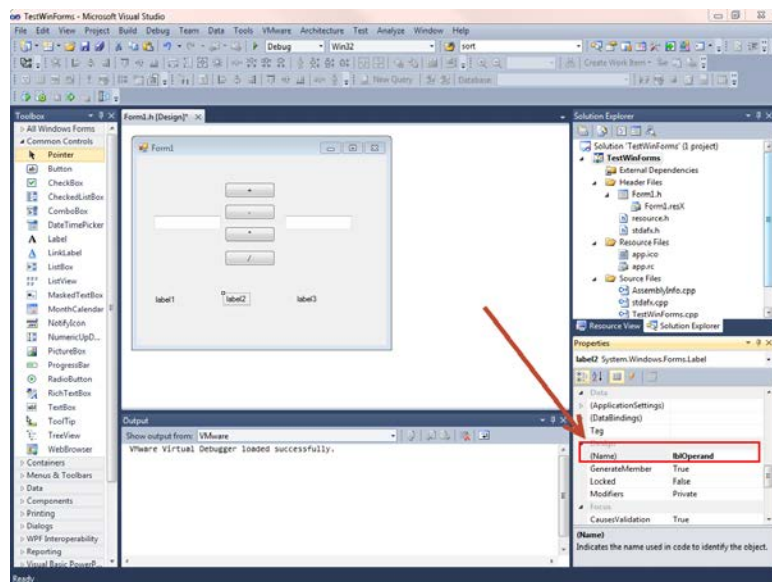


Next, let's change a few properties of labels:

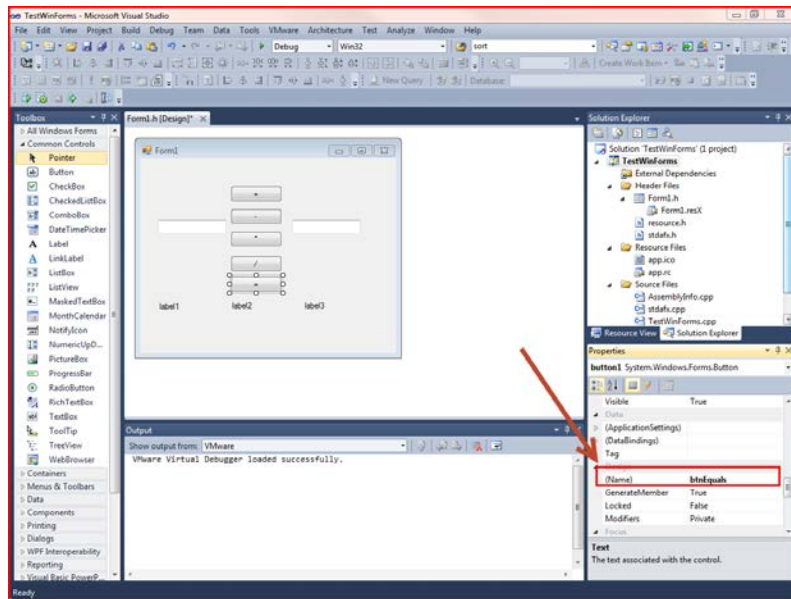


Do the same for label3 – call it **lblOp2**.

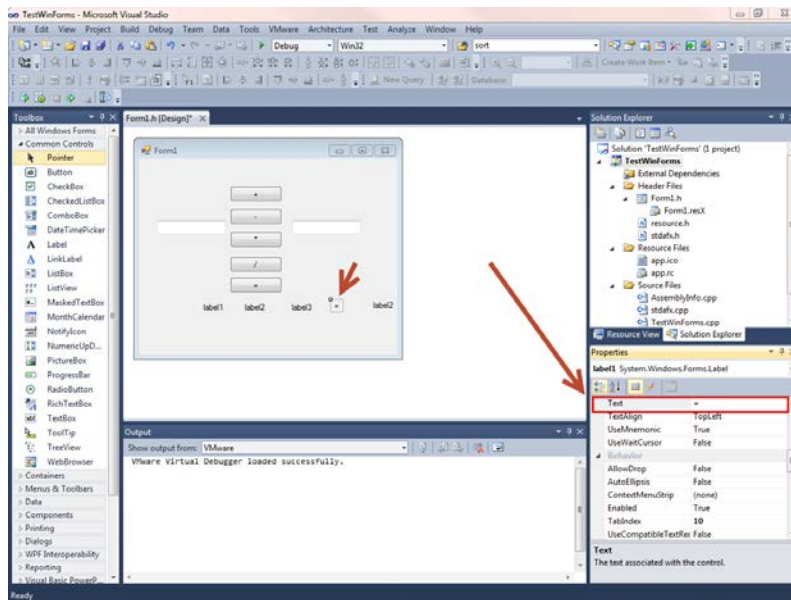
For label2, we are going to call it **lblOperator**:



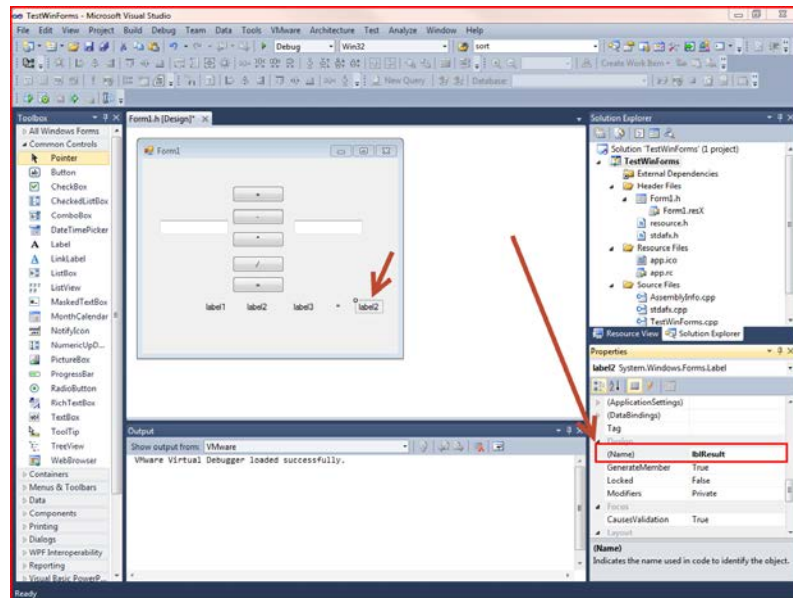
We are almost ready to code at this point. We need to add a few more controls though. Let's add one more button call it **btnEquals**:



And one 2 more labels – a label to display an equal sign “=” :

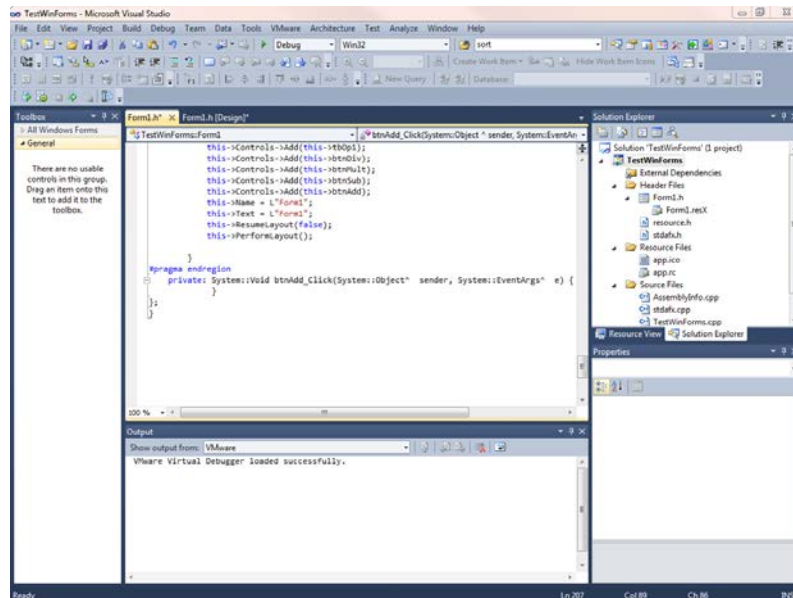


And the last label, we are going to name **lblResult**:



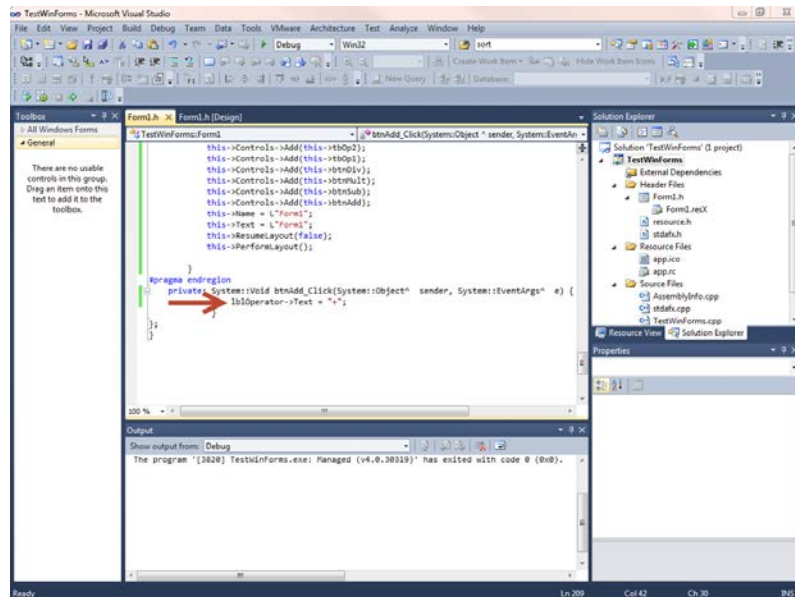
Write Event Handlers

To write an event handler, all you need to do is select a control you want to write an event handler for and double click on it. We'll start with the Add button:

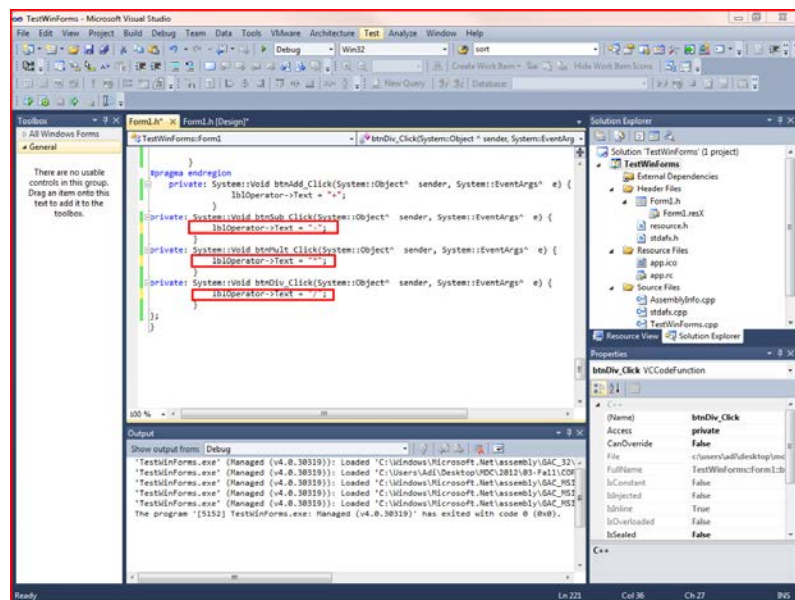


Screenshot after I double clicked on the **btnAdd**.

Now we need to write some code in the supplied event handler stub. We want to change the operator label:

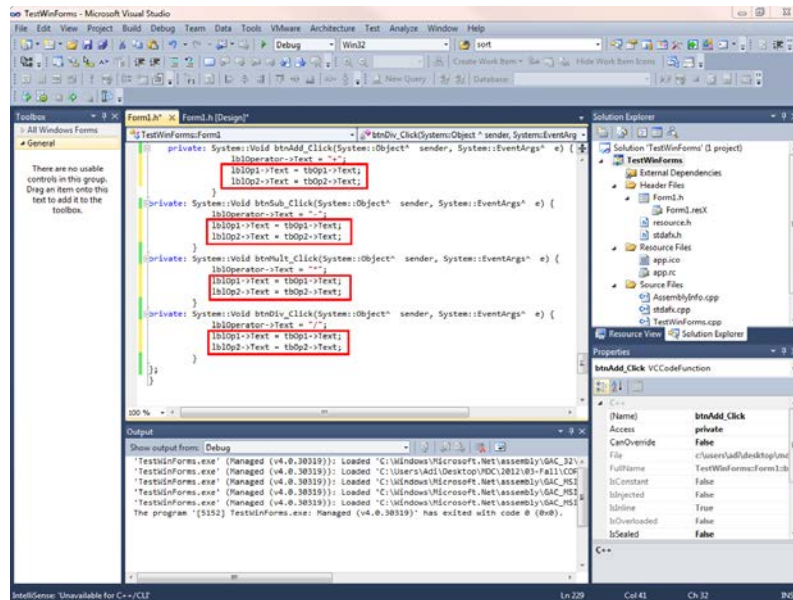


Do the same for **btnSub**, **btnMult**, **btnDiv**:



Note the signs change in the **lblOperator->Text** property according to the button pressed.

Next, let's update text properties of **lblOp1** and **lblOp2**:



Finally, let's add an event handler for the Equals button and the result label:

```
private: System::Void btnEquals_Click(System::Object^ sender, System::EventArgs^ e) {
    char op;
    double x, y, r;
    op = System::Convert::ToChar(lblOperator->Text);
    x = System::Convert::ToDouble(tbOp1->Text);
    y = System::Convert::ToDouble(tbOp2->Text);

    switch(op)
    {
        case '+':
            r = x + y;
            break;
        case '-':
            r = x - y;
            break;
        case '*':
            r = x * y;
            break;
        case '/':
            r = x/y;
            break;
    }

    lblResult->Text = System::Convert::ToString(r);
}
```

The following two statements:

```
x = System::Convert::ToDouble(tbOp1->Text);
y = System::Convert::ToDouble(tbOp2->Text);
```

simply capture the text from the two text boxes and converts them to a double.

This statement takes the result (a double) and converts it to String so that it can be assigned to the text property of lblResult:

```
lblResult->Text = System::Convert::ToString(r);
```

Compile and Run the Program

Here is a screen shot of an addition of 4 and 5:

