Creating Your First C# Application

If you had trouble with the “first application” tutorial listed in the Getting Started panel of the main window please try again using the detailed guide below to build the Knock-Knock project:

Step 1: Launch Visual C# from Start, Programs.

Step 2: Under the File menu, choose New Project. This will open the New Projects window as shown in Figure 1.

Step 3: Click on the Windows Application icon, shown shaded in Figure 1. And name your new project by selecting and replacing the text in the Name textbox. We have named our example project KnockKnock.

Step 4: Creating the new project may take a few moments, so please be patient. No pending process progress bar will appear. Eventually the Start page panel will be covered by a panel named Form1.cs [Design]*.

Figure 1: Visual C# 2005 New Project Window

Figure 2: Opening the Properties Window
Depending upon the state of this program you may or may not see the Properties panel below the Solution Explorer panel on the right side of the program window. If the Properties panel is not visible then open it from the View menu as shown in Figure 2.

The properties of the program we are creating are listed in this panel. We can change the text that appears at the top of the dialog box of the program window by editing the Text property listed in this panel. To demonstrate, find the Text property and change the text value from Form1 to Knock Knock (Fig 3). Note that the new text now appears on the top of the dialog box shown in the Form1.cs [design] panel.

You may want to review some of the other properties listed in the Properties panel, such as BackColor, Icon, Font, and Size. At this time, do not change any other properties.

**Step 5:** Returning again to the View menu, make sure that the Toolbox is selected. When active, the Toolbox panel will appear on the left-hand side of the main window. We want to add a Label and a Button to our form so we will first select Label in the Toolbox and drag (while holding down the left mouse button) a Label onto our form as shown in Fig 4.

With the Label on the form selected, change the Text property of this label to “Knock Knock”.

**Step 6:** Now select the Button item in the Toolbox list and drag a button onto the form to the right of the Knock Knock label. Change the Text property of the Button to “Who's There?”. You may need to resize the button by selecting it and dragging its edge (left or right side) to make all of the text on the button visible. (see Figure 5)

At this point you may want to test your program by compiling and running it. If not, please move along to Step 7. To test the program, press the F5 function key or click on the small triangle in the task bar at the top of the main window. This will start the debugger and (if there are no errors) execute an instance of your program. No actions will result from clicking the Who'sThere? button since we have not yet programmed any actions to occur. To end this program you can click on the Close (X) button located in the standard position on the right-hand side of the top of the Knock Knock dialog box.
While we're at it, let's take a look at the code generated so far. To view the code, double click the top of the dialog box shown in the Form1.cs [designer] panel. This will display a new panel showing the source code similar to that shown below:

```csharp
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Text;
using System.Windows.Forms;
namespace KnockKnock
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }
    }
}
```

This source code includes a number of standard libraries that have been added automatically to your program including the System library, Drawing and Text management libraries and a Windows.Forms library. We will look into the contents of these and many other standard libraries later. For now, it is enough to note that the Microsoft Visual C# 2005 development environment is supporting the graphical and functional design and implementation of your program. To return to the [designer] panel, simply click on the Form1.cs [designer] tab.

Step 7: Double click the Who'sThere? button to create and display the source code associated with this item. Click on the Form1.cs tab to view this newly generated code that has been added to your program. This code should appear similar to,

```csharp
private void button1_Click(object sender, EventArgs e)
{
}
```

We need to modify this method to perform the desired operation. Find the method entitled button1_Click and add the line of source code `labell.Text = "Orange"` to this method. The modified method should appear as,

```csharp
private void button1_Click(object sender, EventArgs e)
{
    labell.Text = "Orange";
}
```

Step 8: Compile and run an instance of your program by pressing the F5 function key or by clicking the Start Debugger triangle on the main window taskbar.

Optional Project

Want to have some fun? Let's use a little of the power of Visual C# 2005 to add the punch line to our Knock Knock program. When you have the Knock Knock program described above working, return to this section... Ready? Here we go...

Return to the Form1.cs [designer] panel and move the button down a bit so that a longer labor will not be blocked by the button. Go back to the button1_Click( ) method and make the following modifications.
private void button1_Click(object sender, EventArgs e)
{
    if (label1.Text == "Knock Knock")
    {
        label1.Text="Orange";
        button1.Text="Orange Who?";
    }
    else if (label1.Text=="Orange")
    {
        label1.Text="Orange you glad you're studying C#?";
        button1.Text="Oh Yeah";
    }
    else if (button1.Text=="Oh Yeah")
    {
        Close();
    }
}

Rather than just slapping the keyboard and hoping for the best, you might consider building this compound if-then conditional statement one step at a time. The first conditional checks to see if the current Text of Label1 is "Knock Knock". If this is true then this label's Text is changed to "Orange", and the Text on button1 is also changed to "Orange Who?".

The second conditional checks to see if the current Text of label1 is "Orange". Is so, then the Text of label1 is changed to the punch line "Orange you glad your studying C#?" and the Text on button1 is changed to "Oh Yeah".

The third conditional checks to see if the Text of button1 is "Oh Yeah". If this is the case the program is closed using the Close( ) method. You will notice that a drop-down menu appears occasionally as you type. This menu displays a list of the many methods, attributes and other features associated with this object. We will study many of these in detail later. For now you can just continue typing. (As an alternative, you may cut and paste this code directly into your program, assuming that you are using the same object names throughout your source code.

**Designer Code** - You may be curious about the source code corresponding to the graphical representation of the program shown in the Form1.cs [designer] panel. To see this code you can double-click the Form1.Designer.cs link in the Solution Explorer panel on the right-hand side of the main window of Visual C# 2005. Your source code should be similar to the code shown below. Note that the attributes (e.g. size and text values) of the label the button and the form itself are listed in this automatically generated code.

```
namespace KnockKnock
{
    partial class Form1
    {
        /// <summary>
        /// Required designer variable.
        /// </summary>
        private System.ComponentModel.IContainer components = null;

        protected override void Dispose(bool disposing)
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
    }
}
```

1 Hey, there were no claims made concerning the level of jocularity of the punch line.
private void InitializeComponent()
{
    this.label1 = new System.Windows.Forms.Label();
    this.button1 = new System.Windows.Forms.Button();
    this.SuspendLayout();
    //
    // label1
    //
    this.label1.AutoSize = true;
    this.label1.Location = new System.Drawing.Point(48, 63);
    this.label1.Name = "label1";
    this.label1.Size = new System.Drawing.Size(72, 13);
    this.label1.TabIndex = 0;
    this.label1.Text = "Knock Knock";
    //
    // button1
    //
    this.button1.Location = new System.Drawing.Point(113, 89);
    this.button1.Name = "button1";
    this.button1.Size = new System.Drawing.Size(113, 23);
    this.button1.TabIndex = 1;
    this.button1.Text = "Who\'s There?";
    this.button1.UseVisualStyleBackColor = true;
    this.button1.Click += new System.EventHandler(this.button1_Click);
    //
    // Form1
    //
    this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
    this.ClientSize = new System.Drawing.Size(357, 196);
    this.Controls.Add(this.button1);
    this.Controls.Add(this.label1);
    this.Name = "Form1";
    this.Text = "Knock Knock";
    this.Load += new System.EventHandler(this.Form1_Load);
    this.ResumeLayout(false);
    this.PerformLayout();
}

#endregion

private System.Windows.Forms.Label label1;
private System.Windows.Forms.Button button1;

The designer-generated code is kept separate from programmer code to avoid unnecessary complexity. This code separation is a simple form of abstraction that helps the programmer concentrate on the important elements of program design and implementation. We recommend that (at least for now) you do not attempt to modify the designer-generated code directly. If you wish to change the graphical layout of your program, do so from the Form1.cs [design] panel, by physically moving and resizing the objects and/or changing their properties in the Properties panel.