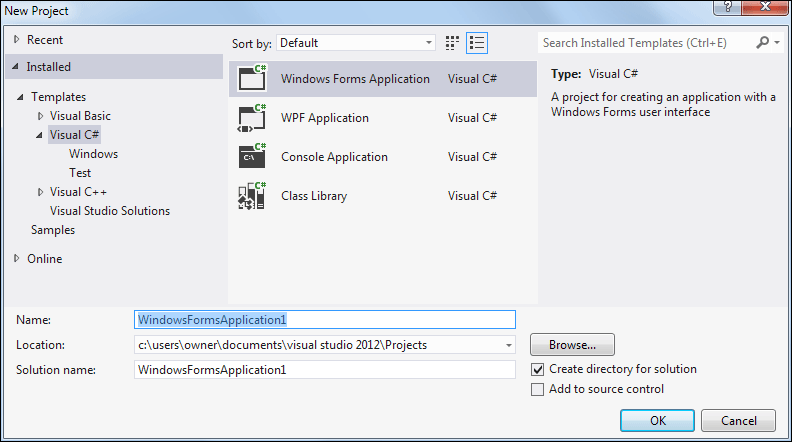
My First Windows Application

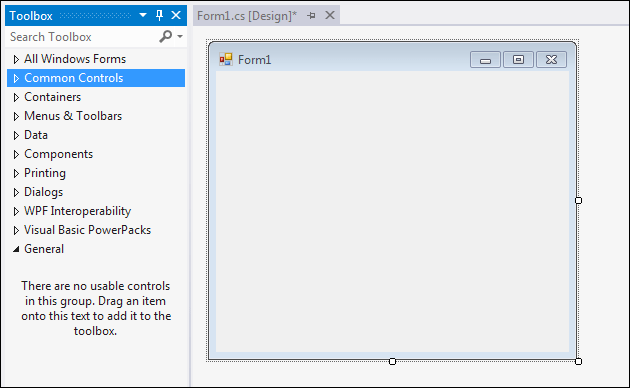
Windows Applications make use of something called a **Form**. The Form is blank at first. You then add control to your form, things like buttons, text boxes, menus, check boxes, radio buttons, etc. To get your first look at a Windows Form, do the following.

To create your first Windows form project, click the File menu and select **New Project** from the menu. When you do, you'll see the New Project dialogue box again. Click Visual C#, under **Templates** on the left:



Select **Windows Forms Application** from the available templates (Community 2015 users should see more templates in the middle). Keep the **Name** on the default of **WindowsFormsApplication1** and then click OK.

When you click OK, a new Windows Application project will be created for you:

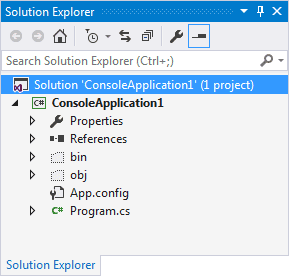
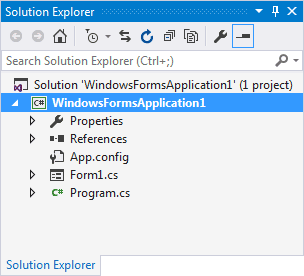


Notice the **Toolbox** on the left hand side. We'll be adding controls from the Toolbox to that blank **Form1** you can see in the image above.

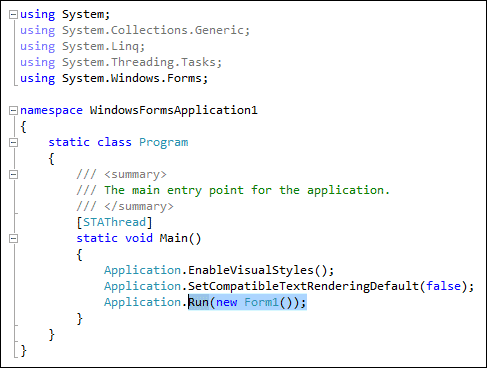
If you want to permanently display the Toolbox, click on the pin symbol in the middle:

Click the Pin symbol

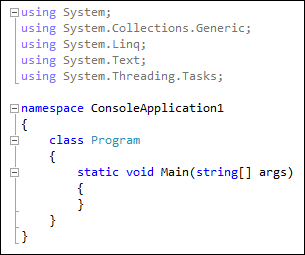
Notice the Solution Explorer on the right side of your screen. If you compare it with the Solution Explorer when you created your Console Application, you can see the similarities:

Both projects have sections for Properties, References, and a Program.cs file. Double click the Program.cs file to open it, and you'll see some familiar code:



And here's the code from the Console Application:



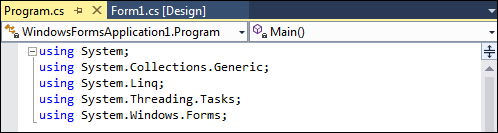
Both have the same **using** lines, a **namespace**, a class called Program, and a **Main** Method.

The Main Method is the entry point for your programme. The code between the curly brackets of Main will get executed when the programme first starts. The last line in the **WindowsApplication1** code above is the one that Runs Form1 when the Application starts.

You can do other things here. For example, suppose you had a programme that connects to a server. If it finds a connection then it loads some information from a database. In the Main Method, you could check that the server connection is OK. If it's not, display a second form; if it's OK, then display the first form.

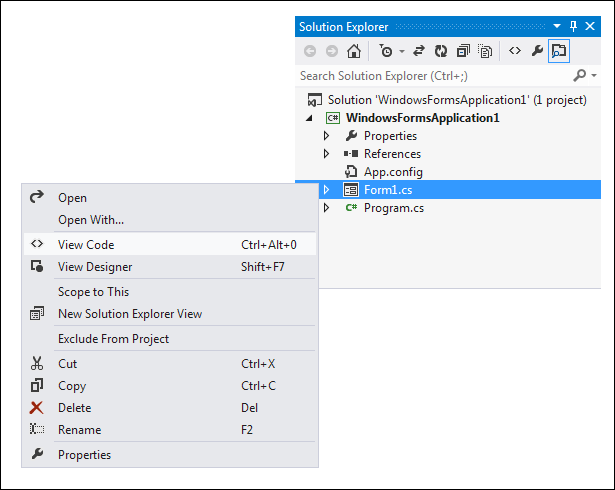
But don't worry if all that code has you scratching your head. The thing to bear in mind here is that a method called **Main** starts your programme. And **Program.cs** in the Solution Explorer on the right is where the code for Main lives.

But we won't be writing code in the Program.cs file, so we can close it. Have a look near the top of the coding window, and you'll some tabs:

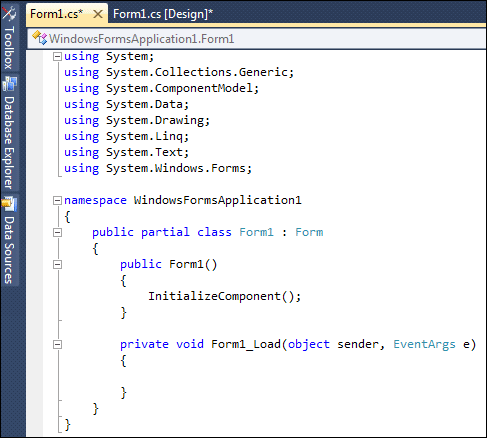


Click the X to close the tab. You should now see your form again (you may have a Start tab as well. You can close this, if you want).

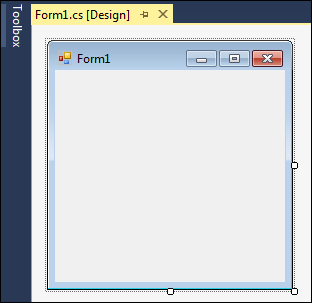
To see the window where you'll write most of your code, right click **Form1.cs** in the Solution Explorer:



The menu has options for **View Code** and **View Designer**. The Designer is the Form you can see at the moment. Click **View Code** from the menu to see the following window appear:

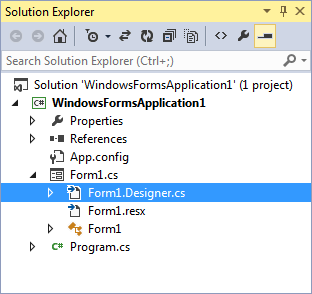


This is the code for the Form itself (ignore the Form1\_Load lines as you may not have them). This Form:

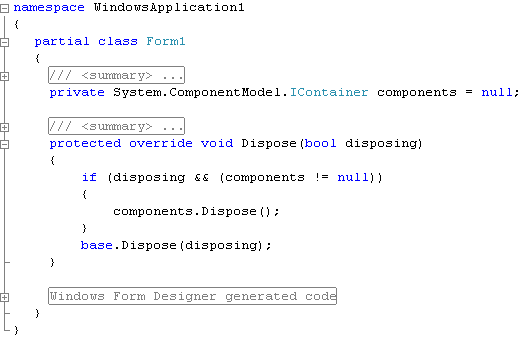


The code has a lot more **using** statements than before. Don't worry about these for now. They just mean "using some code that's already been written".

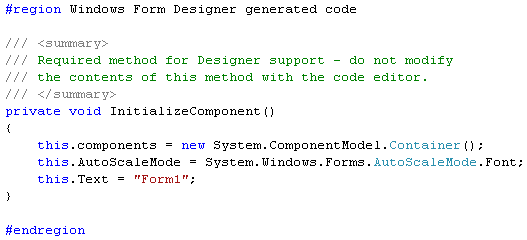
The code also says **partial class Form1**. It's partial because some code is hidden from you. To see the rest of it (which we don't need to alter), click the arrow symbol next to Form1.cs in the Solution Explorer:



Now double click **Form1.Designer.cs**. You'll see the following code:



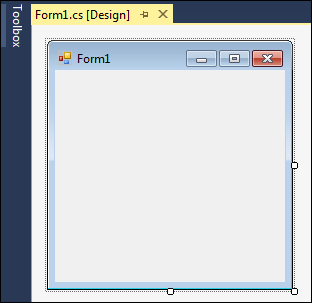
Again, you see **partial class Form1**, which is the rest of the code. Click the plus symbol next to **Windows Form Designer generated code**. You'll see the following:



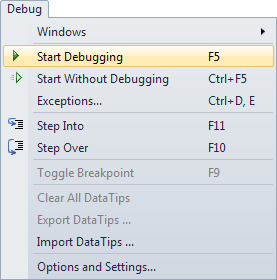
**InitializeComponent** is code (a Method) that is automatically generated for you when you create a new Windows Application project. As you add things like buttons and text boxes to your form, more code will be added here for you.

But you don't need to do anything in this window, so you can right click the **Form1.Designer.cs** tab at the top, and click **Close** from the menu. Or just click the X.

Click back on the **Form1.cs** tab at the top to see you form again. If the tab is not there, right click Form1.cs in the Solution Explorer on the right. From the menu, select **View Designer**. Here's what you should be looking at:



It's in Designer view that we'll be adding things like buttons and text boxes to our form. But you can run this programme as it is. From the **Debug** menu at the top, click **Start Debugging** (Or you can just press the F5 key on your keyboard.):



When you click Start Debugging, Visual C# will Build the programme first, and then run it, if it can. If it can't run your programme you'll see error messages.

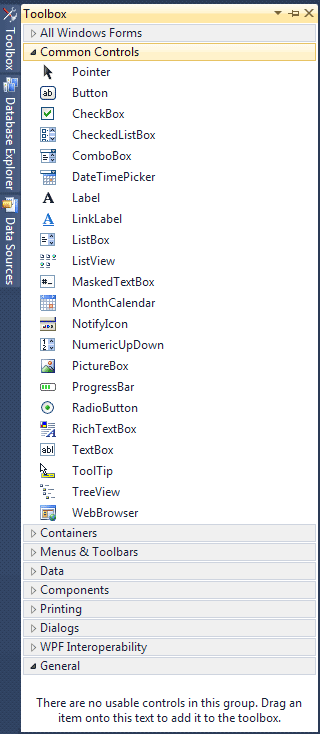
But you should see your form running on top of Visual Studio. It will have its own Red X, and it's own minimize and maximize buttons. Click the Red X to close your programme, and to return to Visual C# Express.

From now on, when we say Run your programme, this is what we mean: either press F5, or click **Debug > Start Debugging**. You can also select **Debug > Start Without Debugging**.

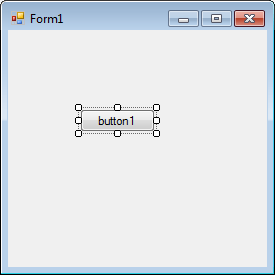
OK, time for you to start adding things to the form, and to do a little coding!

The first thing we'll do is to add a button to the blank form. We'll then write a single line of code, so that you can see how things work.

If you want to add a control to a form, you can use the Toolbox on the left of Visual Studio. Move your mouse over to the Toolbox, and click the arrow symbol next to Common Controls. You should see the following list of things that you can add to your form:



Click the **Button** item under the **Common Controls** heading. This will select it. Now click once anywhere on your form. A button will be drawn for you, and your Form will look like this:

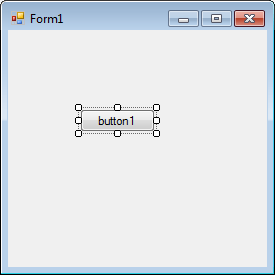


(You can also hold down your left mouse button and drag out a button to the size you want it.)

A button is something you want people to click on. When they do, the code you write gets executed. The text on the button, which defaults to "button1", can be changed. You can add anything you like here, but it should be something that's going to be useful for your users, such as "Open a text file", or "Calculate Now".

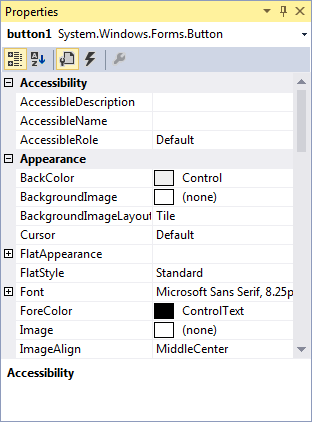
We're going to display a simple message box when the button is clicked. So we need to add some text to the button. You do this by changing something called a property.

The controls you add to a form have something called Properties. A property of a control is things like its Height, its Width, its Name, its Text, and a whole lot more besides. To see what properties are available for a button, make sure the button is selected, as in the image below:

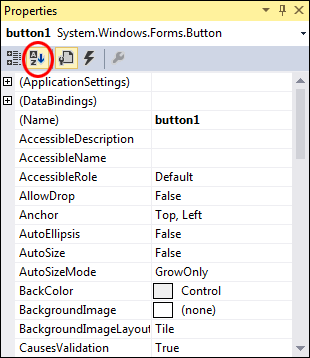


If a control is selected, it will have white squares surrounding it. If your button is not selected, simply click it once.

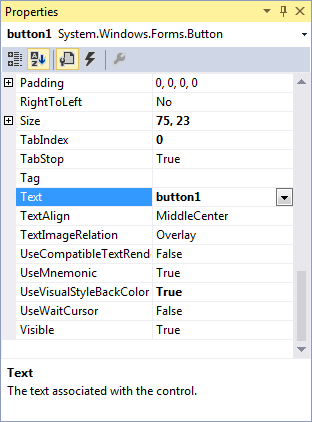
Now look in the bottom right of Visual C#, just below the Solution Explorer. You should see the Properties Window (if it's not there, select it from the **View** menu at the top):



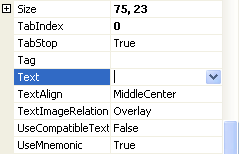
To view the list of Properties in alphabetical order, click the AZ symbol at the top, circled in red in the image below:



As you can see, there's a lot of Properties for a button. Scroll down to the bottom and locate the **Text** Property:



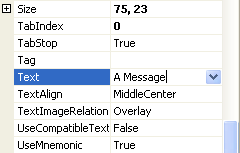
The **Text** Property, as its name suggests, is the Text you want to appear on the button. At the moment, it says button1. Click inside of the text area of button1. Delete the default text:



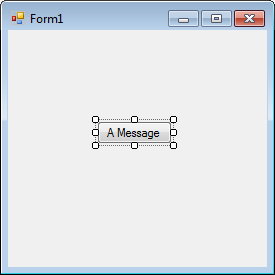
Now type the following:

**A Message**

The Text part of your Properties Window will then look like this:

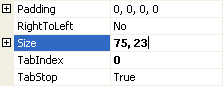


Now press the enter key on your keyboard. Have a look at your Form, and the Text on the button should have changed:

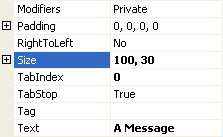


There's a few more Properties we can change before we get to the code.

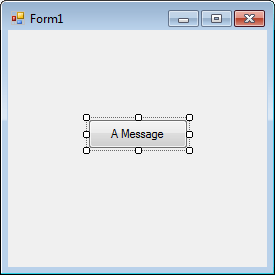
Locate **Size** in the Properties Window:



The first number, 75, is the width of the button. The second number, 23, is the height of the button. The two numbers are separated by a comma. Change the numbers to 100, 30:



Press the enter key again, and the size of your button will change:



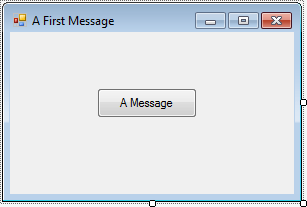
You can move your button around the Form by clicking it with the left mouse button to select it. Hold down you left mouse button and drag your button around the form. Let go of your left mouse button when you're happy with the new location.

**Exercise**  
You can also move a button by changing its **Location** property. Use the Properties Window to change the Location of your button to a position on the form of 100, 50. (100 means 100 units from the left edge of the form; 50 means 50 units down from the top of the form.)

**Exercise**  
A Form also has lots of Properties. Click away from the button and on to the Form itself. The Properties for the form will appear in the Properties Window. Change the **Text** Property of the Form to **A First Message**.

**Exercise**  
The Form, like the button, also has a Size Property. Change the Size of the Form to 300, 200.

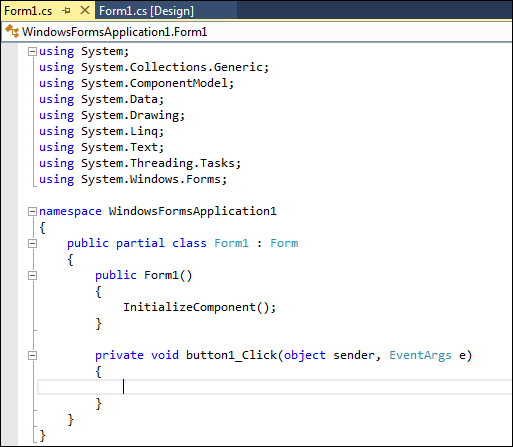
After you complete the three exercises above, your Form should look like this:



When you changed the Text property of the Form, you changed the text that runs across the blue bar at the top, the text in white. You can type anything you like here, but it should be something that describes what the form is all about or what it does. Often, you'll see the name of the software here, like Microsoft Word, or Microsoft Visual Studio.

We can now move on to some code, though, and then run the Form to see what it all looks like.

What we want to do now is to display a message box whenever the button is clicked. So we need the coding window. To see the code for the button, double click the button you added to the Form. When you do, the coding window will open, and your cursor will be flashing inside of the button code. It will look like this:



The only difference from the last time you saw this screen is the addition of the code for the button. This code:

private void button1\_Click(object sender, EventArgs e)  
{

}

This is just another Method, a piece of code that does something. The name of the Method is **button1\_Click**. It's called button1 because that's currently the **Name** of the button. When you changed the Text, Location, and Size properties of the button, you could have also changed the Name property from button1 (the default Name) to something else.

The **\_Click** part after button1 is called an **Event**. Other events are MouseDown, LocationChanged, TextChanged, and lots more. You'll learn more about Events later.

After **\_Click**, and in between a pair of round brackets, we have this:

object sender, EventArgs e

These two are know as **arguments**. One arguments is called **sender**, and the other is called **e**. Again, you'll learn more about arguments later, so don't worry about them for now.

Notice that there is a pair of curly brackets for the button code:

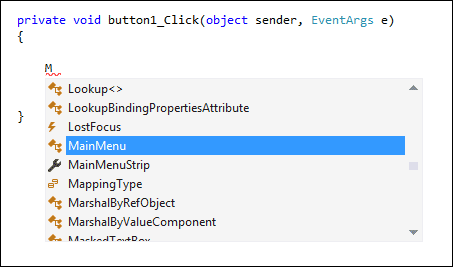
private void button1\_Click(object sender, EventArgs e)  
**{**

**}**

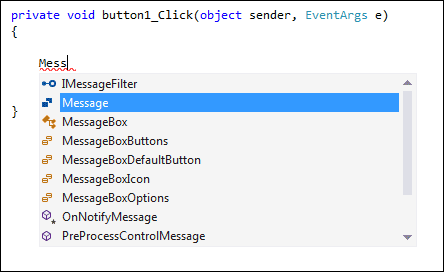
If you want to write code for a button, it needs to go between the two curly brackets. We'll add a single line of code in the next part below.

We want to display a message box, with some text on it. This is quite easy to do in C#.

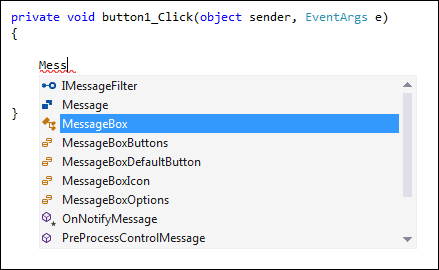
Position your cursor between the two curly brackets. Then type a capital letter "M". You'll see the IntelliSense list appear:



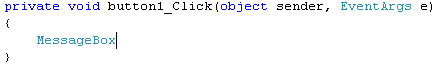
Now type "ess" after the "M". IntelliSense will jump down:



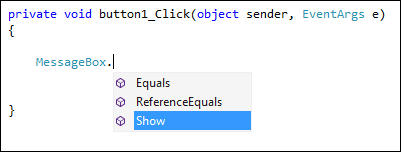
The only options that start with Mess are all Message ones. The one we want is MessageBox. You can either just type the rest, or even easier is to press the down arrow on your keyboard to move down to MessageBox:



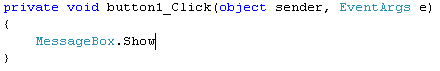
When you have MessageBox selected, hit the enter key on your keyboard (or double click the entry on the list). The code will be added for you:



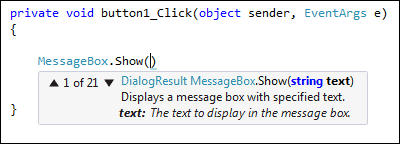
Now type a full stop (period) after the "x" of MessageBox. The IntelliSense list will appear again:



There are only three items on the list now, and all Methods (you can tell they are Methods because they have the purple cube icon next to them) Double click on **Show**, and it will be added to your C# code:



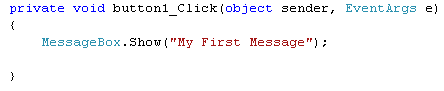
Because **Show** is a Method, we need some round brackets. The text for our message box will go between the round brackets. So type a left round bracket, just after the "w" of "Show":



As soon as you type the left round bracket after the "w", you'll see all the different ways that the Show method can be used. There are 21 different ways in total. Fortunately, you don't have to hunt through them all! Type the following, after the left round bracket (Don't forget the double quotation marks.):

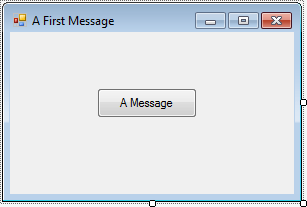
**"My First Message"**

After the final double quote mark, type a right round bracket. Then finish off the line by typing a semi-colon ( ; ), Your coding window will then look like this:



The text in the dark reddish colour is what will be displayed in your message box. To try it out, save your work by clicking **File** from the menu bar at the top of Visual Studio. From the File menu, click **Save All**. You'll then see the same Save box you saw for the Console Application. Save the project.

Run your programme by clicking **Debug > Start Debugging** (or Start Without Debugging). Or just press the F5 key on your keyboard. Your programme will look like this:



Click your button to see your Message Box:



Congratulations! It's your first message!

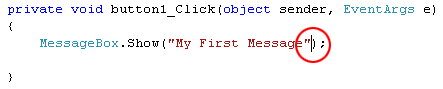
If you look at the message box we created in the previous section, you'll notice there's no Title in the blue area to the left of the red X - it's blank:



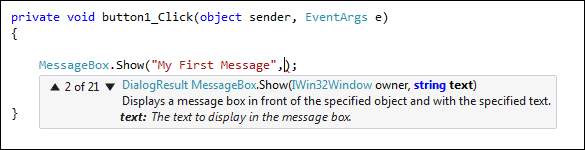
You can add a Title quite easily.

Click OK on your Message Box. Then click the Red X on your programme to exit it. This will return you to Visual C#. Go back to the coding window (press F7 on your keyboard, if you can't see it).

Position your cursor after the final double quote of "My First Message", circled in red in the image below:



Now type a comma. As soon as you type a comma, you'll see the list of **Show** options again:



Type the following:

**"Message"**

Again, you need the double quotes. But your line of code should look like this:



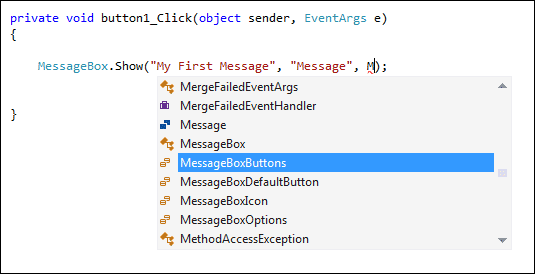
When your line of code looks like the one above, Run your programme again. Click your button and you should see a Title on your Message Box:



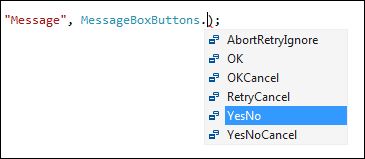
**Other Button Options**

Rather than having just an OK button, you can add buttons like Yes, No, and Cancel to your C# message boxes. We'll add a Yes and a No button.

Return to your coding window. After the second double quote of the Title you've just added, type another comma. Hit the spacebar on your keyboard once, and you'll see the IntelliSense list appear. (If it doesn't appear, just type a capital letter "M").

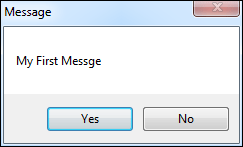


The one that adds buttons to a message box is, you won't be surprised to hear, **MessageBoxButtons**. Press the enter key on your keyboard when this option is highlighted. It will be added to the your code. Now type a full stop (period) after the final "s" of MessageBoxButtons. You'll see the button options:



Double click the one for **YesNo**, and it will be added to your code.

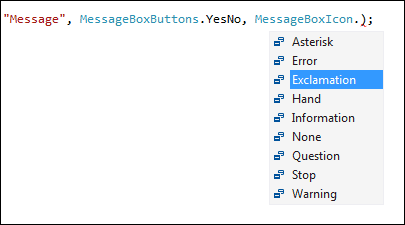
Run your programme again, and click your button. Your Message Box will then look like this:



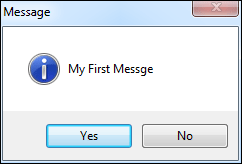
**Adding Icons to a C# Message Box**

Another thing you can add to brighten up your Message Box is an Icon. It's easier to see what these are than to explain!

Type another comma after **MessageBoxButtons.YesNo**. After the comma, type a capital letter "M" again. From the IntelliSense list that appears, double click MessageBoxIcon. After MessageBoxIcon, type a full stop to see the available icons:



We've gone for Exclamation. Double click this to add it to your code. Run your programme again to see what the icon looks like on your Message Box:



Looks pretty impressive, hey! And all that with one line of code!

We'll move on to the important subject of variable, in the next part. First, try this Exercise.

**Exercise**  
Try the other icons on the IntelliSense list, and see what they look like when your programme runs. Does the Information icon differ from the Asterisk? (To quickly display the IntelliSense list again, delete the word Asterisk from your code, then delete the full stop. Type the full stop again, and the IntelliSense list will reappear.)