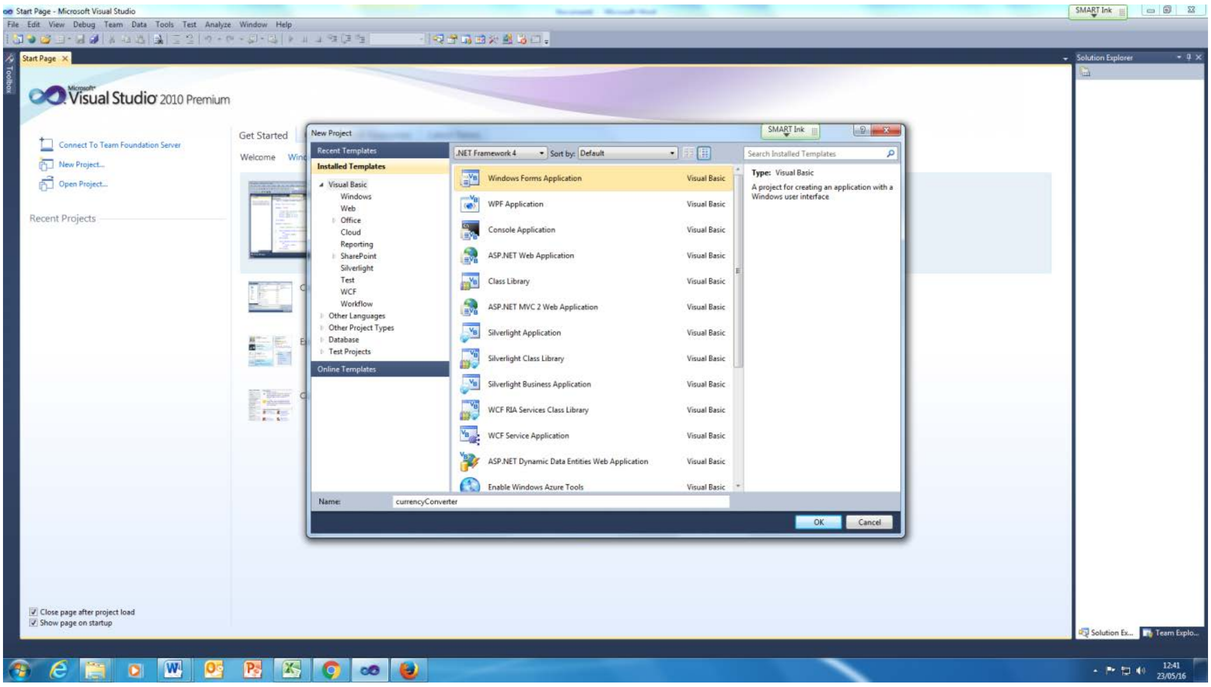
Visual Basic Currency converter

Welcome to the currency converter tutorial in visual basic. In this tutorial we will be using Visual studio 2010 (you can get for free or even the 2015 version) with visual basic programming language. This will be the second currency converter we create in visual basic. The last one we made used Radio buttons to change the currency rates. In this one we will be making one which you can dictate the currency rates and then make sure it does the maths accurately.

Let’s get started.

First create a new visual basic windows form application in visual studio and call it currency converter.



.NET Framework 4 · Sort by: Default

Type: Visual Basic

A project for creating an application with a Windows user interface

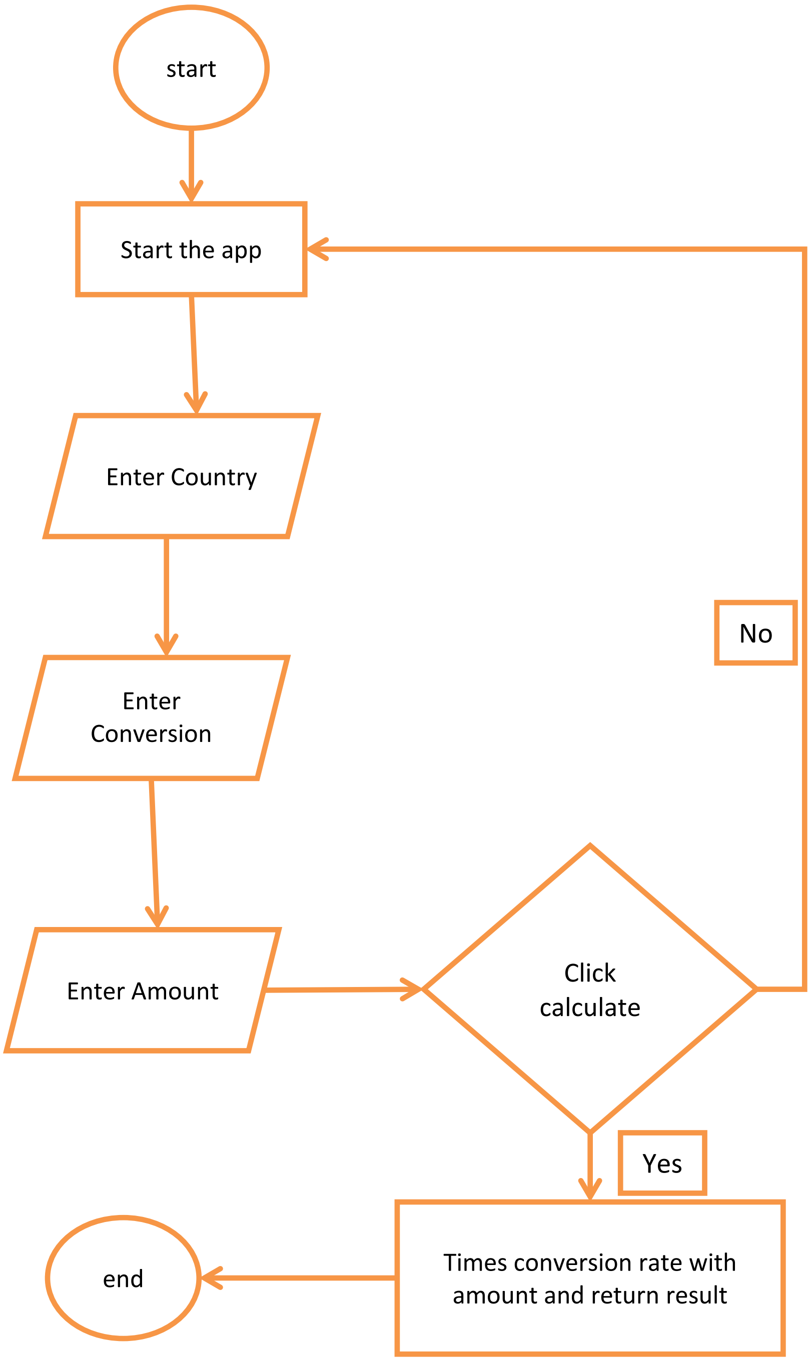
Name: currencyConverter

For this project we will need the following

Component	Description	Name
Textbox	To enter currency rate	TextBox1
Textbox	To enter country of the currency	TextBox2
Textbox	To enter the amount	TextBox3
Button	Calculate the maths	Button1
Label	Show country	Label1
Label	Currency Rate	Label2
Label	Amount	Label3
Label	Show result	Label4

(Note this is not the final list – we will add more components later on)

Here is the flow chart to explain the program



Algorithm

Start the App
Enter country
Enter the conversion rate
Enter the amount to convert

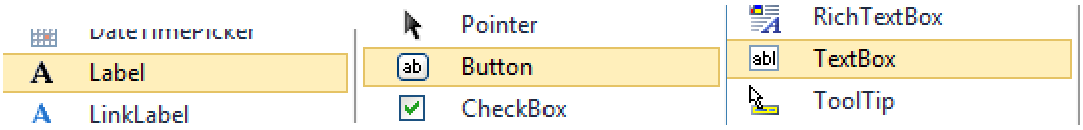
User Clicked button
Times conversion rate with amount and return a total
END

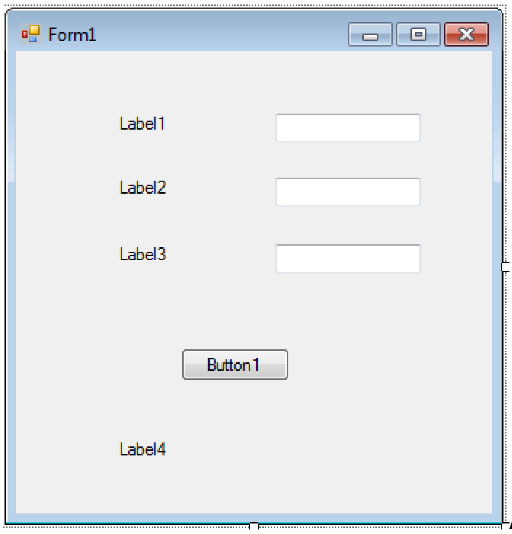
User didn't click button
Return back to the main window
END

More tutorial on **www.mooict.com**

Interface

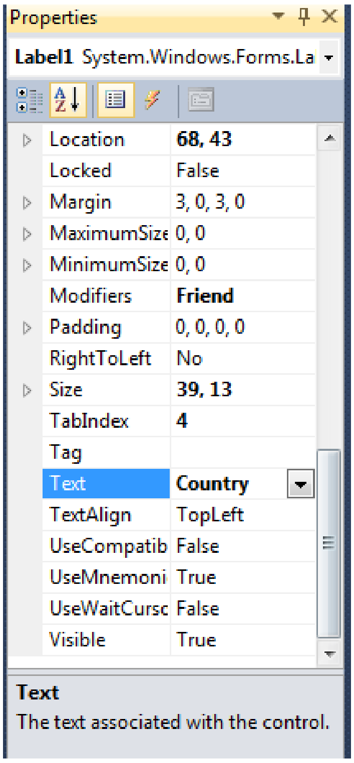
Drag and dop all the components from the toolbox





As you can we have our 4 labels, 3 text boxes and 1 button for the interface of this application.

Lets start changing the texts of the labels on screen. Click on label1 and look into the properties window. You will find an option called text. Inside that we can add any text which will change Label1.

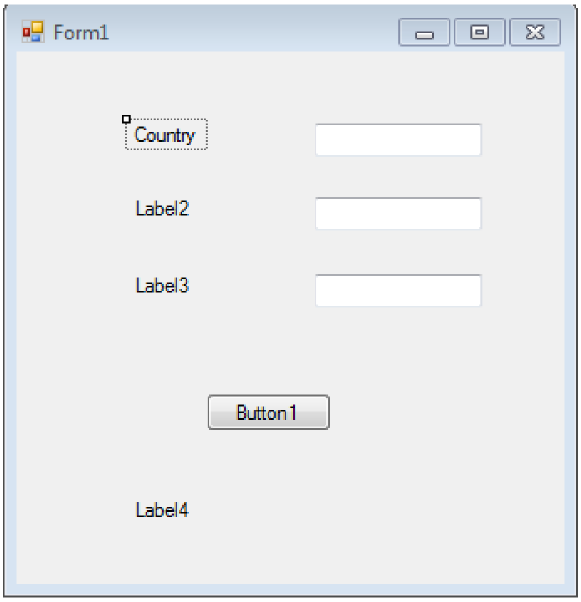


Properties

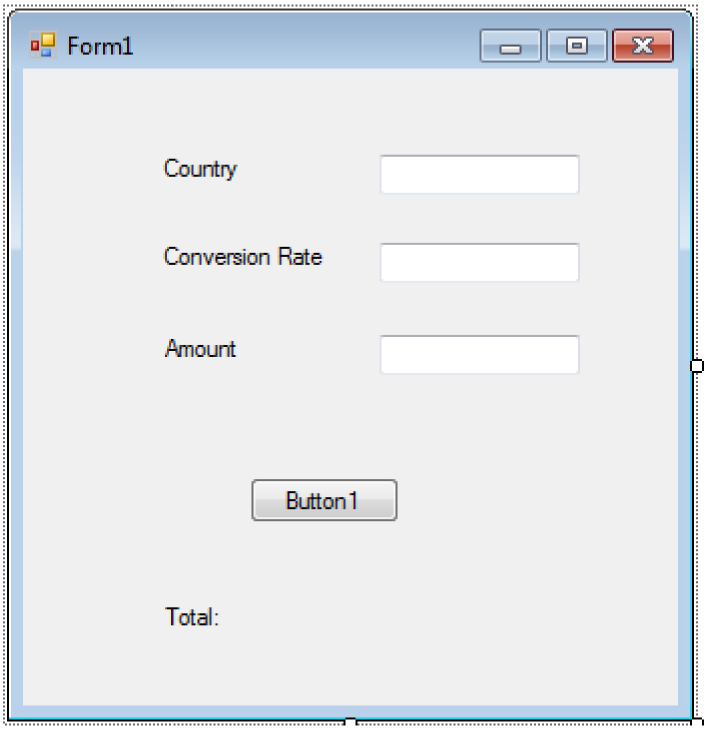
Label1 System.Windows.Forms.La

Location	**68, 43**
Locked	False
Margin	3, 0, 3, 0
MaximumSize	0, 0
MinimumSize	0, 0
Modifiers	**Friend**
Padding	0, 0, 0, 0
RightToLeft	No
Size	**39, 13**
TabIndex	**4**
Tag	
Text	**Country**
TextAlign	TopLeft
UseCompatib	False
UseMnemoni	True
UseWaitCursc	False
Visible	True

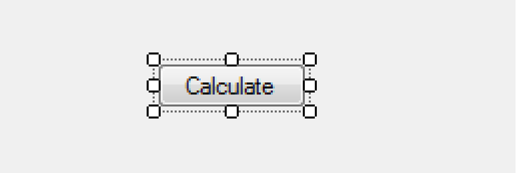
Text
The text associated with the control.



In the properties window there are more options where you can change the font and the colour of the text. We will get to that later on.

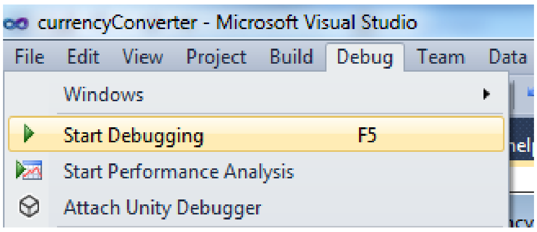


Here is the final view of what the program should look like now with the text changed. You can change the button text as well. Let's give it a text of calculate in it since it will calculate the conversion rate for us.

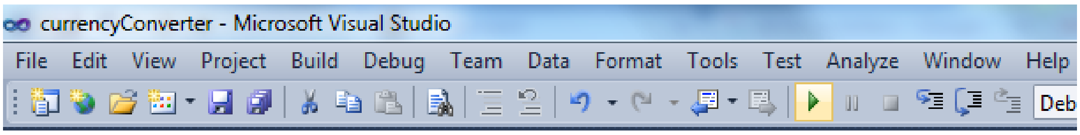


Now run the GUI to see if everything is in order. There are few ways to run an app in visual studio.

1) You can run it by access the debug option in the main menu.



2) Click on the green play button on the tool bar



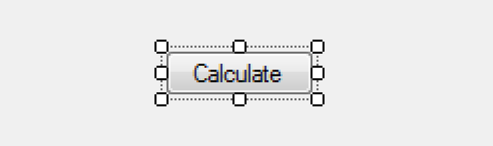
3) Lastly just press F5

See if everything is to your liking and then follow along.

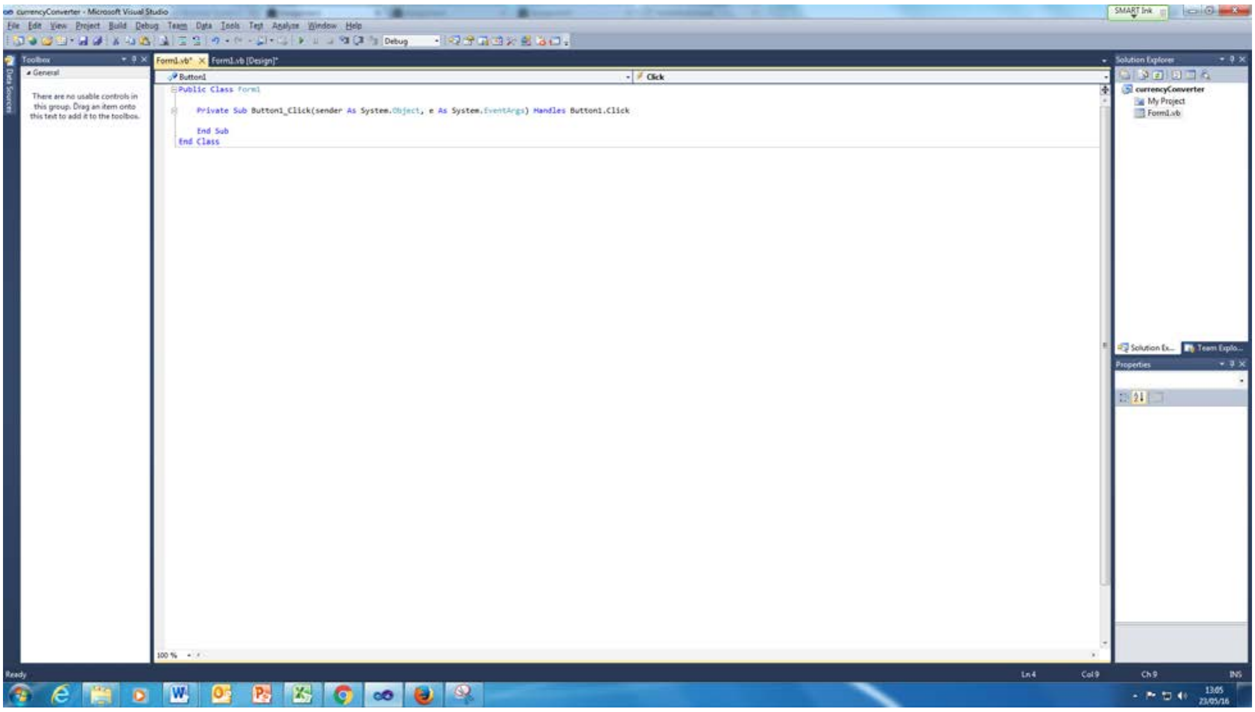
Now then with the interface out of the way we can concentrate on the coding.

More tutorial on **www.mooict.com**

Out program will interact with the user through the button. So what we need to do is double click on the button add an event to it. Inside that event we can add all of our programming logic.



Double click on this


currencyConverter - Microsoft Visual Studio
File Edit View Project Build Debug Team Data Tools Test Analyze Window Help
Debug
Toolbox
General
There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.
Form1.vb*
Form1.vb [Design]*
Button1
Click
Solution Explorer
currencyConverter
My Project
Form1.vb
Properties
Ready

```
Public Class Form1

    Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click

    End Sub
End Class
```

You will see this screen. This is where all the programming codes go in. Our main priority of to convert the currency user put in the boxes and return the converted value to them.

```
Public Class Form1

   Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click

Dim conversionRate As Double = Convert.ToDouble(TextBox2.Text)

       Dim totalAmount As Double = Convert.ToDouble(TextBox3.Text)

   End Sub
End Class
```

Add the two highlighted lines in the button1_click function. We have created two **double** variables to store the information from the text boxes. Text box 2 will hold the conversion rates and text box 3 holds the amount to be converted. The reason we have created them as double is because normal **integers** cannot hold **decimal** numbers.

Integers only hold 1, 2, 400, 321, 834298 etc. it cannot hold values such as 22.8 or 1.99 etc. this is why we need to have a double variable.

```
Public Class Form1

   Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click

```

```
      Dim conversionRate As Double = Convert.ToDouble(TextBox2.Text)

      Dim totalAmount As Double = Convert.ToDouble(TextBox3.Text)

Dim convertedTotal As Double = conversionRate * totalAmount

   End Sub
End Class
```

Lets create another double variable this one will hold the converted total of the amount. As you can see inside the double variable we created called convertedTotal we are calculating it by conversionRate times by totalAmount.

Lets say we want to convert dollars to pounds

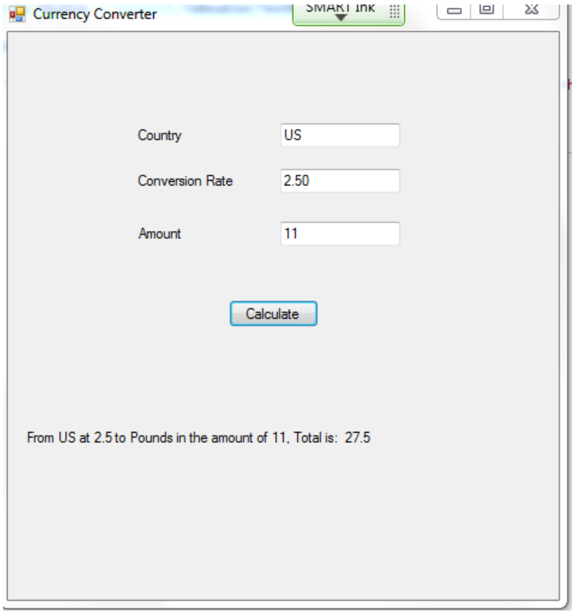
For this purpose 1 pound is equals to 2.50 dollars. So if we had 11 pounds to convert to dollars how much will we have?

11 (times) 2.50 = 27.50 Dollars

Makes sense? Yes

```
Public Class Form1

   Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click

      Dim conversionRate As Double = Convert.ToDouble(TextBox2.Text)

      Dim totalAmount As Double = Convert.ToDouble(TextBox3.Text)

      Dim convertedTotal As Double = conversionRate * totalAmount

Label4.Text = "From " & TextBox1.Text & " at " & conversionRate & " to Pounds in the amount of " &
totalAmount & ", Total is:  " & convertedTotal

   End Sub
End Class
```

Now here is the results line. In this line we are taking label4 and changing the text dynamically to suit our purpose. As you can see first we are calling the label4.text which handles all the string inside the label. We will change it to show for example “Converted from **US** at 2.50 to pounds in the amounts of 11 total is 27.50”.



Country US

Conversion Rate 2.50

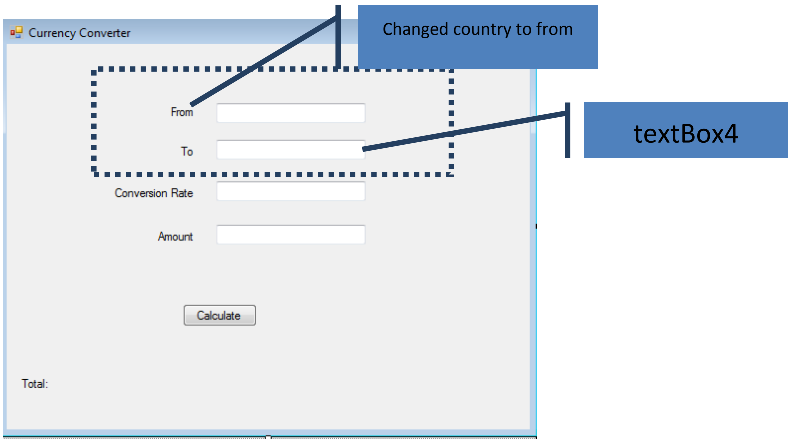
Amount 11

From US at 2.5 to Pounds in the amount of 11. Total is: 27.5

Result is accurate.

This is a working currency converter now. However this program will only convert every other currency to POUND so we need a system where we can change between pound to others as well.

In order to that we will need to another textbox and then change the GUI to the following.



We have added another text box and label to the program and changed the value of country to “From” and the other label to “To”. Now we can manipulate the program to show which currency and from where it was converted.

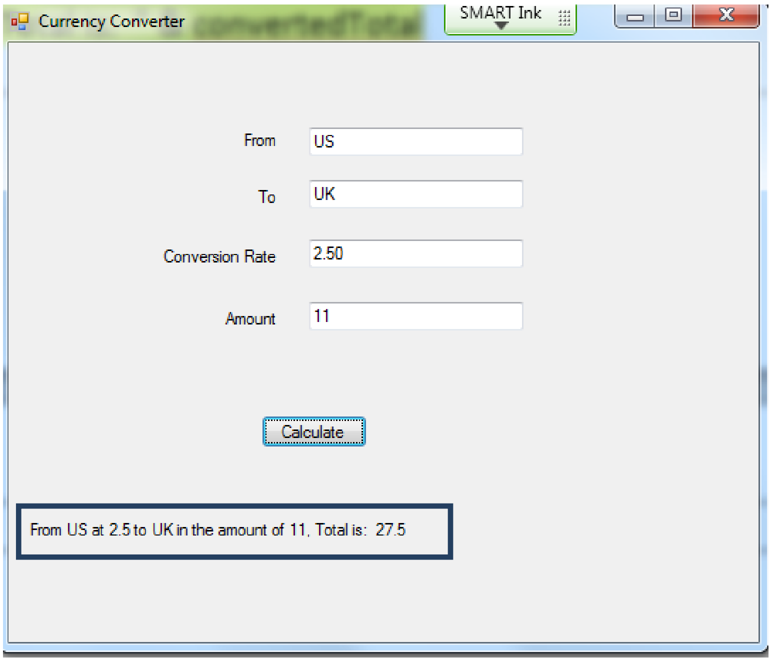
When you are making changes make sure you understand the changes I made to the GUI. Lets make some changes to the code to reflect our new components.

```
    Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click
        Dim conversionRate As Double = Convert.ToDouble(TextBox2.Text)

        Dim totalAmount As Double = Convert.ToDouble(TextBox3.Text)

        Dim convertedTotal As Double = conversionRate * totalAmount

Label4.Text = "From " & TextBox1.Text & " at " & conversionRate & " to " &TextBox4.Text &" in the amount of " & totalAmount & ", Total is:  " & convertedTotal
```

More tutorial on **www.mooict.com**

```
    End Sub
```

Look at the highlighted code in the box. We are calling the value of text box 4 which is the TO information box in the GUI. Now lets run the same calculation as before from US to UK rate is at 2.50 amount of 11



From: US

To: UK

Conversion Rate: 2.50

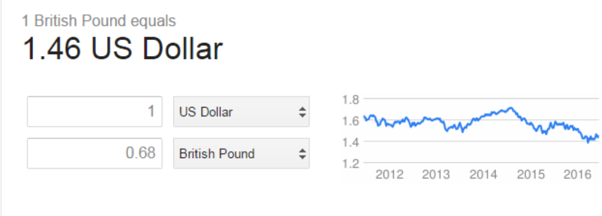
Amount: 11

From US at 2.5 to UK in the amount of 11, Total is: 27.5

Now the program shows both currencies.

Lets convert from UK to US dollars now.

Accordin to GOOGLE the exchange rate is

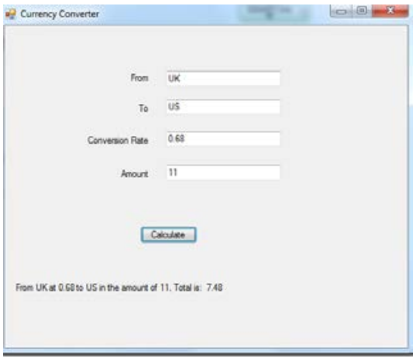

1 British Pound equals

1.46 US Dollar

1 US Dollar

0.68 British Pound

1 dollar is equals to 0.68 pounds. So let’s do the conversion of 11 dollars to pound.



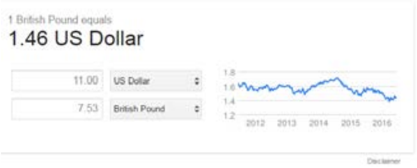
From: UK

To: US

Conversion Rate: 0.68

Amount: 11

From UK at 0.68 to US in the amount of 11, Total is: 7.48



1 British Pound equals

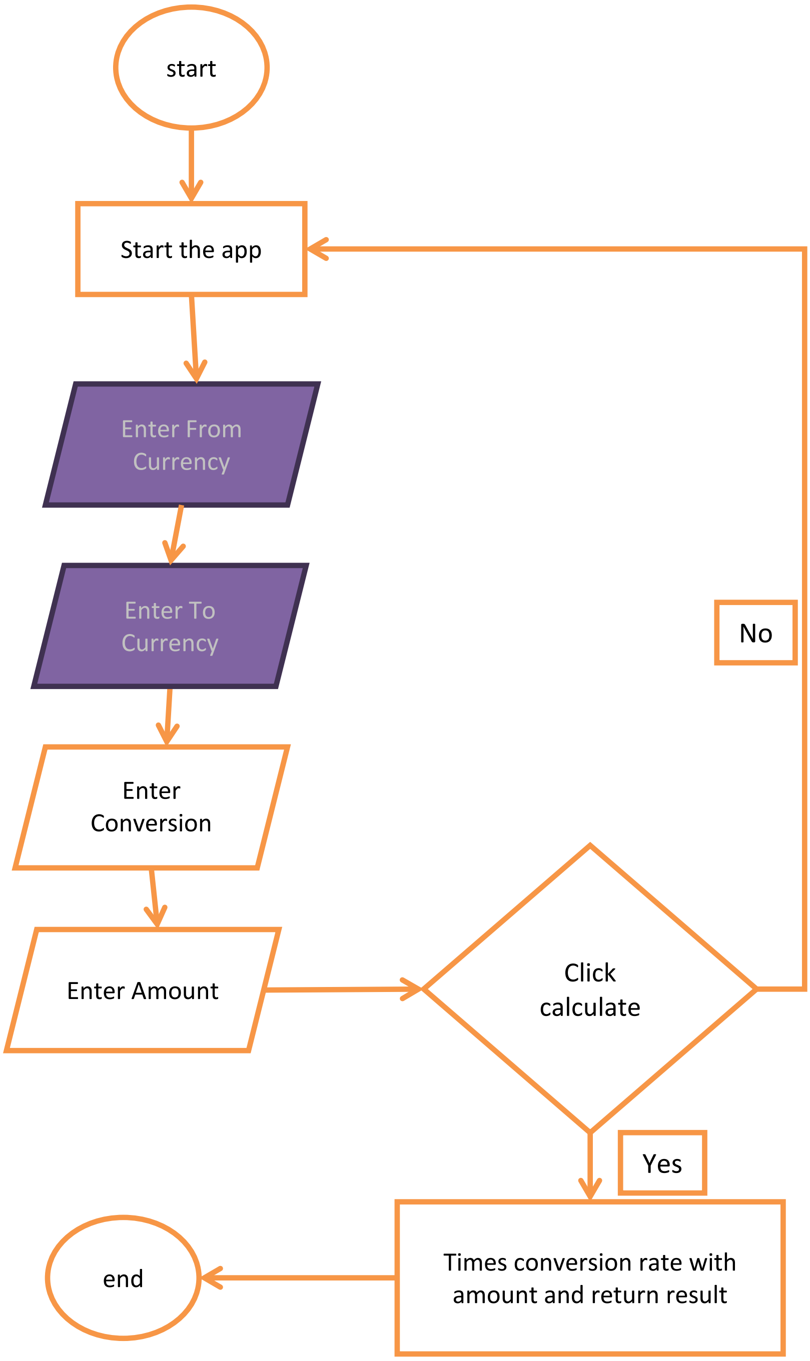
1.46 US Dollar

11.00 US Dollar

7.53 British Pound

Our program seems to be off by a few pence however you need to remember that the money exchange rates are calculating with several decimal points now just 2. So overall the program works and it outputs the right amount.

Lets look another flow chart with the new GUI settings.

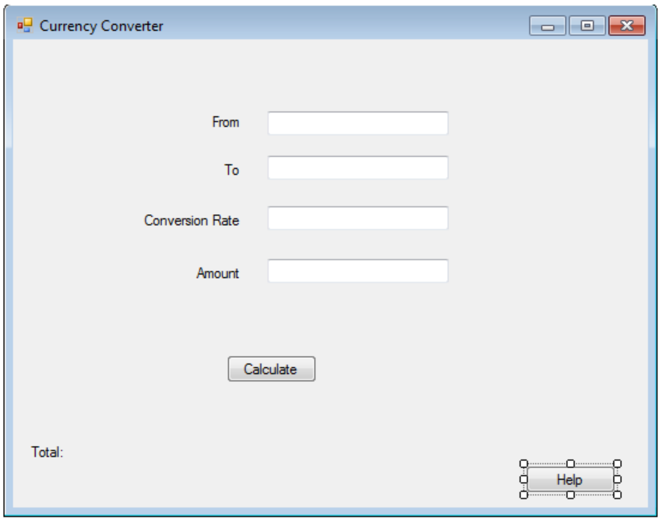


Its still the same as it was before only added the Enter from currency and Enter to currency. The overall program will work exactly as before.

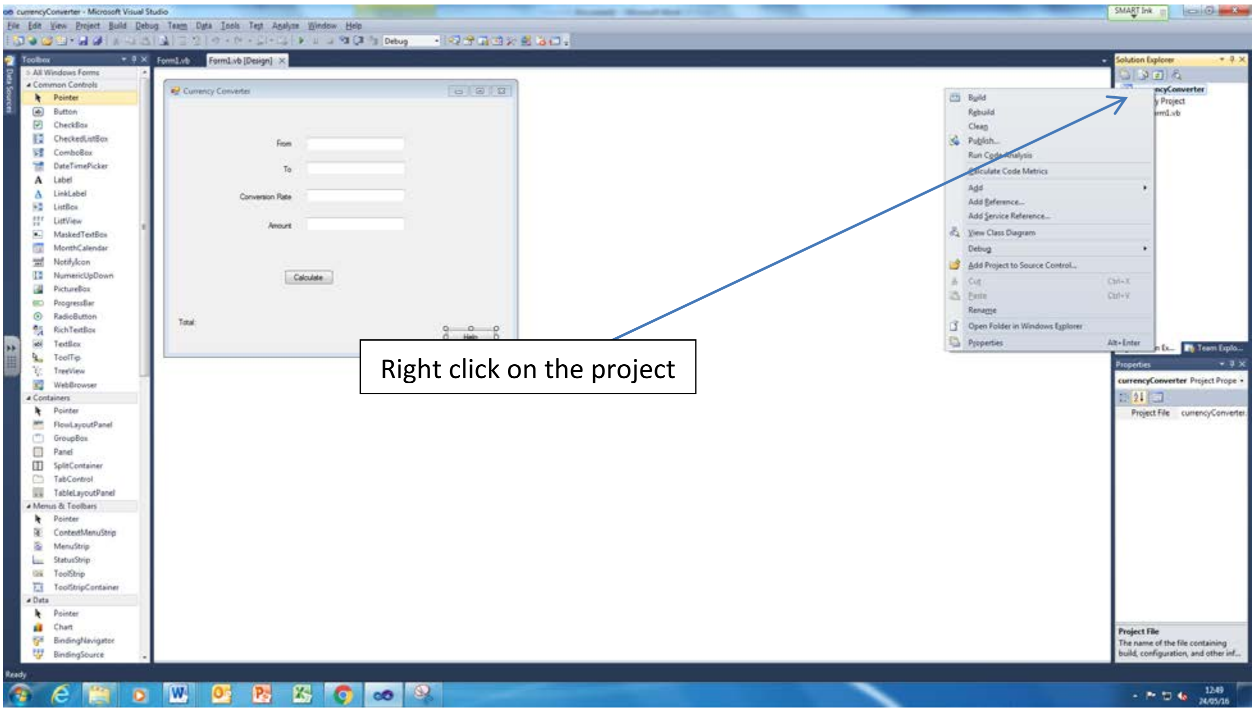
More tutorial on **www.mooict.com**

Now the program needs a help screen ☺

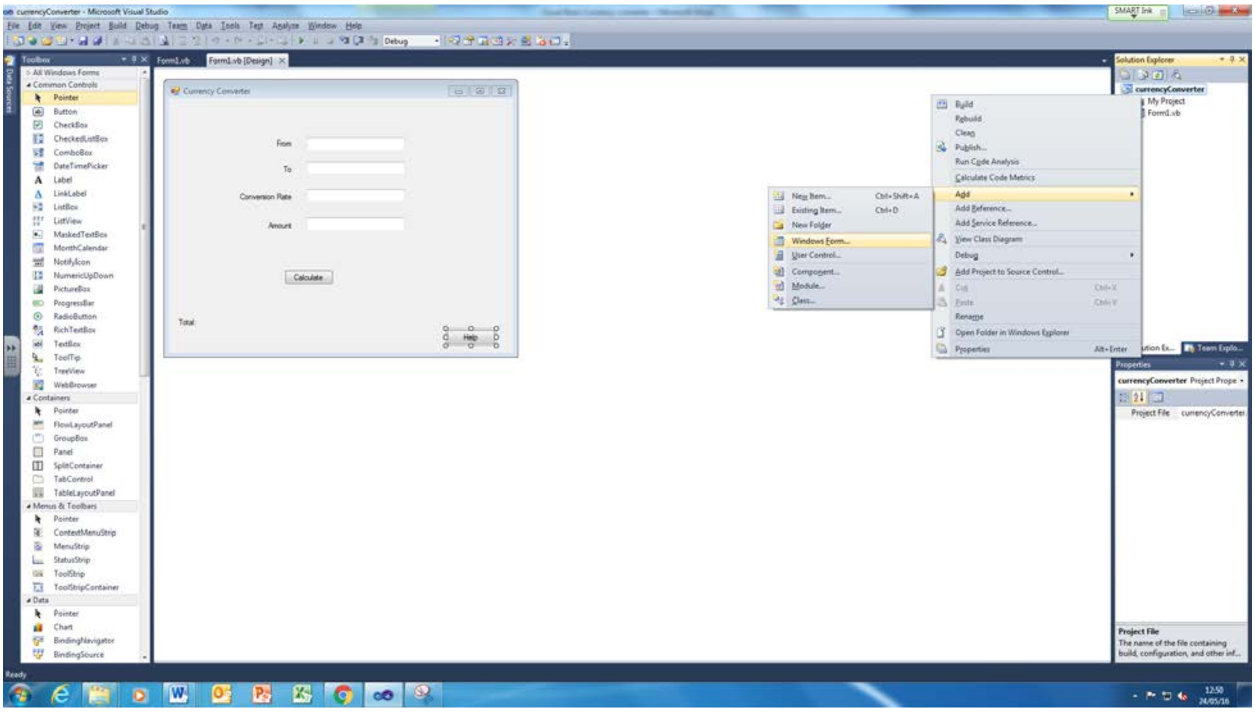
Every program needs a help screen. **Now time to add another button called help to the GUI.**



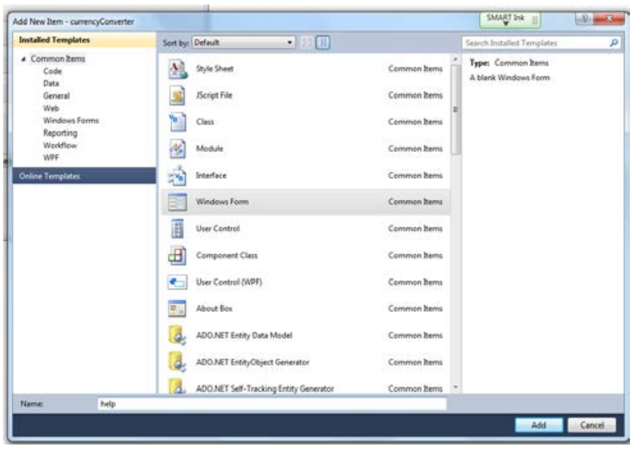
Also we need to add another FORM to the project.



Right click on the project

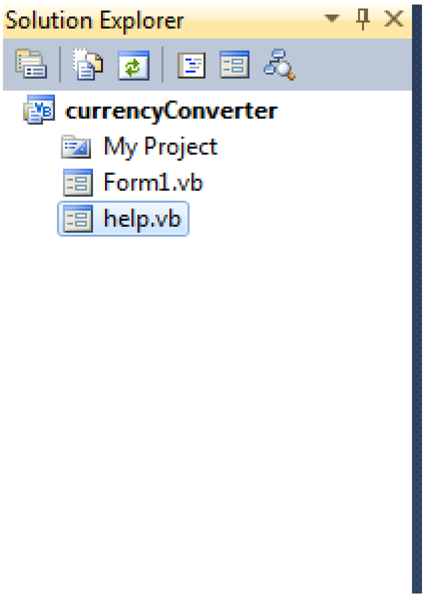


Add → Windows Form

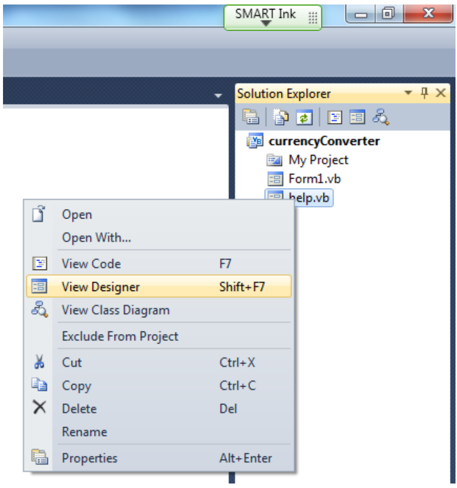


Name: help

Name it help and click on add

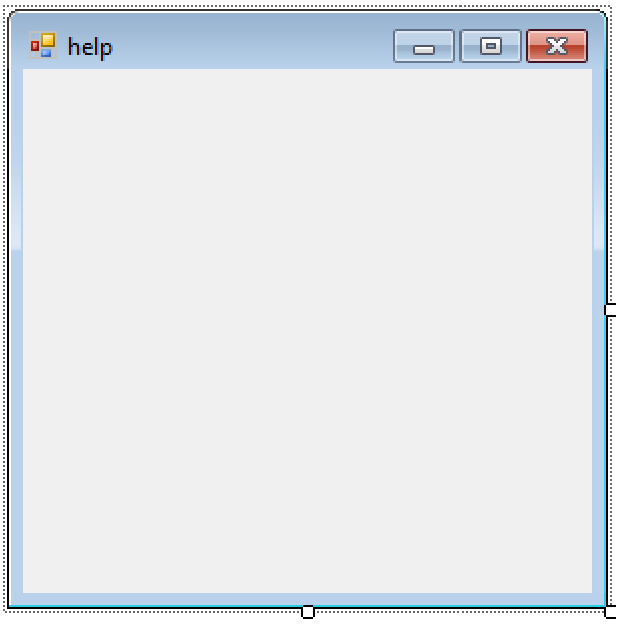


Inside your solution explorer there is help.vb file. That is another form just like the one we have been working on before. We will now right click on the help.vb



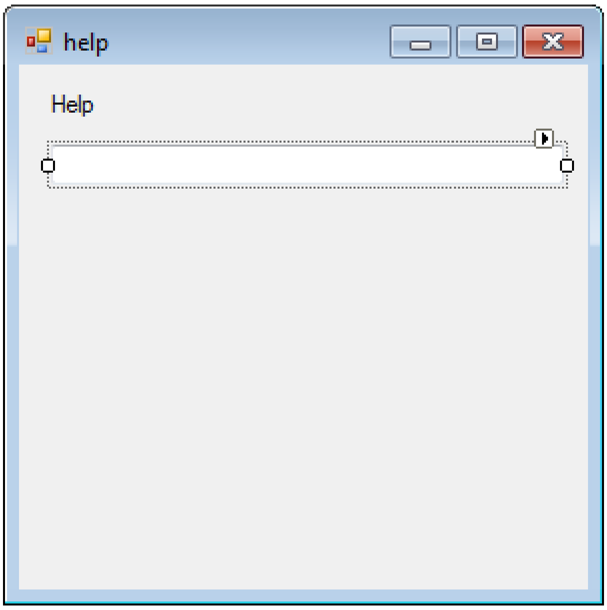
Click on view designer.

More tutorial on **www.mooict.com**

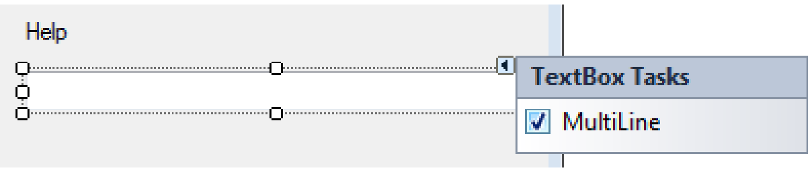


This is the new empty form. It's time to add some helpful tips here for the user right.

Let's add a **label** and call it help. And then add a **text box.**



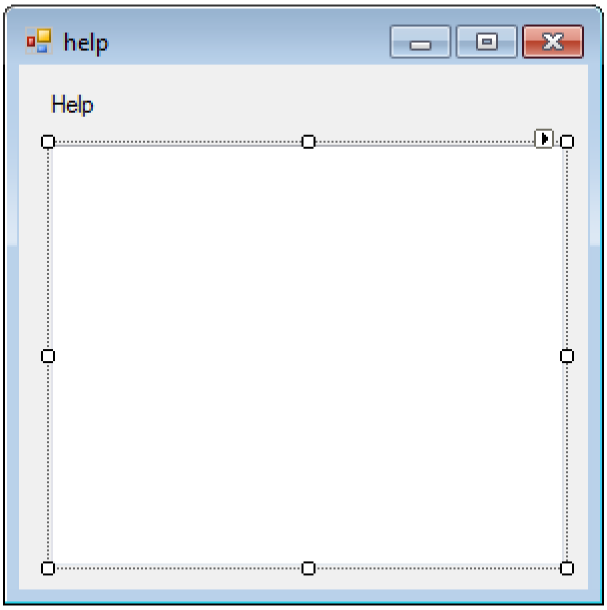
See that little triangle on the corner of the text box. Click on that and select Multiline.


Help

TextBox Tasks

☑ MultiLine

Instead of writing a line in different labels we can do it this way. Much simpler.

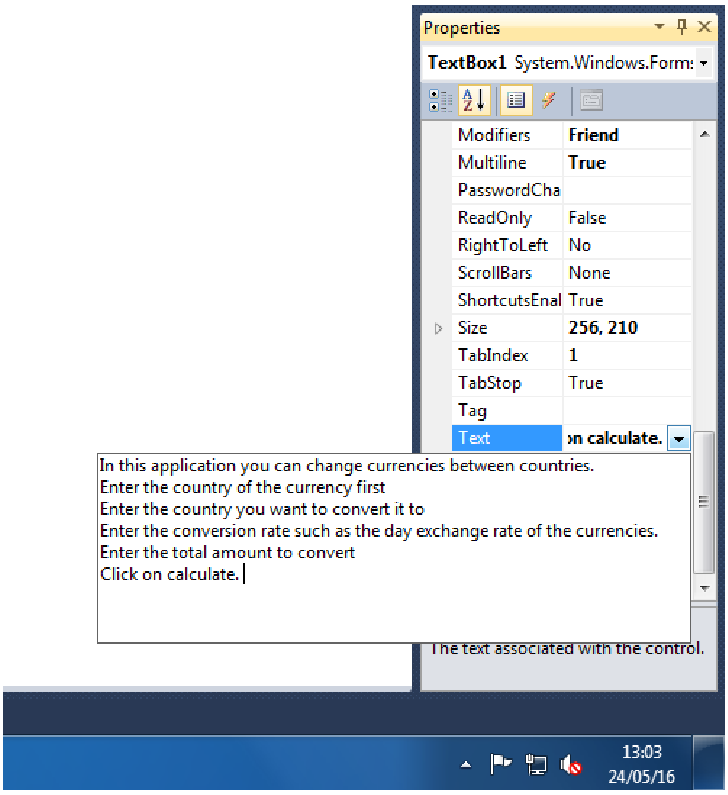


Now stretch it to fit the box.

Here is the help screen text.

In this application you can change currencies between countries.
Enter the country of the currency first
Enter the country you want to convert it to
Enter the conversion rate such as the day exchange rate of the currencies.
Enter the total amount to convert
Click on calculate.

Click on the textbox and check the properties window for Text option.



TextBox1 System.Windows.Form:

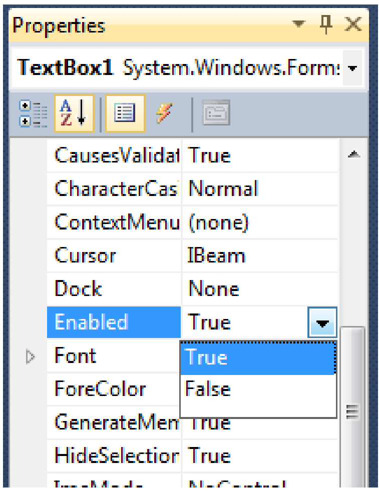
Modifiers	Friend
Multiline	True
PasswordCha	
ReadOnly	False
RightToLeft	No
ScrollBars	None
ShortcutsEnal	True
Size	256, 210
TabIndex	1
TabStop	True
Tag	
Text	on calculate.

In this application you can change currencies between countries.
Enter the country of the currency first
Enter the country you want to convert it to
Enter the conversion rate such as the day exchange rate of the currencies.
Enter the total amount to convert
Click on calculate.

The text associated with the control.

Now we need to change one more option in this which is ENABLED.

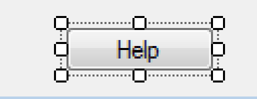
More tutorial on **www.mooict.com**



CausesValidat	True
CharacterCas	Normal
ContextMenu	(none)
Cursor	IBeam
Dock	None
Enabled	True
Font	True
ForeColor	False
GenerateMen	True
HideSelectior	True

Change the enabled from true to false because it disables the textbox and does not allow other users to edit the text.

Now time to link both forms together.



Double click on the help button.

```
  Private Sub Button2_Click(sender As System.Object, e As System.EventArgs) Handles Button2.Click


  End Sub
```

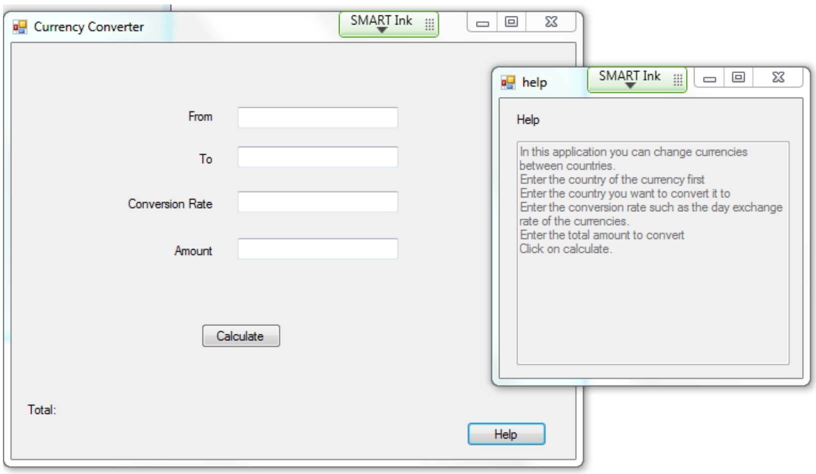
This code above will be added automatically. Now add the following code in the middle of the program.

```
    Private Sub Button2_Click(sender As System.Object, e As System.EventArgs) Handles
Button2.Click

Dim helpScreen As help = New help()
        helpScreen.Show()

    End Sub
```

Here we are creating a variable called helpScreen and then we are giving a data type of help. Remember when we called that form help right it's the same one. We are stating that this helpScreen should be a new instance of the help form which we created earlier. This way the program will make space in the memory for the new form to be deployed to the screen.

Secondly we are calling our newly created helpScreen variable and accessing the built in function called show.

Lets test the program now.



In this application you can change currencies between countries.
Enter the country of the currency first
Enter the country you want to convert it to
Enter the conversion rate such as the day exchange rate of the currencies.
Enter the total amount to convert
Click on calculate.

Okay then its all done and now go make any changes you want to.

If you have managed to get this far then

Don’t leave me hanging yo………