**Write to a Text File in VB .NET**

Writing to a text file is similar to reading a text file. Again we use System.IO. This time, instead of using the StreamReader we use the StreamWriter. The StreamWriter is used to write a stream of text to a file.

Add another Button to the form you've been working on. Set the Text property of the button to "Write to File". Double click your new button to open up the coding window. Add the following:

Dim FILE\_NAME As String = "E:\Text Files\test2.txt"

If System.IO.File.Exists(FILE\_NAME) = True Then

Dim objWriter As New System.IO.StreamWriter( FILE\_NAME )

objWriter.Write( TextBox1.Text )

objWriter.Close()

MessageBox.Show("Text written to file")

Else

MessageBox.Show("File Does Not Exist")

End If

Run your programme, and then click your new button.

Unless you have a file called "test2.txt", you should see the message box display: "File Does Not Exist."

Stop your programme and change this line:

Dim FILE\_NAME As String = "C:\Users\Owner\Documents\test2.txt"

To this:

Dim FILE\_NAME As String = "C:\Users\Owner\Documents\test.txt"

In other words, just change the file name back to test.txt. (Hopefully, you haven't deleted the test.txt file from your hard drive!)

Run your programme again. Type something into the textbox, and then click your button. You should see the message box "Text written to file" appear.

Once again, we check to see if the File Exists. If it's True that the file exists, then the first line that gets executed is setting up our variable:

Dim objWriter As New System.IO.StreamWriter(FILE\_NAME)

It's almost the same as last time. Only two things have changed: we created a new variable name, objWriter, and we're now using StreamWriter instead of StreamReader. Everything else is the same.

To write the text to our file, we've used this:

objWriter.Write( TextBox1.Text )

After the name of our variable (objWriter), we typed a full stop. The drop down box appeared showing available properties and methods. The "Write" method was chosen from the list. In between round brackets, you put what it is you want VB to write to your text file. In our case, this was the text in Textbox1. You can also do this:

objWriter.Write( "Your Text Here" )

The above uses direct text surrounded by double quotes. This is also acceptable:

Dim TheText As String = "Your Text Here"

objWriter.Write( TheText )

This time, we've put the text inside of a variable. The name of the variable is then typed inside of the round brackets of "Write".

But you don't have to write the whole text at once. You can write line by line. In which case, select WriteLine from the available properties and methods. Here's an example of how to use WriteLine:

Dim FILE\_NAME As String = ""E:\Text Files\test.txt"

Dim i As Integer

Dim aryText(4) As String

aryText(0) = "Mary WriteLine"

aryText(1) = "Had"

aryText(2) = "A"

aryText(3) = "Little"

aryText(4) = "One"

Dim objWriter As New System.IO.StreamWriter( FILE\_NAME )

For i = 0 To 4

objWriter.WriteLine( aryText(i) )

Next

objWriter.Close()

The error checking code has been left out here. But notice the new way to write text to the file:

objWriter.WriteLine( aryText(i) )

We're looping round and writing the contents of an array. Each line of text from the array gets written to our text file. But each line is appended. That is, the text file doesn't get erased after each line has been written. All the lines from the array will be written to the text file. However, if you were to run the code a second time then the contents of the file are erased before the new WriteLine() springs into action. In other words, you'd only get one version of "Mary WriteLine had a little one" instead of two.