**Reading a Text File Line by Line**

Quite often, you don't want to read the whole file at once. You want to read it line by line. In which case, instead of using the ReadToEnd method, as we did in the previous section, you can use the ReadLine method:

The ReadLine method, as its name suggests, reads text one line at a time. In order to do this, though, you need to use a loop. You can then loop round each line and read it into a variable. Here's a coding example:

Dim TextLine As String

Do While objReader.Peek() <> -1

TextLine = TextLine & objReader.ReadLine() & vbNewLine

Loop

The first line of the Do While loop is rather curious:

Do While objReader.Peek() <> -1

The Peek method takes a peek at the incoming text characters. It's looking ahead one character at a time. If it doesn't see any more characters, it will return a value of minus 1. This will signify the end of the text file. Our loop checks for this minus 1, and bails out when Peek has this value.

Inside the loop, we're reading each line from the text file and putting into new variable. (We're also adding a new line character on the end. Delete the & vbNewLine and see what happens).

objReader.ReadLine()

So the ReadLine method reads each line for you, instead of the ReadToEnd method which gets the whole of the text file.

Once you have a line of text in your variable, though, it's up to you to parse it. For example, suppose the line of text coming in from the text file was this:

"UserName1, Password1, UserName2, Password2"

You would then have to chop the line down and do something which each segment.

But what you are doing in the Do Loop is building up your variable with lines of text that are pulled from your text file. Once you have pulled all the text from your file, you can then put it into the text box. Here's our programme once more:

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

Dim TextLine As String

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

Dim objReader As New System.IO.StreamReader(FILE\_NAME)

Do While objReader.Peek() <> -1

TextLine = TextLine & objReader.ReadLine() & vbNewLine

Loop

Textbox1.Text = TextLine

Else

MessageBox.Show("File Does Not Exist")

End If

So inside the loop, we go round building up the TextLine variable. Once all the file has been read (when Peek() has a value of -1), we then place it into Textbox1.