CSC317 Visual Programming: Day 09

- Attendance/download Day09 from D2L
- PowerPoint with Illustrations:

We learn how to process text and image files. The first demo problem accepts a line of text into a TextBox and saves it to a file named *demo.txt* in whatever folder is selected by the user in yet another DialogBox – the *FolderBrowserDialog* object. The second demo problem accepts a picture file as a bitmap used for the image in a PictureBox, and it permits the user to save this image to a different file named demo.bmp.

Both problems use a *FolderBrowserDialog* object named fbd1, which is selected from the ToolBox and automatically placed in the component tray, *below* the form window. Its property *SelectedPath* is used to get the folder name, then concatenated, using the operator & with the predetermined file name ("\demo.txt" or "\demo.bmp") to get the complete file name.

New objects and methods:

- For a TextBox, the Clear method removes its text completely
- For a TextFile
    - System.IO.StreamReader object for input
        - System.IO.File.OpenText function
    - System.IO.StreamWriter object for output
        - System.IO.File.CreateText function
    - ReadLine gets a line of text from the file
        - The Peek function returns a -1 if there are NO MORE LINES after the last ReadLine
    - WriteLine writes a line of text to the file

- Demo Problems: TextFileSaveOpen,GetImage
- Practice Problem: TextAndImageFiles

CSC317 Visual Programming: Day 09
First Demo Project: TextFileSaveOpen
CODE

Notice the format required to associate this file name with a *StreamWriter* object, whose *WriteLine* method can be used to write a line of text to the file, and whose *Close* method should be used when finished, so that the file can be opened again, either by this project or another one. We allow for up to 3 lines of text, with a global declaration of 3 TextBoxes for saving lines and 3 TextBoxes for receiving lines. This is accomplished by declaring the arrays and then matching them with predesigned TextBoxes in the Form Load procedure, with index 0 being unused.

NOTE: the format for declaring a loop variable in the FOR statement.

```
Dim FolderName, FileName As String
Dim r(3), s(3) As TextBox 'do not use index 0

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        r(1) = Retrieve1 : r(2) = Retrieve2 : r(3) = Retrieve3
        s(1) = Save1 : s(2) = Save2 : s(3) = Save3
    End Sub

    Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnSave.Click
        If fbd1.ShowDialog = Windows.Forms.DialogResult.OK Then
            FolderName = fbd1.SelectedPath
            FileName = FolderName & "\demo.txt"
            Dim TextFile As System.IO.StreamWriter
            TextFile = System.IO.File.CreateText(FileName)
            For i As Integer = 1 To 3
                If s(i).Text <> "" Then TextFile.WriteLine(s(i).Text)
            Next
            TextFile.Close()
        End If

    End Sub
```

CSC317 Visual Programming: Day 09
First Demo Project: TextFileSaveOpen
CODE

Here we associate the file name with a *StreamReader* object, whose *ReadLine* method reads one line of text, and whose Close method is similarly used. If the user tries to open a non-existent file, an exception is caught. This example shows, that the system generates `ex.Message`, which can be displayed in a MessageBox, with customized choice of MessageBox buttons and icon.

```
 Private Sub btnRetrieve_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
btnRetrieve.Click
        For i As Integer = 1 To 3
            r(i).Clear()
        Next
        If fbd1.ShowDialog = Windows.Forms.DialogResult.OK Then
            Try
                FolderName = fbd1.SelectedPath
                FileName = FolderName & "\demo.txt"
                Dim TextFile As System.IO.StreamReader
                TextFile = System.IO.File.OpenText(FileName)
                'might not have 3 lines in file, so need to use Peek method
                Dim i = 1
                Do Until TextFile.Peek = -1 ‘use this instead of FOR loop!
                    r(i).Text = TextFile.ReadLine
                    i += 1
                Loop
                TextFile.Close()
            Catch ex As Exception
                MessageBox.Show(ex.Message, "TextFileSaveOpen", MessageBoxButtons.OK,
MessageBoxIcon.Error)
            End Try

        End If

    End Sub
```

CSC317 Visual Programming: Day 09
Second Demo Project: GetImage
CODE

For picture files, we also use an *OpenFileDialog* object named *ofd1*. It has a property *FileName*, which is used as a parameter to Bitmap, which creates a new Image object to be placed in our PictureBox picImage. We also use a second PictureBox picImage, which has the blank file from the Images folder put into its image at the design phase.

```
 Private Sub picImage_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles picImage.Click
        Dim FileName As String
        Dim result As DialogResult = ofd1.ShowDialog
        If result = Windows.Forms.DialogResult.OK Then
            FileName = ofd1.FileName
            picImage.Image = New Bitmap(FileName)
        End If
    End Sub

    Private Sub btnClear_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles btnClear.Click
        picImage.Image = picBlank.Image
    End Sub
```

CSC317 Visual Programming: Day 09
Second Demo Project: GetImage
CODE

The code for saving an image uses a *Stream* object *ImageFile*, which is associated with the file name by using the Create method in the System.IO.File class. *ImageFile* has a close method. Unlike text files, picture files are saved all at once, not line by line:

```
picImage.Image.Save(ImageFile,System.Drawing.Imaging.ImageFormat.Bmp)
```

This format is complicated and should be used as reference, rather than being memorized!

NOTE: an exception is caught if you do not save the image to a NEW file! You cannot save to the file used for getting the image in the first place because it was never closed!

Why was it never closed? Good question!

CHALLENGE: If you can figure out how to close a bitmap file after opening it, you will be given a bonus of 2 points toward your final grade! Dr. Fabrey cannot find the solution, nor have 3 earlier CSC317 classes found it! The blogosphere has few theories about it, and they all seem to fail…

CSC317 Visual Programming: Day 09
Second Demo Project: GetImage
CODE

```
 Private Sub btnSave_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnSave.Click
        If fbd1.ShowDialog = Windows.Forms.DialogResult.OK Then
            Try
                Dim FolderName As String = fbd1.SelectedPath
                Dim FileName As String = FolderName & "\demo.bmp"
                Dim ImageFile As System.IO.Stream
                 ImageFile = System.IO.File.Create(FileName)
                picImage.Image.Save(ImageFile,
System.Drawing.Imaging.ImageFormat.Bmp)
                  ImageFile.Close()
            Catch ex As Exception
                 MessageBox.Show("Must save image to a new file!", "Text
And Image Files")
            End Try

        End If
    End Sub
```

CSC317 Visual Programming: Day 09
Practice Project: TextAndImageFiles

This is a combination of the two demo projects.

Have fun!