File Hashing

These guidelines show how implement file hashing feature in your applications using tools available in .NET platform.

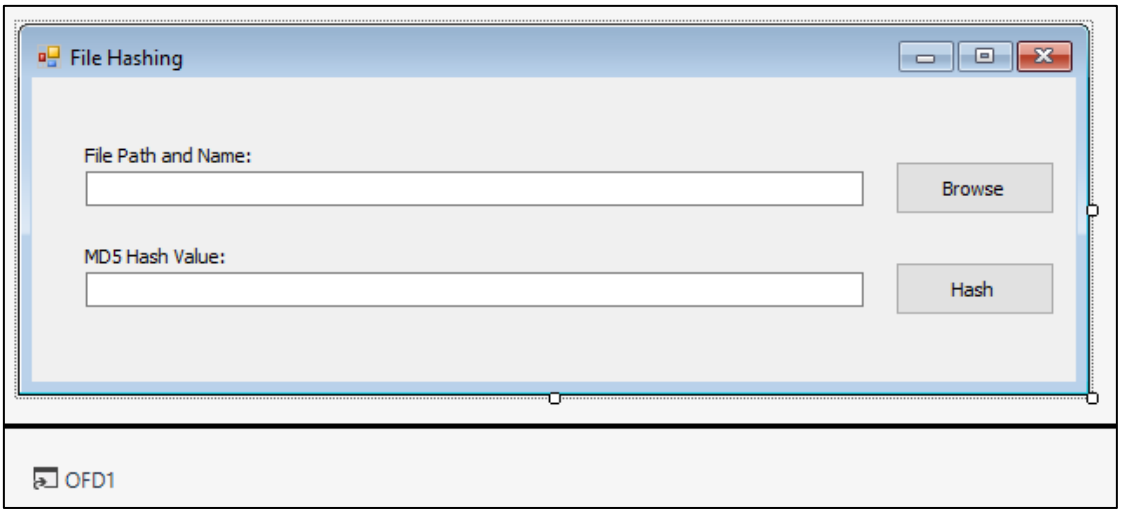
Time needed to accomplish this task: 20 - 30 minutes

GUI

- Create a new VB.NET project "Windows Form Application", change the project name and path according to your preferences.
- Create a new windows form (if it is not created by default), name it "frmHash", and change its caption/text to "File Hashing"
- In "frmHash" form, create 2 textboxes, 2 buttons, 2 labels and 1 OpenFileDialog. Change the properties of these components as follows:

Old Name	New Name	Caption/Text	Other Properties
TextBox1	txtFilePath		
TextBox2	txtHashValue		
Button1	btnBrowse	Browse	
Button1	btnHash	Hash	
Label1	lblPath	File Path and Name:	
Label2	lblHash	MD5 Hash Value:	
OpenFileDialog1	OFD1		

- By the end of this step you should have something looks like the following picture:



Now… Let's do some coding…

Coding

- At the beginning, let's do some imports. Add these lines to the beginning of your "frmEncr.vb" file

```
Imports System
Imports System.IO
Imports System.Text
Imports System.Security.Cryptography
```

Browsing for the file:

To select the file that you want to hash, we will use "OpenFileDialog" tool that you can find it in the Toolbox under "Dialogs" subcategory.

- Drag and drop "OpenFileDialog" into your form. You will notice that this tool is added to under the divider below your form.
- Change the name of this tool from "OpenFileDialog1" to "OFD1"
- Double click on btnBrowse button. On btnBrowse_Click event, write the following code

```
OFD1.ShowDialog()
txtFilePath.Text = OFD1.FileName
```

Note: The code above will show the OpenFileDialog "OFD1" to the user and let the user browse and select the file. When a file is selected, the filename (with the full path) will be returned and displayed on txtFilePath textbox.

Since the file is selected successfully, now it is time to do the real work…

Hashing

To do the hashing of the file, we will use the services provided by .NET platform.
The file hashing is carried out as follows:

- In the *btnHash_Click* sub write the following code:

```
Dim path As String = txtFilePath.Text

   Try
        Using reader As New System.IO.FileStream(path, IO.FileMode.Open,
IO.FileAccess.Read)
            Using md5 As New System.Security.Cryptography.MD5CryptoServiceProvider
                Dim hash() As Byte = md5.ComputeHash(reader)
                txtHashValue.Text = ByteArrayToString(hash)
            End Using
        End Using
    Catch ex As Exception
        txtHashValue.Text = ""
    End Try
```

Note: Look into the previous code and understand how it is used to generate the MD5 hash value of the file. Compare what is done here with what we have done before in file encryption procedure.

- The *ByteArrayToString* mentioned in the above code is used to convert Byte data in to string (text) data and it is defined as follows:

```
Public Function ByteArrayToString(ByVal arrInput() As Byte) As String
    Try
        Dim sb As New System.Text.StringBuilder(arrInput.Length * 2)
        For i As Integer = 0 To arrInput.Length - 1
           sb.Append(arrInput(i).ToString("X2"))
        Next
        Return sb.ToString().ToLower
    Catch ex As Exception
        Return ""
    End Try
End Function
```

- Write the above *ByteArrayToString* function code in the Public Class code as shown below:

```
Public Class frmHash

    Public Function ByteArrayToString(ByVal arrInput() As Byte) As String
        ...
        ...
    End Function
    ...
```

Testing the System for the file:

- Test the system capabilities and limitations by hashing different files in terms of formats, sizes, contents, storage medium,… etc.
- Try to modify the current project (or create another project) to hash a given text input (not a file), like a name or a password. What are the changes/modifications that you need to make?
- Suppose that you decided to use another hashing algorithm such as SHA1 or any other hash function instead of MD5, what are the modifications that you need to do? Try to implement this hash function in your project.
- The procedure above is only used to generate/calculate hash values for a given file. However, the optimum use of hashing techniques involves another step to verify and compare a calculated hash value for a file with the hash value received/stored with that file. Create another form to compare and verify the integrity of a given file be comparing the received hash value with the calculated hash value for that file.