Content

FILE HANDLING IN VB.NET

FILE HANDLING IN VB.NET

File Info class

It is Provides properties and instance methods for the creation, copying, deletion, moving, and opening of files, and aids in the creation of FileStream objects.

File info class is present in the IO package. Whenever we call fileinfo class method need to import namespace like this

```
Imports System
Imports System.IO
```

Properties:

Main properties of this class to get file information like file extension, last access time, Full name etc.

CreationTime : Gets or sets the creation time of the current file or directory.

Directory : Gets an instance of the parent directory.

DirectoryName : Gets a string representing the directory's full path.

Exists : Gets a value indicating whether a file exists.

Extension : Gets the string representing the extension part of the file.

FullName : Gets the full path of the directory or file.

IsReadOnly : Gets or sets a value that determines if the current file is read only.

LastAccessTime : Gets or sets the time the current file or directory was last accessed.

LastWriteTime : Gets or sets the time when the current file or directory was last written to.

Length : Gets the size, in bytes, of the current file.

Name : Gets the name of the file.

Methods

Below File Info class methods we are frequently used in .NET

CopyTo(String): Copies an existing file to a new file, disallowing the overwriting of an existing file.

CopyTo(String, Boolean) : Copies an existing file to a new file, allowing the overwriting of an existing file.

Create : Creates a file.

CreateText: Creates a StreamWriter that writes a new text file.

Delete : Permanently deletes a file.

MoveTo : Moves a specified file to a new location, providing the option to specify a new file name.

Open(FileMode): Opens a file in the specified mode.

Open(FileMode, FileAccess) : Opens a file in the specified mode with read, write, or read/write access.

Open(FileMode, FileAccess, FileShare) : Opens a file in the specified mode with read, write, or read/write access and the specified sharing option.

OpenRead : Creates a read-only FileStream.

OpenWrite : Creates a write-only FileStream.

File Stream class

Exposes a Stream around a file, supporting both synchronous and asynchronous read and write operations.

Properties

CanRead: Gets a value indicating whether the current stream supports reading.

CanWrite: Gets a value indicating whether the current stream supports writing.

Length : Gets the length in bytes of the stream.

Name : Gets the name of the FileStream that was passed to the constructor.

Position : Gets or sets the current position of this stream.

ReadTimeout : Gets or sets a value, in miliseconds, that determines how long the stream will attempt to read before timing out.

WriteTimeout : Gets or sets a value, in miliseconds, that determines how long the stream will attempt to write before timing out.

Methods:

BeginRead : Begins an asynchronous read operation.

BeginWrite : Begins an asynchronous write operation

Close : Closes the current stream and releases any resources (such as sockets and file handles) associated with the current stream. Instead of calling this method, ensure that the stream is properly disposed.

CopyTo(Stream) : Reads the bytes from the current stream and writes them to another stream.

CopyTo(Stream, Int32) : Reads the bytes from the current stream and writes them to another stream, using a specified buffer size.

Dispose : Releases all resources used by the Stream.

Dispose(Boolean) : Releases the unmanaged resources used by the FileStream and optionally releases the managed resources.

Flush : Clears buffers for this stream and causes any buffered data to be written to the file.

Flush(Boolean) : Clears buffers for this stream and causes any buffered data to be written to the file, and also clears all intermediate file buffers.

Lock : Prevents other processes from reading from or writing to the FileStream.

Read : Reads a block of bytes from the stream and writes the data in a given buffer.

ReadByte : Reads a byte from the file and advances the read position one byte.

Seek : Sets the current position of this stream to the given value.

Unlock : Allows access by other processes to all or part of a file that was previously locked.

Write : Writes a block of bytes to the file stream.

WriteByte : Writes a byte to the current position in the file stream.

FileAccess Enumerator

It is used to define constants for read, write, or read/write access to a file.

Members

Read: Read access to the file. Data can be read from the file. Combine with Write for read/write access.

Write: Write access to the file. Data can be written to the file. Combine with Read for read/write access.

ReadWrite: Read and write access to the file. Data can be written to and read from the file.

Example:

The following FileStream constructor grants read-only access to an existing file (FileAccess.Read).

```
Dim s2 As New FileStream(name, FileMode.Open, FileAccess.Read,
FileShare.Read)
```

In the example we set FileAccess enumeration as Read so only able to read file not able to alter text etc.

How to create new File in VB.NET?

You can create file using File.Create method like using below code

```
'Create File in VB.NET
    Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button1.Click
        Dim path As String = String.Empty
        'create new text file so we use extension as
.txt
        path = "D:\\test_" &
DateTime.Now.ToString("ddMMyyyy_HHmmss") & ".txt"
        File.Create(path)
        MsgBox("File Successfully created in this path"
& path)
    End Sub
```

How to create File and write text that file in VB.NET?

Using Stream writer function we can write lines in the existing file

```
'CREATE FILE AND WRITE TEXT IN THAT FILE
    Private Sub Button2_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button2.Click
        Dim path As String = String.Empty
        path = "D:\\test_" &
DateTime.Now.ToString("ddMMyyyy_HHmmss") & ".txt"

        Dim txt As String = "Hi !!! How are you??"
        Dim sw As StreamWriter
        sw = File.CreateText(path)
        sw.WriteLine(txt)
        sw.Close()
        MsgBox("Create new file and write
Successfully!! Path is " & path)
    End Sub
```

How to read all content (all lines) from a file in VB.NET?

Using Stream Reader ReadtoEnd function we can read all lines from start to end like below example.

```
'READ ALL CONTENT FROM TEXT FILE
    Private Sub Button3_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button3.Click
        Dim path As String = String.Empty
        'Read file from Application executable path
mean i have place text file in bin directory
        path =
System.Windows.Forms.Application.ExecutablePath.ToStrin
g()
        path = Mid(path, 1, path.LastIndexOf("\")) &
"\test.txt"

        'Read File content
        Dim sr As StreamReader
        Try
            sr = File.OpenText(path)
            label1.Text = sr.ReadToEnd()
            sr.Close()
        Catch ex As Exception
            MessageBox.Show(ex.ToString())
        End Try
    End Sub
```

How to read line by line content from a file in VB.NET?

Using Stream Reader ReadLine() function we can read each lines content and write in label or somewhere.

```
'READ CONTENT FROM TEXT FILE ONE BY ONE
    Private Sub Button4_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button4.Click
        Label1.Text = ""
        Dim path As String = String.Empty
        path =
System.Windows.Forms.Application.ExecutablePath.ToStrin
g()
        path = Mid(path, 1, path.LastIndexOf("\")) &
"\test2.txt"

        Dim obj As StreamReader
        Dim strLine1 As String

        obj = New StreamReader(path)

        Do While obj.Peek() <> -1
            'Read Line
            strLine1 = obj.ReadLine()
            Label1.Text += strLine1 & vbCrLf
        Loop

    End Sub
```

How to add new line in existing file content in VB.NET?

Using Append function we can add new line with existing content.

```
'OPEN EXISTING FILE AND  WRITE
    Private Sub Button7_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button7.Click
        Dim path As String = String.Empty
        path =
System.Windows.Forms.Application.ExecutablePath.ToStrin
g()
        path = Mid(path, 1, path.LastIndexOf("\")) &
"\test3.txt"

        If File.Exists(path) Then
            Dim oFile As System.IO.File
            Dim oWrite As System.IO.StreamWriter
            oWrite = File.AppendText(path)
            oWrite.WriteLine(vbCrLf & Today & " " &
TimeString & " " & "Appended Text")
            oWrite.Close()
            MsgBox("Successfully new line append with
existing content")
        Else
```

```
            MsgBox("File not present")
        End If
    End Sub
```

File Info Class Example1

In this example we get file information using file info class and display in the label.

```
'EXAMPLE OF FILE INFO CLASS
    Private Sub Button5_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button5.Click
        Label1.Text = ""
        Dim path As String = String.Empty
        path =
System.Windows.Forms.Application.ExecutablePath.ToStrin
g()
        path = Mid(path, 1, path.LastIndexOf("\")) &
"\test2.txt"
        Dim finfo As New FileInfo(path)

        Label1.Text += " Name : " & finfo.Name & vbCrLf
        Label1.Text += "Full Name : " & finfo.FullName
& vbCrLf
        Label1.Text += "Creation Time : " &
finfo.CreationTime & vbCrLf
        Label1.Text += "Directory : " &
finfo.Directory.ToString() & vbCrLf
        Label1.Text += "File Exist : " &
finfo.Exists.ToString() & vbCrLf
        Label1.Text += "Extension : " & finfo.Extension
& vbCrLf
        Label1.Text += "Last AccessTime : " &
finfo.LastAccessTime & vbCrLf
        Label1.Text += "Last WriteTime : " &
finfo.LastWriteTime & vbCrLf
```

```
    End Sub
```

File Info Class Example2

In this example we can move or copy files from one directory to other etc.

```
'EXAMPLE OF FILE INFO CLASS
    Private Sub Button6_Click(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
Button6.Click
        Dim path As String = String.Empty
        Dim path1 As String = String.Empty
        path =
System.Windows.Forms.Application.ExecutablePath.ToStrin
g()
        path = Mid(path, 1, path.LastIndexOf("\")) &
"\test4.txt"
        path1 = Mid(path, 1, path.LastIndexOf("\")) &
"\test5.txt"

        Dim finfo As New FileInfo(path)
        Dim finfoMove As New FileInfo(path1)

        If finfo.Exists Then
            'Copy that file and placed under D:\\
            finfo.CopyTo("D:\\" & finfo.Name)

            'Copy that file and placed under D:\\ and
that file is already present in that location so i set
overwrite is true
            finfo.CopyTo("D:\\" & finfo.Name, True)

            'Delete that file after copy
            finfo.Delete()
            MsgBox("File is successfully copy and
delete")
        Else
            MsgBox("No file present in that location "
& path)
        End If
        If finfoMove.Exists Then
            'Move file from one to another location
            finfoMove.MoveTo("D:\\" & finfoMove.Name)
            MsgBox("File is successfully moved!")
        Else
            MsgBox("No file present" & path1)
        End If

    End Sub
```