

**ជំពូកទី៣ :**

# **ការប្រើប្រាស់ File System Object**

**នៅក្នុងមេរៀននេះលោកអ្នកនឹងធ្វើការសិក្សាពី:**

- 👉 របៀបប្រើប្រាស់ FileSystem Methods & Properties
- 👉 របៀបប្រើ TextStream Object's Methods
- 👉 របៀបប្រើ Text Stream Object's Properties
- 👉 របៀបធ្វើការជាមួយ Files និង Folders Collection

កាលពីជំពូកទី២ លោកអ្នកបានសិក្សាពី File រួចមកហើយ ។ នៅក្នុងជំពូកនេះ លោកអ្នកនឹងធ្វើការសិក្សាពី File ម្តងទៀតដោយប្រើនូវ Object មួយដែលមានឈ្មោះថា File System Object ។ ចំពោះ File System Object នេះគឺវាផ្តល់នូវ Properties និង Methods ជាច្រើនដែលអនុញ្ញាតអោយលោកអ្នកកាន់តែងាយស្រួលក្នុងការធ្វើការជាមួយ Files ព្រមទាំង Folders ដែលមាននៅក្នុងកុំព្យូទ័រ របស់លោកអ្នក ឬក៏ កុំព្យូទ័រផ្សេងទៀតដែលបានភ្ជាប់បណ្តាញ Network ជាមួយលោកអ្នក។ មុននឹងលោកអ្នក អាចប្រើ File System Object បានលោកអ្នកត្រូវចូលទៅផឹកយក References មួយដែលមានឈ្មោះថា Microsoft Scripting Run Time ជាមុនសិន ឬ លោកអ្នកមិនចាំបាច់ចូលទៅផឹកក៏បាន តែភាសា Visual Basic ទាមទារអោយលោកអ្នកត្រូវប្រើនូវអនុគមន៍មួយដែលមានឈ្មោះថា CreateObject ដូចខាងក្រោម :

```
Dim FSys As Object
Set FSys = CreateObject(" Scripting.FileSystemObject")
```

# I. របៀបប្រើប្រាស់ FileSystemObject's Methods & Properties

## I.1. CreateFolder

ត្រូវបានប្រើសំរាប់បង្កើត Folder ។  
Syntax:

```
FSys.CreateFolder (Path As String ) As Folder
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit
Dim FSys As New FileSystemObject
```

---

```
Private Sub Form_Load()
    FSys.CreateFolder "C:\BBU"
End Sub
```

---

## I.2. CreateTextFile

ត្រូវបានប្រើសំរាប់បង្កើត Text File ។

Syntax:

```
FSys.CreateTextFile (Filename As String, [Overwrite  
As Boolean] )
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit  
Dim FSys As New FileSystemObject
```

---

```
Private Sub Form_Load()  
    FSys.CreateTextFile ("C:\BBU\test1.txt")  
End Sub
```

---

## I.3. CopyFile

ត្រូវបានប្រើសំរាប់ចំលង File ទៅដាក់ទីតាំងណាមួយរបស់ Hard Disk ។

Syntax:

```
FSys.CopyFile (Source As String, Destination As  
String, [Overwrite As Boolean=True] )
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit  
Dim FSys As New FileSystemObject
```

---

```
Private Sub Form_Load()  
    FSys.CopyFile "C:\BBU\test1.txt", "C:\"
```

---

End Sub

---

## I.4. CopyFolder

ត្រូវបានប្រើសំរាប់ចម្លង Folder ព្រមទាំង Sub Folder និង File ដែលមាននៅក្នុង Folder នោះពីទីតាំងមួយទៅដាក់ក្នុងទីតាំងមួយ។

Syntax:

```
FSys.CopyFolder(Source As String, Destination As String, [Overwrite As Boolean=True] )
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit  
Dim FSys As New FileSystemObject
```

---

```
Private Sub Form_Load()  
    FSys.CopyFolder "C:\BBU", "D:\"  
End Sub
```

---

## I.5. DeleteFile

ត្រូវបានប្រើសំរាប់លុប File ចេញពី Hard Disk ។

Syntax:

```
FSys.DeleteFile(FileSpec As String, [Force As Boolean=False] )
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit
```

```
Dim FSys As New FileSystemObject
```

```
Private Sub Form_Load()  
    FSys.DeleteFile "C:\BBU\Test1.txt"  
End Sub
```

## I.6. DeleteFolder

ត្រូវបានប្រើសំរាប់លុប Folder ចេញពី Hard Disk ។

Syntax:

```
FSys.DeleteFolder(FileSpec As String, [Force As Boolean=False] )
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit  
Dim FSys As New FileSystemObject  
  
Private Sub Form_Load()  
    FSys.DeleteFolder ("D:\BBU")  
End Sub
```

## I.7. DriveExists

ត្រូវបានប្រើសំរាប់ត្រួតពិនិត្យមើលថាតើមាន Drive ណាមួយដូចដែល  
ដែលលោកអ្នកបានបោះអោយនោះ។ ប្រសិនបើមាននោះអនុគមន៍ទទួលបានតំលៃ True  
ផ្ទុយទៅវិញអនុគមន៍ទទួលបានតំលៃ False ។

Syntax:

```
FSys.DriveExists (DriveSpec As String) As Boolean
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit
Dim FSys As New FileSystemObject
```

```
Private Sub Form_Load( )
    MsgBox FSys.DriveExists("G:\")
End Sub
```

## I.8. Drives Property

គឺជា Collection ដែលផ្តុំក Drive ទាំងអស់ ។

**ឧទាហរណ៍ទី១ :**

☞ Source Code:

```
Option Explicit
Dim FSys As New FileSystemObject
```

```
Private Sub Form_Load( )
    Dim i%
    Dim D As Drive
    For Each D In FSys.Drives
        MsgBox D.DriveLetter
    Next
End Sub
```

**ឧទាហរណ៍ទី២ :**

☞ Source Code:

```
Option Explicit
Dim FSys As New FileSystemObject
```

```
Private Sub Form_Load( )
    Dim i%
    Dim D As Drive
    Call AddColumnHeader(Me.ListView1, "Drive", "DriveType", "FileSystem",
        "Total Size", "SerialNumber")
    For Each D In FSys.Drives
```

On Error Resume Next  
 Call AddDataToLvw(Me.ListView1, D.DriveLetter, D.DriveType,  
 D.FileSystem, D.RootFolder, D.SerialNumber)

Next  
 End Sub

**លទ្ធផល :**

| Drive | DriveType | FileSystem | Total Size  | SerialNumber |
|-------|-----------|------------|-------------|--------------|
| C:\   | 2         | NTFS       | 20374428160 | 281673944    |
| D:\   | 2         | NTFS       | 10487197696 | -56038399    |
| F:\   | 2         | NTFS       | 9631768576  | 1087358730   |
| G:\   | 1         | FAT        | 128067584   | 35424        |

## I.9. FileExists

ត្រូវបានប្រើសំរាប់ត្រួតពិនិត្យមើលថា តើមាន File នៅក្នុង Hard Disk ដែរឬទេ។  
 ប្រសិនបើមានអនុគមន៍ទទួលបានតំលៃ True ផ្ទុយទៅវិញអនុគមន៍ទទួលបានតំលៃ False ។

Syntax:

```
FSys.FileExists (FileSpec As String) As Boolean
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit  

Dim FSys As New FileSystemObject
```

```
Private Sub Form_Load()  

  If FSys.FileExists("C:\BBU\Test1.txt") = True Then  

    MsgBox "File already have"  

  Else  

    MsgBox "File Not Found"
```

End If  
End Sub

---

## I.10. FolderExists

ប្រើសំរាប់ត្រួតពិនិត្យមើលថា តើមាន Folder ណាមួយនៅក្នុង Hard Disk ដែរឬទេ។

Syntax:

```
FSys.FolderExists (FoderSpec As String) As Boolean
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit  
Dim FSys As New FileSystemObject
```

---

```
Private Sub Form_Load()  
    If FSys.FolderExists("C:\BBU\") = True Then  
        MsgBox "Foler's already have"  
    Else  
        MsgBox "Folder's Not Found"  
    End If  
End Sub
```

---

## I.11. GetAbsolutePathName

ប្រើសំរាប់ទទួលយក Path ដែលលោកអ្នកកំពុងបិទនៅ។

Syntax:

```
FSys.GetAbsolutePathName (Path As String) As String
```

**ឧទាហរណ៍ :**

☞ Source Code:



```
Option Explicit
Dim FSys As New FileSystemObject
```

```
Private Sub Form_Load()
    MsgBox FSys.GetAbsolutePathName("BBU")
End Sub
```

## I.12. GetDrive

ប្រើសំរាប់ពោះ Drive របស់ Hard Disk ទៅអោយអថេរជា Drive Object ។

Syntax:

```
Dim D As Drive
Set D = FSys.GetDrive (DriveSpec As String)
```

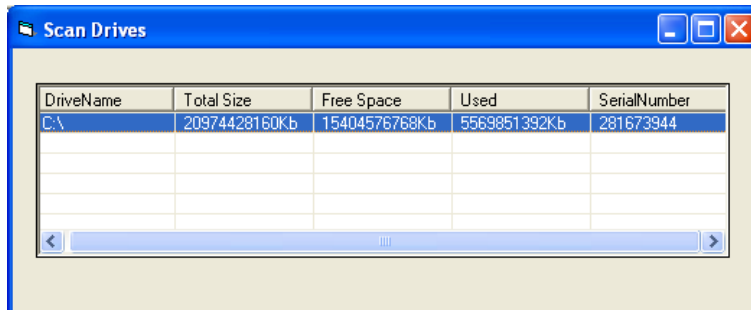
**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit
Dim FSys As New FileSystemObject
```

```
Private Sub Form_Load()
    Dim d As Drive
    Set d = FSys.GetDrive("C:")
    Call AddColumnHeader(Me.ListView1, "DriveName", "Total Size", "Free
        Space", "Used", "SerialNumber")
    Call AddDataToLvw(Me.ListView1, d.RootFolder, d.TotalSize & "Kb",
        d.FreeSpace & "Kb", (d.TotalSize - d.FreeSpace) & "Kb",
        d.SerialNumber)
End Sub
```

**លទ្ធផល :**



| DriveName | Total Size    | Free Space    | Used         | SerialNumber |
|-----------|---------------|---------------|--------------|--------------|
| C:\       | 20974428160Kb | 15404576768Kb | 5569851392Kb | 281673944    |
|           |               |               |              |              |
|           |               |               |              |              |
|           |               |               |              |              |

### I.13. GetFile

ប្រើសំរាប់បោះ File Name ទៅអោយអថេរមួយ ជាប្រភេទ File Object ។  
Syntax:

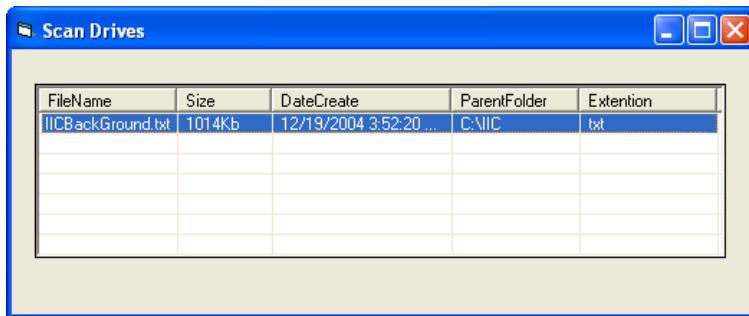
```
Dim F As File  
Set F = FSys.GetFile (PathSpec As String)
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit  
Dim FSys As New FileSystemObject  
.....  
Private Sub Form_Load()  
    Dim f As File  
    If FSys.FileExists("C:\BBU\BBUBackGround.txt") = True Then  
        Set f = FSys.GetFile("C:\BBU\BBUBackGround.txt")  
        Call AddColumnHeader(Me.ListView1, "FileName", "Size", "DateCreate",  
            "ParentFolder", "Extention")  
        Call AddDataToLvw(Me.ListView1, f.Name, f.Size & "Kb", f.DateCreated,  
            f.ParentFolder, FSys.GetExtensionName(f.Path))  
    Else  
        MsgBox "File Not Found "  
    End If  
End Sub  
.....
```

**លទ្ធផល :**



## I.14. GetFolder

ប្រើសំរាប់ពោះ Folder Name ទៅអោយអថេរមួយជាប្រភេទ Folder Object ។  
Syntax:

```
Dim Fd As Folder  
Set Fd = FSys.GetFolder (PathSpec As String)
```

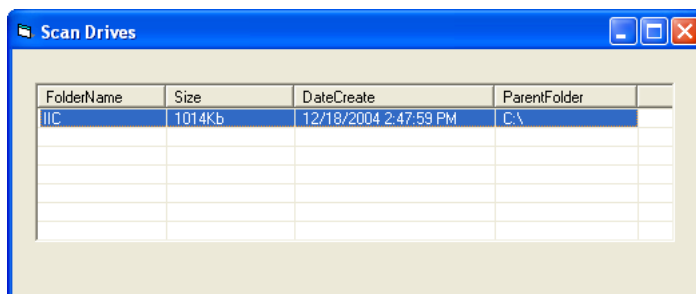
**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit  
Dim FSys As New FileSystemObject
```

```
Private Sub Form_Load()  
    Dim Fd As Folder  
    If FSys.FolderExists("C:\BBU\") = True Then  
        Set Fd = FSys.GetFolder("C:\BBU\  
        Call AddColumnHeader(Me.ListView1, "FolderName", "Size",  
            "DateCreate", "ParentFolder")  
        Call AddDataToLvw(Me.ListView1, Fd.Name, Fd.Size & "Kb",  
            Fd.DateCreated, Fd.ParentFolder)  
    Else  
        MsgBox "File Not Found "  
    End If  
End Sub
```

**លទ្ធផល :**



| FolderName | Size   | DateCreate            | ParentFolder |
|------------|--------|-----------------------|--------------|
| IIC        | 1014Kb | 12/18/2004 2:47:59 PM | C:\          |
|            |        |                       |              |
|            |        |                       |              |
|            |        |                       |              |
|            |        |                       |              |

## I.15. MoveFile

ប្រើសំរាប់ Cut File ពីទីតាំងមួយទៅដាក់ទីតាំងមួយទៀត។

Syntax:

```
FSys.MoveFile (Source As String , Destination As String)
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit
Dim FSys As New FileSystemObject
```

```
.....
Private Sub Form_Load()
    FSys.MoveFile "C:\BBU\Test1.txt", "D:\"
End Sub
.....
```

## I.16. MoveFolder

ប្រើសំរាប់ Cut Folder ពីទីតាំងមួយទៅដាក់ទីតាំងមួយទៀត។

Syntax:

```
FSys.MoveFolder (Source As String , Destination As String)
```

## II. របៀបប្រើ TextStream Object's Methods & Properties

TextStream Object គឺជាប្រភេទ Object មួយដែលអនុញ្ញាតអោយលោកអ្នក កាន់តែងាយស្រួលក្នុងការអានទិន្នន័យចេញពី Text File ឬ បញ្ចូលទិន្នន័យទៅអោយ Text File ។

មុននឹងលោកអ្នកអាចប្រើវាបានលោកអ្នកត្រូវប្រកាសអថេរមួយជាប្រភេទ TextStream Object ជាមុនសិន។

### II.1. Read

ប្រើសំរាប់អានទិន្នន័យចេញពី TextStream File ទៅតាមចំនួនតួអក្សរដែល លោកអ្នកបោះអោយ Argument ។

Syntax:

```
TextStreamObj.Read (Characters As Long) As String
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit
Dim FSys As New FileSystemObject
Dim InStream As TextStream
.....
Private Sub Form_Load( )
    Set InStream = FSys.OpenTextFile("
        C:\BBU\BBUBackGround.txt")
    MsgBox InStream.Read(5)
End Sub
.....
```

## II.2. ReadAll

ប្រើសំរាប់អានទិន្នន័យចេញពី TextStream File ទាំងអស់។

Syntax:

```
TextStreamObj.ReadAll
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit
Dim FSys As New FileSystemObject
Dim InStream As TextStream
.....
Private Sub Form_Load( )
    Set InStream = FSys.OpenTextFile("C:\BBU\BBUBackGround
        .txt", ForReading)
    MsgBox InStream.ReadAll
End Sub
.....
```

### II.3. ReadLine

ប្រើសំរាប់អានទិន្នន័យចេញពី TextStream File ម្តងមួយបន្ទាត់។  
Syntax:

```
TextStreamObj.ReadLine
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit
Dim FSys As New FileSystemObject
Dim InStream As TextStream
.....
Private Sub Form_Load( )
    Set InStream = FSys.OpenTextFile("C:\BBU\BBUBackGround
        .txt", ForReading)
    MsgBox InStream.ReadLine
End Sub
.....
```

### II.4. AtEndOfStream Property

ប្រើសំរាប់ត្រួតពិនិត្យមើលថាតើ File Pointer បានចង្អុលដល់ទីបញ្ចប់នៃ File ហើយឬនៅ។  
ប្រសិនបើ File Pointer បានចង្អុលដល់ ទីបញ្ចប់នៃ File ហើយនោះ Property និងទទួលបានតំលៃ True  
ផ្ទុយទៅវិញ Property នេះនឹងទទួលបានតំលៃ False ។

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit
Dim FSys As New FileSystemObject
Dim InStream As TextStream
.....
Private Sub Form_Load( )
    Set InStream = FSys.OpenTextFile("C:\BBU\BBUBackGround.txt",
        ForReading)
    Do While InStream.AtEndOfLine = False
```

```
MsgBox InStream.ReadLine
Loop
End Sub
```

---

## II.5. Skip

ប្រើសំរាប់ចំលងចំនួនតួអក្សរទៅតាម Characters Argument ដោយគិតចាប់ពី ទីតាំងរបស់ File

Pointer ។

Syntax:

```
TextStreamObj.Skip (Characters As Long)
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit
Dim FSys As New FileSystemObject
```

```
Dim InStream As TextStream
```

---

```
Private Sub Form_Load()
    Set InStream = FSys.OpenTextFile("C:\BBU\BBUBackGround
        .txt", ForReading)
    Do While InStream.AtEndOfLine = False
        InStream.Skip (5)
        MsgBox InStream.ReadLine
    Loop
End Sub
```

---

## II.6. SkipLine

ប្រើសំរាប់ចំលងចំនួនតួអក្សរម្តងមួយបន្ទាត់ដោយគិតចាប់ពីទីតាំងរបស់ File Pointer ។

Syntax:

```
TextStreamObj.SkipLine
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit
Dim FSys As New FileSystemObject
Dim InStream As TextStream
```

```
.....
Private Sub Form_Load()
    Set InStream = FSys.OpenTextFile("C:\BBU\BBUBackGround
        .txt", ForReading)
    InStream.SkipLine
    Do While InStream.AtEndOfLine = False
        MsgBox InStream.ReadLine
    Loop
End Sub
.....
```

## II.7. Write

ប្រើសំរាប់បញ្ចូលទិន្នន័យអោយ Text File ហើយទិន្នន័យទាំងនោះ គឺវាទៅតែក្នុងមួយបន្ទាត់។

Syntax:

```
TextStreamObj.Write (Text As String)
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit
Dim FSys As New FileSystemObject
Dim InStream As TextStream
```

```
.....
Private Sub Form_Load()
    Set InStream = FSys.OpenTextFile("C:\BBU\test1.txt", ForAppending, True)
    InStream.Write "Welcome to BBU"
    InStream.Write "Thank for buy my IT Magazine"
End Sub
.....
```

## II.8. WriteLine

ប្រើសំរាប់បញ្ចូលទិន្នន័យអោយ Text File ។ ហើយបន្ទាប់ពី Statement នេះត្រូវបាន Process ម្តងនោះ File Pointer ត្រូវរុញទៅកាន់បន្ទាត់ថ្មី។



Syntax:

```
TextStreamObj.WriteLine (Text As String)
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit  
Dim FSys As New FileSystemObject  
Dim InStream As TextStream
```

```
.....  
Private Sub Form_Load()  
    Set InStream = FSys.OpenTextFile("C:\BBU\test1.txt", ForAppending, True)  
    InStream.WriteLine "Welcome to BBU"  
    InStream.WriteLine "Thank for buy my IT Magazine"  
End Sub  
.....
```

## II.9. WriteBlankLines

ប្រើសំរាប់បញ្ចូលតែបន្ទាត់ទទេហើយគ្មានទិន្នន័យទៅអោយ TextStream File ។

Syntax:

```
TextStreamObj.WriteBlankLine (Lines As Long)
```

**ឧទាហរណ៍ :**

☞ Source Code:

```
Option Explicit  
Dim FSys As New FileSystemObject  
Dim InStream As TextStream
```

```
.....  
Private Sub Form_Load()  
    Set InStream = FSys.OpenTextFile("C:\BBU\test1.txt", ForAppending, True)  
    InStream.WriteLine "Welcome to BBU"  
    InStream.WriteBlankLines (2)  
    InStream.WriteLine "Thank for buy my IT Magazine"  
End Sub  
.....
```

## II.10. លំហាត់អនុវត្តន៍លើ TextStreamFile

សូមលោកអ្នកយកលំហាត់នៅក្នុងមេរៀន File មកធ្វើឡើងវិញដោយប្តូរមកប្រើ ជាទំរង់ TextStreamFile Object ( សូមត្រលប់ទៅមើលមេរៀនទី៧វិញ ) ។

## III. របៀបធ្វើការជាមួយ Files និង Folders Collection

### III.1. របៀបធ្វើការជាមួយ Files Collection

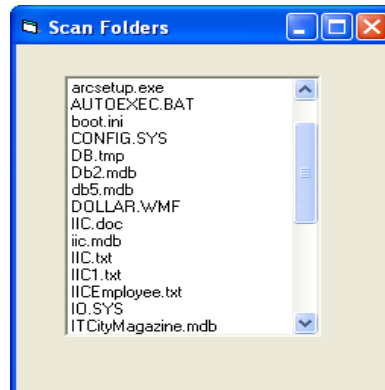
ឧបមាថាលោកអ្នកធ្វើការ Scan រាល់ File ដែលមាននៅក្នុង Drive C:\ អោយវាបង្ហាញឈ្មោះនៅក្នុង ListBox ។

☞ Source Code:

```
Option Explicit  
Dim Fd As Folder  
Dim F As File  
Dim AllFiles As Files  
Dim FSys As New FileSystemObject
```

```
Private Sub Form_Load()  
    Set Fd = FSys.GetFolder("C:\")  
    Set AllFiles = Fd.Files  
    For Each F In AllFiles  
        Me.List1.AddItem F.Name  
    Next  
End Sub
```

លទ្ធផល :



### III.2. ប្រើប្រាស់ការរៀបចំ Folders Collection

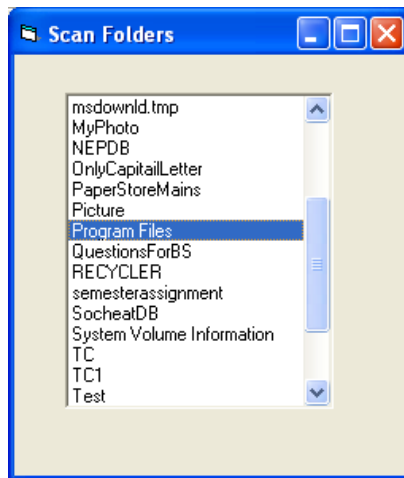
ឧបមាថាលោកអ្នកធ្វើការ Scan រាល់ ឈ្មោះ Folder ដែលមាននៅក្នុង Drive C:\ អោយវាបង្ហាញឈ្មោះនៅក្នុង ListBox ។

☞ Source Code:


```
Option Explicit  
Dim FSys As New FileSystemObject  
Dim Fd As Folder  
Dim F As File  
Dim AllFolders As Folders
```

```
.....  
Private Sub Form_Load ()  
    Set Fd = FSys.GetFolder("f:\")  
    Set AllFolders = Fd.SubFolders  
    For Each Fd In AllFolders  
        Me.List1.AddItem Fd.Name  
    Next  
End Sub  
.....
```

**លទ្ធផល :**



### III.3. Scan Folders Recursive

 Source Code:

```
Option Explicit
Dim FSys As New FileSystemObject
Dim Fd As Folder
Dim F As File
Dim AllFolders As Folders
```

---

```
Private Sub Form_Load()
    ScanFolder ("C:\Program Files\")
End Sub
```

---

```
Sub ScanFolder(FolderSpec As String)
    Dim thisFolder As Folder
    Dim sFolder As Folders
    Dim fileItem As File, folderItem As Folder
    Dim AllFiles As Files

    Set thisFolder = FSys.GetFolder(FolderSpec)
    For Each folderItem In thisFolder.SubFolders
        Me.List1.AddItem folderItem.Name
        ScanFolder (folderItem.Path)
    Next
End Sub
```

---

===== \* =====