

نکات ریز و درشت در مورد استفاده از فایل ها

- ۱- بارگذاری محتویات یک فایل به درون Combo Box
- ۲- بارگذاری محتویات یک فایل به درون Text Box
- ۳- بررسی وجود یک فایل :
- ۴- بررسی وجود یک دایرکتوری
- ۵- حذف فایل
- ۶- جستجوی سریع
- ۷- نحوه مواجهه با نامهای طولانی
- ۸- ایجاد دایرکتوری و شاخه در زمان اجرا
- ۹- پر کردن list box با فایل و دایرکتوری و غیره
- ۱۰- بدست آوردن اندازه یک فایل
- ۱۱- بدست آوردن مسیر فایل از رشته ای که حاوی نام فایل نیز می باشد
- ۱۲- بدست آوردن پسوند فایل از رشته ای که حاوی نام و مسیر فایل می باشد
- ۱۳- کوتاه سازی مسیر طولانی
- ۱۴- جستجو بر روی دیسک سخت
- ۱۵- چگونه file list box را وادار کنیم که محتویات دیسک را دوباره بخواند
- ۱۶- File Error

۱- بارگذاری محتویات یک فایل به درون Combo Box :
در کد ذیل هر خط فایل یک آیتم ورودی می باشد .

```
Open "compnames.txt" For Input As #1
Do While Not EOF(1)
    Line Input #1, lne$
    combol.AddItem lne$
Loop
Close #1
```

۲- بارگذاری محتویات یک فایل به درون Text Box :

```
Dim FileLength
Open "yourfile.txt" For Input As #1
FileLength = LOF(1)
var1 = Input(FileLength, #1)
Text1.Text = var1
Close #1
```

۳- بررسی وجود یک فایل :

```
Public Function FileExists(ByVal sFileName As String) As Boolean
    Dim sFile As String
    On Error Resume Next
    FileExists = False
    sFile = Dir$(sFileName)
    If (Len(sFile) > 0) And (Err = 0) Then
        FileExists = True
    End If
End Function
```

۴- بررسی وجود یک دایرکتوری :

```
public Function DirExists(ByVal sDirName As String) As Boolean
    Dim sDir As String
    On Error Resume Next
    DirExists = False
    sDir = Dir$(sDirName, vbDirectory)
    If (Len(sDir) > 0) And (Err = 0) Then
        DirExists = True
    End If
End Function
```

۵- حذف فایل :

```
Public Sub FileKill(ByVal sFileName As String)
    On Error Resume Next

    If FileExists(sFileName) Then
        Kill sFileName
    End If
End Function
```

۶- جستجوی سریع :

```
Sub DirWalk(ByVal sPattern As String, ByVal CurrDir As String, _
    sFound() As String)
Dim i As Integer
Dim sCurrPath As String
Dim sFile As String
Dim ii As Integer
Dim iFiles As Integer
Dim iLen As Integer
If Right$(CurrDir, 1) <> "\" Then
    Dir1.Path = CurrDir & "\"
Else
    Dir1.Path = CurrDir
End If
For i = 0 To Dir1.ListCount
    If Dir1.List(i) <> "" Then
        DoEvents
        Call DirWalk(sPattern, Dir1.List(i), sFound)
    Else
        If Right$(Dir1.Path, 1) = "\" Then
            sCurrPath = Left(Dir1.Path, Len(Dir1.Path) - 1)
        Else
            sCurrPath = Dir1.Path
        End If
        File1.Path = sCurrPath
        File1.Pattern = sPattern
        If File1.ListCount > 0 Then 'matching files found in the
            'directory
            For ii = 0 To File1.ListCount - 1
                ReDim Preserve sFound(UBound(sFound) + 1)
                sFound(UBound(sFound) - 1) = sCurrPath & "\" & _
                    File1.List(ii)
            Next ii
        End If
        iLen = Len(Dir1.Path)
        Do While Mid(Dir1.Path, iLen, 1) <> "\"
            iLen = iLen - 1
        Loop
        Dir1.Path = Mid(Dir1.Path, 1, iLen)
    End If
Next i
End Sub
```

۷- نحوه مواجهه با نامهای طولانی :

مثال :

```
C:\MyLongestPath\MyLongerPath\MyFilename.txt
C:\Mylong~1\MyLong~2\Myfile~1.txt
```

```
'Put the declaration in a .bas module
Declare Function GetShortPathName Lib "kernel32" Alias _
    "GetShortPathNameA" (ByVal lpszLongPath As String, _
    ByVal lpszShortPath As String, ByVal cchBuffer As Long) As Long
```

'Next comes the function (Place in a module):

```

Public Function GetDosPath(LongPath As String) As String
Dim s As String
Dim i As Long
Dim PathLength As Long
    i = Len(LongPath) + 1
    s = String(i, 0)
    PathLength = GetShortPathName(LongPath, s, i)
    GetDosPath = Left$(s, PathLength)
End Function

'Lastly call it like this:
    DosPathName = GetDosPath(Long Path Goes in here)

```

۸- ایجاد دایرکتوری و شاخه در زمان اجرا :

```

' Here's how to create a file directory and subdirectories

' Add these 3 functions and the MakeDir subroutine:
' This function is used by MakeDir to validate if a
' directory already exists.
Function bValDir (ByVal sDirIn As String) As Integer
Dim iCheck As String, iErrResult As Integer
On Local Error GoTo ValDirError
    sDirIn = sParsePath(sDirIn)
    sDirIn = sFixDirString(sDirIn)
    iCheck = Dir$(sDirIn)
    If iErrResult <> 0 Then
        bValDir = False
    Else
        bValDir = True
    End If
Exit Function
ValDirError:
    iErrResult = Err
Resume Next

End Function
' This procedure will add a \ to the end of the directory
' name if needed.
Function sFixDirString (sInComming As String) As String
Dim sTemp As String
    sTemp = sInComming
    If Right$(sTemp, 1) <> "\" Then
        sFixDirString = sTemp & "\"
    Else
        sFixDirString = sTemp
    End If
End Function
' This procedure will return just the path name from the
' string containing the path.
Function sParsePath (sPathIn As String) As String
Dim I As Integer
    For I = Len(sPathIn) To 1 Step -1
        If InStr(":\", Mid$(sPathIn, I, 1)) Then Exit For
    Next
    sParsePath = Left$(sPathIn, I)
End Function
' The MakeDir routine will create a directory even if the
' underlying directories do not exist.

```

```

Sub MakeDir (sDirName As String)
Dim iMouseState As Integer
Dim iNewLen As Integer
Dim iDirLen As Integer
    'Get Mouse State
    iMouseState = Screen.MousePointer
    'Change Mouse To Hour Glass
    Screen.MousePointer = 11
    'Set Start Length To Search For [\]
    iNewLen = 4
    'Add [\] To Directory Name If Not There
    sDirName = sFixDirString(sDirName)
    'Create Nested Directory
    While Not bValDir(sDirName)
        iDirLen = InStr(iNewLen, sDirName, "\")

        If Not bValDir(Left$(sDirName, iDirLen)) Then
            MkDir Left$(sDirName, iDirLen - 1)
        End If

        iNewLen = iDirLen + 1
    Wend
    'Leave The Mouse The Way You Found It
    Screen.MousePointer = iMouseState
End Sub
'Example:
' For instance, typing "C:\aaa\biggins" in Text1 will create
' the directory named "C:\aaa" and also create a subdirectory
' under "C:\aaa" called "biggins" (C:\aaa\biggins)
' Typing "\aaa" will create the directory on the current drive
Sub Command1_Click ()
Dim sDirString As String
    'Use the string in Text1
    sDirString = Text1.Text
    'Trap for errors
    On Error GoTo ErrHandle
    'Call the MakeDir routine
    MakeDir sDirString
ErrHandle:
    MsgBox Error$
Exit Sub
End Sub

```

۹- پر کردن list box با فایل و دایرکتوری و غیره :

```

' The SendMessage API provides a simple way to use a
' standard List Box to list files. By sending the LB_DIR
' message to the List Box, the list box automatically
' fills itself with a list of files, directories and drives.
' Create a new project. Add a List box (List1) and a
' Command button (Command1) to the form.
' In a .BAS module add the following declarations:
' Declare API
Private Declare Function SendMessage Lib "user32" Alias _
    "SendMessageA" (ByVal hwnd As Long, ByVal wParam As Long, _
    ByVal lParam As Long, lParam As Any) As Long
' Note**, The author defined these constants, so you
' won't find them in API documentation, although they

```

```

' are based on values used with LB_DIR.
Global Const WM_USER = &H400
Global Const LB_DIR = (WM_USER + 14)
Global Const DIR_NORMALFILES = &H0
Global Const DIR_READONLY = &H8001
Global Const DIR_HIDDEN = &H8002
Global Const DIR_SYSTEM = &H8004
Global Const DIR_DIRECTORIES = &H8010
Global Const DIR_ARCHIVED = &H8020
Global Const DIR_DRIVES = &HC000
' Add this routine to your Form's declaration section:
Sub ListFiles (sFileSpec As String)
Dim i As Long
' Clear existing data
List1.Clear
' Add files / directories of specified types
i = SendMessage(List1.hWnd, LB_DIR, DIR_DRIVES, ByVal sFileSpec)
i = SendMessage(List1.hWnd, LB_DIR, DIR_DIRECTORIES, ByVal sFileSpec)
i = SendMessage(List1.hWnd, LB_DIR, DIR_NORMALFILES, ByVal sFileSpec)
End Sub
' In Command1_Click call the ListFiles routine:
Sub Command1_Click ()
' List all files from C:\Windows directory
Call ListFiles("C:\WINDOWS\*.*.*)
' Or to list the whole tree for the current drive:
' Call ListFiles("*.*.*)
End Sub

```

۱۰- بدست آوردن اندازه یک فایل :

```

' You can get the size of a file two ways.
' If you have the file open you can use the LOF function.
Dim nFileNum As Integer
Dim lFileSize As Long

'Get a new file number
nFileNum = FreeFile

'Open the file
Open "C:\SOMEFILE.TXT" For Input As nFileNum

'Get the Length
lFileSize = LOF(nFileNum)

'Close the file
Close nFileNum

' If you don't have the file open you can use the FileLen function.
Dim lFileSize As Long

lFileSize = FileLen("C:\SOMEFILE.TXT")

```

۱۱- بدست آوردن مسیر فایل از رشته ای که حاوی نام فایل نیز می باشد :

```

Function ParsePath (sPathIn As String) As String
Dim I As Integer
For I = Len(sPathIn) To 1 Step -1

```

```

        If InStr(":\", Mid$(sPathIn, I, 1)) Then Exit For
    Next
    ParsePath = Left$(sPathIn, I)
End Function

```

۱۲- بدست آوردن پسوند فایل از رشته ای که حاوی نام و مسیر فایل می باشد :

```

Function GetFileExt (sFileName As String) As String
Dim P As Integer
    For P = Len(sFileName) To 1 Step -1
        'Find the last occurrence of "." in the string
        If InStr(".", Mid$(sFileName, P, 1)) Then Exit For
    Next

    GetFileExt = Right$(sFileName, Len(sFileName) - P)
End Function

' Example:
' Add a Textbox (Text1), add command button (Command1) and
' add 3 labels (Label1, 2, 3)
' Add this code to the Command1 click event:
Sub Command1_Click ()
Dim sTargetString As String
    ' Store the string
    sTargetString = Text1.Text

    'Display the path
    Label1 = ParsePath(sTargetString)
    'Display the filename
    Label2 = ParseFileName(sTargetString)
    'Display the file's extension
    Label3 = GetFileExt(sTargetString)
End Sub

' Run the program and type a valid path and file name in the Text1
' textbox and click the Command1 button.

' Returns the file portion of a file + pathname
,
Public Function GetFile(s As String) As String
    Dim i As Integer
    Dim j As Integer

    i = 0
    j = 0

    i = InStr(s, "\")
    Do While i <> 0
        j = i
        i = InStr(j + 1, s, "\")
    Loop

    If j = 0 Then
        GetFile = ""
    Else
        GetFile = Right$(s, Len(s) - j)
    End If
End Function

```

```

'
' Returns the path portion of a file + pathname
'
Public Function GetPath(s As String) As String
    Dim i As Integer
    Dim j As Integer

    i = 0
    j = 0

    i = InStr(s, "\")
    Do While i <> 0
        j = i
        i = InStr(j + 1, s, "\")
    Loop

    If j = 0 Then
        GetPath = ""
    Else
        GetPath = Left$(s, j)
    End If
End Function

```

۱۳- کوتاه سازی مسیر طولانی :

```

Function LongDirFix (TargetString As String, Max As Integer) As
String
Dim i, LblLen, StringLen As Integer
Dim TempString As String
    TempString = TargetString
    LblLen = Max
    If Len(TempString) <= LblLen Then
        LongDirFix = TempString
        Exit Function
    End If

    LblLen = LblLen - 6
    For i = Len(TempString) - LblLen To Len(TempString)
        If Mid$(TempString, i, 1) = "\" Then Exit For
    Next

    'On one line:
    LongDirFix = Left$(TempString, 3) & "..." &
Right$(TempString,
        Len(TempString) - (i - 1))
End Function
' Example:
Dim DirName As String
    DirName = "C:\DICKENS\DIVISION\RESTRNT\PAYROLL\JONES_D"
    Label1.Caption = LongDirFix(DirName, 32)

' The second paramater is the max length of the returned string.

```

۱۴- جستجو بر روی دیسک سخت :

```

Global searchme$
'The following needs to be on one line.

```

```

Declare Function SendMessageBystring& Lib "User32" ALIAS _
"SendMessage" (ByVal hWnd As Long, ByVal wParam As Long, _
ByVal lParam As Any)
Global Const WM_USER = &H400
Global Const LB_SELECTSTRING = (WM_USER + 13)
Global Const LB_FINDSTRING = (WM_USER + 16)
'In File1 under keydown, add the following:
'This checks if the user has pressed the up or down arrow.
'If they have, reset searchme$ to "".
If KeyCode = 40 Or KeyCode = 38 Then
    searchme$ = ""
End If
'In File1 under lostfocus, pathchange, patternchange, and click add:
'If the user has done any of the above, reset the searchme$
'string.
searchme$ = ""
'In File1 under keypress add:
Dim result&
Select Case KeyAscii
    Case 8 'Backspace
        If searchme$ <> "" Then
            searchme$ = Left$(searchme$, Len(searchme$) - 1)
        Else
            File1.ListIndex = 0
        End If
        KeyAscii = 0
        Exit Sub
    Case 27 'Escape
        searchme$ = ""
        KeyAscii = 0
        Exit Sub
    Case 13 'Enter
        searchme$ = ""
        KeyAscii = 0
        Exit Sub
    Case Asc("a") To Asc("z"), Asc("A") To Asc("Z"), Asc("'"), _
        Asc("."), Asc(" "), Asc("0") To Asc("9")
        searchme$ = searchme$ & Chr$(KeyAscii)
        KeyAscii = 0
End Select
result& = SendMessageBystring(FILE1.hWnd, LB_FINDSTRING, 0, _
searchme$)
If result& = -1 Then
    searchme$ = Left$(searchme$, Len(searchme$) - 1)
Else
    result& = SendMessageBystring(FILE1.hWnd, LB_SELECTSTRING, -1, _
searchme$)
End If

```

۱۵- چگونه file list box را وادار کنیم که محتویات دیسک را دوباره بخواند :

```

drivel.refresh
dir1.refresh
file1.refresh

```

: File Error -۱۶

```

Function FileErrors(errVal As Integer) As Integer
Dim MsgType As Integer
Dim Response As Integer
Dim Action As Integer
Dim Msg As String
MsgType = vbExclamation
Select Case errVal
    Case errDeviceUnavailable ' Error #68
        Msg = "That device appears to be unavailable."
        MsgType = vbExclamation + 5
    Case errDiskNotReady ' Error #71
        Msg = "The disk is not ready."
    Case errDeviceIO
        Msg = "The disk is full."
    Case errBadFileName, errBadFileNameOrNumber ' Errors #64 & 52
        Msg = "That filename is illegal."
    Case errPathDoesNotExist ' Error #76
        Msg = "That path doesn't exist."
    Case errBadFileMode ' Error #54
        Msg = "Can't open your file for that type of access."
    Case errFileAlreadyOpen ' Error #55
        Msg = "That file is already open."
    Case errInputPastEndOfFile ' Error #62
        Msg = "This file has a nonstandard end-of-file marker,"
        Msg = Msg + "or an attempt was made to read beyond "
        Msg = Msg + "the end-of-file marker."
    Case Else
        FileErrors = 3
        Exit Function
End Select
Response = MsgBox(Msg, MsgType, "File Error")
Select Case Response
    Case 4 ' Retry button.
        FileErrors = 0
    Case 5 ' Ignore button.
        FileErrors = 1
    Case 1, 2, 3 ' OK and Cancel buttons.
        FileErrors = 2
    Case Else
        FileErrors = 3
End Select
End Function

```