

HOW TO: Recursively Search Directories by Using FileSystemObject

This step-by-step article demonstrates how to use the **FileSystemObject** class to recursively search directories and to find specific files.

Information About the FileSystemObject Class

The **FileSystemObject** class is located in the Microsoft Scripting Runtime (Srrun.dll). To obtain the Srrun.dll file, install any of the following packages:

- Microsoft Windows Script Host
- Microsoft Windows NT Option Pack
- Microsoft Internet Information Server 3.0
- Scripting 3.1 upgrade
- Microsoft Visual Studio 98
- Microsoft Visual Basic 6.0

The **FileSystemObject** class provides better performance than if you use Visual Basic intrinsic functions such as **Dir** and **GetAttr**. Additionally, **FileSystemObject** is much simpler to implement than the Visual Basic intrinsic functions.

Build the Sample

1. Create a new Standard EXE project in Visual Basic. By default, Form1 is created.
2. On the **Project** menu, click **References**, and then add a reference to the Microsoft Scripting Runtime. If this option is not listed, locate the Srrun.dll file on your system. If you have to, install one of the tools that is listed in the [Information About the FileSystemObject Class](#) section.
3. Add a **CommandButton**, a **Label**, and a **ListBox** control to Form1. Adjust the width of the **Label** control so that it is the same as the width of the form.
4. Add the following code in the General Declarations section of Form1:

```

Option Explicit

Dim fso As New FileSystemObject
Dim fld As Folder

Private Sub Command1_Click()
    Dim nDirs As Long, nFiles As Long, lSize As Currency
    Dim sDir As String, sSrchString As String
    sDir = InputBox("Type the directory that you want to search for", _
        "FileSystemObjects example", "C:\")
    sSrchString = InputBox("Type the file name that you want to search for", _
        "FileSystemObjects example", "vb.ini")
    MousePointer = vbHourglass
    Label1.Caption = "Searching " & vbCrLf & UCase(sDir) & "... "
    lSize = FindFile(sDir, sSrchString, nDirs, nFiles)
    MousePointer = vbDefault
    MsgBox Str(nFiles) & " files found in" & Str(nDirs) & _
        " directories", vbInformation
    MsgBox "Total Size = " & lSize & " bytes"
End Sub

Private Function FindFile(ByVal sFol As String, sFile As String, _
    nDirs As Long, nFiles As Long) As Currency
    Dim tFld As Folder, tFil As File, FileName As String

    On Error GoTo Catch
    Set fld = fso.GetFolder(sFol)
    FileName = Dir(fso.BuildPath(fld.Path, sFile), vbNormal Or _
        vbHidden Or vbSystem Or vbReadOnly)
    While Len(FileName) <> 0
        FindFile = FindFile + FileLen(fso.BuildPath(fld.Path, _
            FileName))
        nFiles = nFiles + 1
        List1.AddItem fso.BuildPath(fld.Path, FileName) ' Load ListBox
        FileName = Dir() ' Get next file
        DoEvents
    Wend
    Label1 = "Searching " & vbCrLf & fld.Path & "... "
    nDirs = nDirs + 1
    If fld.SubFolders.Count > 0 Then
        For Each tFld In fld.SubFolders
            DoEvents
            FindFile = FindFile + FindFile(tFld.Path, sFile, nDirs, nFiles)
        Next
    End If
    Exit Function
Catch: FileName = ""
    Resume Next
End Function

```

5. Run the project, and then click **Command1**.
6. Type the directory and the file name to search for. Notice that when each file is found, the file name is added to the list box. When the process is complete, the number of files that are found is displayed in a message box. The total size of the files is also displayed.

Troubleshooting

- Some files and directories, such as System Volume Information on Microsoft Windows XP, cause an access violation if an application tries to access them. The error handling code stops looking in a directory when a problem occurs. You must use a different approach if you must have a more robust workaround.

- If you use a file filter (such as *.*), a large number of files may be returned. The **ListBox** control can only contain a limited number of characters. When that limit is reached, no more entries are added to the list box.
- This code sample was tested by using version 5.6.0.6626 of the Microsoft Scripting Runtime (Scriun.dll). If problems occur, you may have to download a newer version of this file.