

Re: Fast Size Folder

Source:

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2007-09/msg00181.html>

- *From:* Lorin <Lorin@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Mon, 3 Sep 2007 18:24:01 -0700
-

Something akin to what you can call in FileSystemObject only in an API so I do not have to include the Scripting Lib which is verboten some place.

Dim fFldr As Folder

```
Set fFldr = fso.GetFolder(sfolder)
FolderSizeFSO = CDec(fFldr.Size)
Set fFldr = Nothing
```

"Mike Williams" wrote:

"Lorin" <Lorin@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message news:15CD0F37-8291-4CF2-8F98-6A0BBEB4A485@xxxxxxxxxxxxxxxxxxxxx

Now, how about getting size information about a Folder quickly. Currently I am recursively looping on files sizes and adding... too long a wait. Is there an API to get Folder (with subfolders) size?

No. The system doesn't store the size of folders in quite the same way that it stores the size and space used for a Drive, which is immediately accessible. Recursive looping is about the only way to do it. In fact, as far as I know that is exactly what Windows itself does when you right click a folder and choose Properties. You can see this if you select a really large folder containing thousands of sub folders and tens of thousands of files and right click it and select Properties. It takes a long time for Windows to finish calculating the size of the folder when you first perform such an act. You can see the result gradually building up as Windows displays a running count of the total.

Thereafter the result is shown very quickly, either because there is a lot of relevant data already in the disk buffers by that time or perhaps because the file info data is buffered elsewhere, but the "first time" calculation when you choose Properties for any folder with tens of thousands of files

Re: Fast Size Folder

always takes Windows a considerable amount of time, no matter how long the system has been running. This is exactly the same behaviour that you get with VB code which performs the same task, and the timing in both cases (first check and subsequent checks) for your VB code is more or less exactly the same as the timing you get when you manually right click a folder and select Properties. VB takes about the same amount of time to do the job as does Windows itself.

By the way, when you do the job in code you need to make sure that you do it in such a way as to properly account for individual files and totals exceeding both the 2GB and the 4GB barrier. And of course you will probably want to write your code in such a way that it also properly calculates the "space used on disk" as well as the total file data size. Here's an example for you. paste it into a VB Form containing a Command Button.

Mike

Option Explicit

```
Private Declare Function GetDiskFreeSpace _
Lib "kernel32" Alias "GetDiskFreeSpaceA" _
(ByVal lpRootPathName As String, _
lpSectorsPerCluster As Long, _
lpBytesPerSector As Long, _
lpNumberOfFreeClusters As Long, _
lpTotalNumberOfClusters As Long) As Long
Private Declare Sub CopyMemory Lib "kernel32" _
Alias "RtlMoveMemory" (ByRef Destination As Any, _
Source As Any, ByVal Length As Long)
Private Declare Function SHBrowseForFolder _
Lib "shell32.dll" (bBrowse As BrowseInfo) As Long
Private Declare Function SHGetPathFromIDList _
Lib "shell32.dll" (ByVal lItem As Long, _
ByVal sDir As String) As Long
Private Declare Function FindFirstFile Lib "kernel32" _
Alias "FindFirstFileA" (ByVal lpFileName As String, _
lpFindFileData As WIN32_FIND_DATA) As Long
Private Declare Function FindNextFile Lib "kernel32" _
Alias "FindNextFileA" (ByVal hFindFile As Long, _
lpFindFileData As WIN32_FIND_DATA) As Long
Private Declare Function FindClose Lib "kernel32" _
(ByVal hFindFile As Long) As Long
Private Type FILETIME
dwLowDateTime As Long
dwHighDateTime As Long
End Type
Private Const MAX_PATH = 260
Private Type WIN32_FIND_DATA
lngFileAttributes As Long
ftCreationTime As FILETIME
ftLastAccessTime As FILETIME
ftLastWriteTime As FILETIME
```

Re: Fast Size Folder

```
lngFileSizeHigh As Long
lngFileSizeLow As Long
lngReserved0 As Long
lngReserved1 As Long
fileName As String * MAX_PATH
strAlternate As String * 14
End Type
Private Type BrowseInfo
hWndOwner As Long
pidlRoot As Long
sDisplayName As String
sTitle As String
ulFlags As Long
lpfn As Long
lParam As Long
iImage As Long
End Type
Private Const FILE_ATTRIBUTE_ARCHIVE = &H20
Private Const FILE_ATTRIBUTE_COMPRESSED = &H800
Private Const FILE_ATTRIBUTE_DIRECTORY = &H10
Private Const FILE_ATTRIBUTE_HIDDEN = &H2
Private Const FILE_ATTRIBUTE_NORMAL = &H80
Private Const FILE_ATTRIBUTE_READONLY = &H1
Private Const FILE_ATTRIBUTE_SYSTEM = &H4
Private Const FILE_ATTRIBUTE_TEMPORARY = &H100
Private Const BIF_RETURNONLYFSDIRS As Long = &H1
Private directories As Long
Private folders As Long
Private files As Long
Private totalSize As Currency
Private totalSizeOnDisk As Currency
Private loMask As Long
Private hiMask As Long
Private clusterSize As Currency

Public Sub GetClusterSize(disk As String)
Dim s1 As String, sectorsPerCluster As Long
Dim bytesPerSector As Long, free As Long
Dim total As Long, retVal As Long
Dim bytesperCluster As Long
retVal = GetDiskFreeSpace _
(disk, sectorsPerCluster, bytesPerSector, free, total)
bytesperCluster = sectorsPerCluster * bytesPerSector
loMask = bytesperCluster - 1
hiMask = &HFFFFFFF - loMask
clusterSize = CCur(bytesperCluster / 10000&)
End Sub

Private Function Browse_Folder() As String
Dim bInf As BrowseInfo
Dim lItem As Long
```

Re: Fast Size Folder

```
Dim sDirName As String
Dim hwnd As Long
bInf.hWndOwner = Me.hwnd
bInf.sDisplayName = Space$(MAX_PATH)
bInf.sTitle = "Select Folder"
bInf.ulFlags = BIF_RETURNONLYFSDIRS
IItem = SHBrowseForFolder(bInf)
If IItem Then
sDirName = Space$(MAX_PATH)
If SHGetPathFromIDList(IItem, sDirName) Then
Browse_Folder = Left(sDirName, InStr(sDirName, _
Chr$(0)) - 1)
Else
Browse_Folder = ""
End If
End If
End Function

Private Sub countFiles(folderPath As String, _
recurse As Boolean)
Dim fd As WIN32_FIND_DATA
Dim hFind As Long
Dim strFile As String
Dim strSearch As String
Dim fileSize As Currency
Dim SizeOnDisk As Currency
Dim roundUp As Boolean
If Right$(folderPath, 1) <> "\" Then
folderPath = folderPath & "\"
End If
strSearch = folderPath & "*"
hFind = FindFirstFile(strSearch, fd)
If hFind > 0 Then
Do
If (fd.lngFileAttributes And FILE_ATTRIBUTE_DIRECTORY) _
= FILE_ATTRIBUTE_DIRECTORY Then
If Left$(fd.fileName, 1) = "." Then
' ignore these
Else
' It is a directory
folders = folders + 1
If recurse Then
countFiles folderPath & Left$(fd.fileName, _
InStr(fd.fileName, Chr(0)) - 1), recurse
End If
End If
Else
files = files + 1
If (fd.lngFileSizeLow And loMask) <> 0 Then
roundUp = True
Else
```

Re: Fast Size Folder

```
roundUp = False
End If
CopyMemory fileSize, _
fd.ftLastWriteTime.dwHighDateTime, 8
CopyMemory fileSize, fd.lngFileSizeLow, 4
totalSize = totalSize + fileSize
If roundUp Then
fd.lngFileSizeLow = fd.lngFileSizeLow _
And hiMask
End If
CopyMemory SizeOnDisk, _
fd.ftLastWriteTime.dwHighDateTime, 8
CopyMemory SizeOnDisk, fd.lngFileSizeLow, 4
If roundUp Then '
totalSizeOnDisk = totalSizeOnDisk + _
SizeOnDisk + clusterSize
Else
totalSizeOnDisk = totalSizeOnDisk + _
SizeOnDisk
End If
End If
Loop While CBool(FindNextFile(hFind, fd))
Call FindClose(hFind)
End If
End Sub
```

```
Private Sub Command1_Click()
Dim sfolder As String, s1 As String
Dim t1 As Single, t2 As Single
Dim Sectors As Long, Bytes As Long
Dim FreeC As Long, TotalC As Long
Dim total As Long, Freeb As Long
Command1.Enabled = False
sfolder = Browse_Folder
If sfolder = "" Then
Caption = "No Folder Selected"
Else
GetClusterSize Left$(sfolder, InStr(sfolder, "\"))
directories = 0
folders = 0
files = 0
totalSize = 0
totalSizeOnDisk = 0
s1 = Caption: Caption = "Please wait . . ."
t1 = Timer
countFiles sfolder, True
t2 = Timer
Caption = s1
s1 = Format(files, "###,###,##0") & " files, "
s1 = s1 & Format(folders, "###,###,##0") & " folders."
s1 = s1 & vbCrLf & _
```

Re: Fast Size Folder

```
"Total file size = " & _  
Format((totalSize * 10000&), "###,###,###,##0")  
s1 = s1 & vbCrLf & _  
"Total size on disk = " & _  
Format((totalSizeOnDisk * 10000&), "###,###,###,##0")  
s1 = s1 & vbCrLf & _  
"Time taken = " & Format(t2 - t1, "###.00") & " seconds."  
MsgBox s1  
End If  
DoEvents  
Command1.Enabled = True  
End Sub
```