

# Returning selected items in an open folder

---

*Source:*

<http://www.tech-archive.net/Archive/VB/microsoft.public.vb.general.discussion/2007-08/msg01081.html>

---

- *From:* "mayayana" <[mayaXXyana1a@xxxxxxxxxxxxxxxxxxxx](mailto:mayaXXyana1a@xxxxxxxxxxxxxxxxxxxx)>
  - *Date:* Mon, 13 Aug 2007 23:39:08 -0400
- 

Last week there was a thread about getting the selected items in an open folder window. This code does that. It may not be of interest to all that many people, but it's quite interesting stuff for anyone who likes to fiddle around with the Shell.

This code uses two different methods to returns an item list and a selected item list for any open folder.

On Win9x (and maybe 2000?)

there's an IE-class window under the window object returned by a Shell.Windows Item. (The collection of open folder and IE windows.) On those systems, this code returns a file list and a selected items list. It can also return pointers to the document object and the ShellFolderView ActiveX control object that's on the folder.htt page in the given folder, which represents the ListView showing the actual files. (That object, in turn, can return a Shell.Folder object from its Folder property.)

A folder view and a folder's files can be dynamically scripted in any way desired, using the IE DOM and the ShellFolderView object model!

This code should also be adaptable to return the document object of any open IE instance.

On WinXP, the folder "WebView" seems to be a web view in name only. There's no IE-class window in the folder window hierarchy, and that hierarchy is entirely different from the Win9x version. (See Spy++ and/or AccExplorer.) Even though Shell.Windows still returns alleged IE objects, it's really just returning some kind of wrapper around the top level window. (A "CabinetWClass" window.) The XP code here uses Active Accessibility to get a file list and selected files list. It can't return an IE document, unfortunately, because there's no IE involved. (I guess that answers why folder.htt doesn't work on XP.)

This code is a rough draft, tested so far on 98, ME, and

## Returning selected items in an open folder

XP SP1. It needs Active Accessibility (oleacc) which is not installed on Win98 but can be.

---

Code in a form with a button named B1

---

```
Private Sub B1_Click()  
dim s1 as string  
Dim boo As Boolean  
'-- s1 is a unique string that's part of a specific open folder path.  
boo = GetFolder(s1)  
End Sub  
  
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)  
Set FolView = Nothing  
Set Doc1 = Nothing  
End Sub
```

---

code in a bas module

---

```
'-- References:  
' oleacc.dll# Accessibility  
' olelib.tlb# Edanmo 's OLE interfaces & functions v1.6  
' Microsoft Internet Controls  
' Microsoft Shell Controls And Automation  
' MSHTML.tlb#Microsoft HTML Object Library
```

```
'----- general windows ops -----
```

```
Public Const SMTO_ABORTIFHUNG = &H2
```

```
Public Declare Function GetClassName Lib "user32.dll" Alias "GetClassNameA"  
(ByVal hWnd As Long, ByVal lpClassName As String, ByVal nMaxCount As Long)  
As Long  
Public Declare Function EnumChildWindows Lib "user32" (ByVal hWndParent As  
Long, ByVal lpEnumFunc As Long, lParam As Long) As Long  
Public Declare Function RegisterWindowMessage Lib "user32" Alias  
"RegisterWindowMessageA" (ByVal lpString As String) As Long  
Public Declare Function SendMessageTimeout Lib "user32" _  
Alias "SendMessageTimeoutA" (_  
ByVal hWnd As Long, _  
ByVal msg As Long, _  
ByVal wParam As Long, _  
lParam As Any, _  
ByVal fuFlags As Long, _  
ByVal uTimeout As Long, _
```

Returning selected items in an open folder

## Returning selected items in an open folder

lpdwResult As Long) As Long

'-- oleacc -----

```
Private Const OBJID_WINDOW = &H0&
Private Const STATE_SELECTED = &H2&
Private Const STATE_FOCUSED = &H4&
Private Const STATE_INVISIBLE = &H8000&
Private Const ROLE_LIST = &H21&
Private Const ROLE_LISTITEM = &H22&
```

```
Public Declare Function ObjectFromLresult Lib "oleacc" (ByVal IResult As
Long, riid As UUID, ByVal wParam As Long, ppvObject As Any) As Long
Private Declare Function AccessibleObjectFromWindow Lib "oleacc" (ByVal hWnd
As Long, ByVal dwId As Long, riid As UUID, ppvObject As Object) As Long
Private Declare Function AccessibleChildren Lib "oleacc" (ByVal
paccContainer As IAccessible, ByVal iChildStart As Long, ByVal cChildren As
Long, rgvarChildren As Variant, pcObtained As Long) As Long
```

```
Public Doc1 As IHTMLDocument2 '-- document object of folder window.
Public FolView As WebViewFolderContents '-- file list control in
folder.htm.
Private AFils() As String '-- 9x: array of files.
Private AFols() As String '--9x: array of folders.
Private ASel() As String '-- array of selected items.
Private AAllItems() As String '-- array of all items.
```

```
Private sFolPath As String '-- holds path of this folder.
Private SelCnt As Long '-- number of selected items.
Private ItemCnt As Long, FilCnt As Long, FolCnt As Long
Private BooDoc As Boolean ' 9x: Doc object avail.?
Private BooFL As Boolean ' 9x: FileList object avail.?
Private BooXP As Boolean '-- XP: success getting file list.?
```

```
Private AWins() As Long
Private APaths() As String
Private WinCount As Long
```

```
Public HCur As Long
Public sClassName As String
```

'called from form.

```
Public Function GetFolder(sVal As String) As Boolean
Dim ret As Variant
Dim i2 As Long, H1 As Long
Dim sPath1 As String, s2 As String
ret = GetOpenWindows
If ret = 0 Then Exit Function
```

```
For i2 = 0 To ret - 1
If InStr(1, APaths(i2), sVal, 1) > 0 Then
```

Returning selected items in an open folder

## Returning selected items in an open folder

```
sPath1 = APaths(i2)
H1 = AWins(i2)
Exit For
End If
Next
'-- get folder info. given explorer window handle.
If H1 <> 0 Then Init H1, sPath1

MsgBox "GotDoc: " & BooDoc & vbCrLf & "GotList: " & BooFL & vbCrLf &
"GotXP: " & BooXP

If ItemCnt > 0 Then
For i2 = 0 To ItemCnt - 1
s2 = s2 & AItems(i2) & vbCrLf
Next
End If
MsgBox "Item Count: " & ItemCnt & vbCrLf & s2

s2 = ""
If SelCnt > 0 Then
For i2 = 0 To SelCnt - 1
s2 = s2 & ASel(i2) & vbCrLf
Next
MsgBox "SelCount: " & SelCnt & vbCrLf & s2
End If
GetFolder = True
End Function

'-- Use Shell.Windows to get all open explorer windows hWnd and path.
'-- According to the docs, a Windows collection item is an IE instance,
'-- but that's not really true. On Win9x it's the parent window
'-- of an IE instance. Most of the IE properties are not available.
'-- On XP, Web View is Web View in name only. There is no sub IE-class
'-- window in the window hierarchy of an explorer window.
Public Function GetOpenWindows() As Variant
Dim SHP As Shell
Dim Wins As Object
Dim ie1 As IWebBrowserApp
Dim SV As ShellFolderView
Dim i2 As Long, i3 As Long
Dim sPath1 As String
Err.Clear
' On Error Resume Next
Clear
GetOpenWindows = CVar(0)
Set SHP = New Shell
Set Wins = SHP.Windows
If Wins.Count = 0 Then
Set Wins = Nothing
Set SHP = Nothing
Exit Function
```

## Returning selected items in an open folder

```
End If

i3 = Wins.Count - 1
ReDim AWins(i3) As Long
ReDim APaths(i3) As String
i2 = 0
For Each ie1 In Wins
sPath1 = ie1.LocationURL
CleanPath sPath1
If (FolderExists(sPath1) = True) Then
AWins(i2) = ie1.hWnd
APaths(i2) = sPath1
i2 = i2 + 1
End If
Next

Set ie1 = Nothing
Set Wins = Nothing
Set SHP = Nothing

WinCount = i2
GetOpenWindows = CVar(WinCount)

End Function

Public Sub Init(ByVal H1 As Long, sPathIn As String)
Dim Boo1 As Boolean
Dim LRet As Long, HOut As Long

sFolPath = sPathIn
HOut = H1
Set Doc1 = GetIEDoc(HOut, Boo1)
BooDoc = Boo1
If BooDoc = True Then
BooFL = GetFileList
Exit Sub
End If
'-- if boofl = true then all set. otherwise, have to go the
XP/Act.Acc. way.
HOut = H1
LRet = GetListXP(HOut)
If LRet = 0 Then
BooXP = True
Else
BooXP = False
MsgBox LRet
End If
End Sub

Public Sub Clear()
Erase AWins
```

## Returning selected items in an open folder

```
Erase APaths  
WinCount = 0  
End Sub
```

```
'----- Win9x (and 2000?) -----
```

```
'-- Get document object from hWnd of open window.  
'-- This only works if there's a sub window of class "Internet  
Explorer_Server"  
Private Function GetIEDoc(ByVal H1 As Long, Success As Boolean) As  
IHTMLDocument  
Dim IID_IHTMLDocument2 As UUID  
Dim LMsg As Long, LRes As Long, LRet As Long, H2 As Long  
Success = False  
HCur = 0 '-- tracks child window enumeration.  
sClassName = "DefView"  
LRet = EnumChildWindows(H1, AddressOf EnumChildProc, 0)  
If (HCur = 0) Then Exit Function  
  
H2 = HCur  
HCur = 0  
sClassName = "Explorer_Server" "Internet Explorer_Server"  
  
LRet = EnumChildWindows(H2, AddressOf EnumChildProc, 0)  
If (HCur = 0) Then Exit Function  
  
H2 = HCur  
LMsg = RegisterWindowMessage("WM_HTML_GETOBJECT")  
LRet = SendMessageTimeout(H2, LMsg, 0, 0, SMTO_ABORTIFHUNG, 1000, LRes)  
If LRes = 0 Then Exit Function  
  
With IID_IHTMLDocument2  
.Data1 = &H332C4425  
.Data2 = &H26CB  
.Data3 = &H11D0  
.Data4(0) = &HB4  
.Data4(1) = &H83  
.Data4(2) = &H0  
.Data4(3) = &HC0  
.Data4(4) = &H4F  
.Data4(5) = &HD9  
.Data4(6) = &H1  
.Data4(7) = &H19  
End With  
  
LRet = ObjectFromLresult(LRes, IID_IHTMLDocument2, 0, GetIEDoc)  
If LRet = 0 Then Success = True  
End Function  
  
Private Function GetFileList() As Boolean  
Dim LRet As Long, i2 As Long, iFils As Long, iFols As Long
```

Returning selected items in an open folder

## Returning selected items in an open folder

```
Dim FIs As shell32.FolderItems
'Dim FI As Item
Dim oFol As shell32.Folder
Dim s2 As String

GetFileList = False
If Doc1 Is Nothing Then Exit Function
Set FolView = Doc1.All.Item("FileList")
If FolView Is Nothing Then Exit Function

Set FIs = FolView.SelectedItems
SelCnt = FIs.Count
If (SelCnt > 0) Then
ReDim ASel(SelCnt - 1)
i2 = 0
For Each Item In FIs
ASel(i2) = Item.Name
i2 = i2 + 1
Next
End If
Set FIs = Nothing

Set oFol = FolView.Folder
ItemCnt = oFol.Items.Count
iFils = 0
iFols = 0
If (ItemCnt > 0) Then
ReDim AAItems(0 To ItemCnt - 1)
i2 = 0
Set FIs = oFol.Items
For Each Item In FIs
s2 = Item.Name
AAItems(i2) = s2
If Item.IsFolder Then
ReDim Preserve AFols(iFols) As String
AFols(iFols) = s2
iFols = iFols + 1
Else
ReDim Preserve AFils(iFils) As String
AFils(iFils) = s2
iFils = iFils + 1
End If
i2 = i2 + 1
Next
Set FIs = Nothing
FilCnt = iFils
FolCnt = iFols
End If
Set oFol = Nothing
GetFileList = True
End Function
```

Returning selected items in an open folder

## Returning selected items in an open folder

```
'-- ===== WinXP =====

Private Function GetListXP(ByVal H1 As Long) As Long
Dim HList As Long, AccRet As Long
Dim Boo2 As Boolean
GetListXP = -6 ' didn't get listview window handle.
HList = GetListView(H1)
If HList = 0 Then Exit Function '--failed to get listview hWnd.

AccRet = GetXPAcc(HList)
GetListXP = AccRet

End Function

Private Function GetXPAcc(ByVal H1 As Long) As Long
Dim LRet As Long, CNum As Long, LStart As Long, LNumRet As Long, LState As
Long
Dim i2 As Long, iSel As Long
Dim Ob1 As IAccessible, Ob2 As IAccessible, Ob3 As IAccessible
Dim Boo1 As Boolean
Dim sName As String
Dim AV() As Variant
Dim AccID As UUID

GetXPAcc = -1
With AccID '-- iaccessible.
.Data1 = &H618736E0
.Data2 = &H3C3D
.Data3 = &H11CF
.Data4(0) = &H81
.Data4(1) = &HC
.Data4(2) = &H0
.Data4(3) = &HAA
.Data4(4) = &H0
.Data4(5) = &H38
.Data4(6) = &H9B
.Data4(7) = &H71
End With

LRet = AccessibleObjectFromWindow(ByVal H1, OBJID_WINDOW, AccID, Ob1)
If (LRet <> 0) Then Exit Function

GetXPAcc = -2
CNum = Ob1.accChildCount
If (CNum = 0) Then GoTo EndOps

GetXPAcc = -3
Boo1 = GetSubListObject(Ob1, CNum, Ob2)
If Boo1 = False Then GoTo EndOps
```

## Returning selected items in an open folder

```
CNum = Ob2.accChildCount
LStart = 0
LNumRet = 0
ReDim AV(0 To CNum - 1) As Variant
'--- now get the IAccessible objects for the window children:
GetXPAcc = -4
LRet = AccessibleChildren(Ob2, LStart, CNum, AV(0), LNumRet)
If (LRet <> 0) Or (LNumRet = 0) Then GoTo EndOps

ReDim AItems(0 To LNumRet - 1) As String
iSel = 0
For i2 = 0 To (LNumRet - 1)
If IsObject(AV(i2)) Then
Set Ob3 = AV(i2)
AItems(i2) = Ob3.accName
LState = Ob3.accState
MsgBox LState
If (LState And STATE_SELECTED) = STATE_SELECTED Then
ReDim Preserve ASel(0 To iSel) As String
ASel(iSel) = Ob3.accName
iSel = iSel + 1
End If
Set Ob3 = Nothing
Else
AItems(i2) = Ob2.accName(AV(i2))
LState = Ob2.accState(AV(i2))
MsgBox LState
If (LState And STATE_SELECTED) = STATE_SELECTED Then
ReDim Preserve ASel(0 To iSel) As String
ASel(iSel) = AItems(i2)
iSel = iSel + 1
End If
End If
Next

GetXPAcc = 0
ItemCnt = LNumRet - 1
SelCnt = iSel
EndOps:
Set Ob2 = Nothing
Set Ob1 = Nothing

End Function

'--- Get listview child of SysListView32 window that has list items.
'--- This returns the list iAccessible object to GetXPAcc, which
'--- then goes down to the next level to get the file/folder elements.
Private Function GetSubListObject(Ob As IAccessible, ByVal LNum As Long,
ObRet As IAccessible) As Boolean
Dim LRet As Long, LNumRet As Long, LState As Long
Dim LStart As Long, i As Long, LRole As Long
```

Returning selected items in an open folder

## Returning selected items in an open folder

```
Dim Ob4 As IAccessible
Dim AV() As Variant

'On Error Resume Next
GetSubListObject = False
LStart = 0
LNumRet = 0
ReDim AV(0 To LNum - 1) As Variant
LRet = AccessibleChildren(Ob, LStart, LNum, AV(0), LNumRet)
If (LRet = 0) And (LNumRet > 0) Then
For i = 0 To (LNumRet - 1)
If IsObject(AV(i)) Then
Set Ob4 = AV(i)
LRole = CLng(Ob4.accRole)
If (LRole = 33) Then 'role_list
Set ObRet = Ob4
Exit For
End If
Else
LRole = CLng(Ob.accRole(AV(i)))
If (LRole = 33) Then
Set ObRet = Ob
Exit For
End If
End If
Next
Set Ob4 = Nothing
If IsObject(ObRet) Then GetSubListObject = True
End If
End Function

'-- go down through explorer window in XP to get listview window handle.
Private Function GetListView(ByVal H1 As Long) As Long
Dim LRet As Long, H2 As Long
HCur = 0
GetListView = 0
sClassName = "DefView" 'SHELLDLL_DefView - one of the child windows
under handle returned
' from Shell.Windows item hWnd.
LRet = EnumChildWindows(H1, AddressOf EnumChildProc, 0)
If (HCur = 0) Then Exit Function

H2 = HCur
HCur = 0
sClassName = "DUIViewWnd" 'DUIViewWndClassName
LRet = EnumChildWindows(H2, AddressOf EnumChildProc, 0)
If (HCur = 0) Then Exit Function

H2 = HCur
HCur = 0
sClassName = "DirectUI" 'DirectUIHWND
```

## Returning selected items in an open folder

```
LRet = EnumChildWindows(H2, AddressOf EnumChildProc, 0)
If (HCur = 0) Then Exit Function
```

```
H2 = HCur
HCur = 0
sClassName = "CtrlNotify" 'CtrlNotifySink
LRet = EnumChildWindows(H2, AddressOf EnumChildProc, 0)
If (HCur = 0) Then Exit Function
```

```
H2 = HCur
HCur = 0
sClassName = "SysListView" 'SysListView32
LRet = EnumChildWindows(H2, AddressOf EnumChildProc, 0)
```

```
GetListView = HCur '--- either 0 or handle of listview window.
End Function
```

'--- callback for function above.

```
Public Function EnumChildProc(ByVal hWnd As Long, lParam As Long) As Long
Dim s2 As String
s2 = GetWinClass(hWnd)
If InStr(1, s2, sClassName, 1) > 0 Then
HCur = hWnd
EnumChildProc = 0
Else
EnumChildProc = 1
End If
End Function
```

```
Public Function GetWinClass(ByVal H1 As Long) As String
Dim sBuf As String
Dim LRet As Long
On Error Resume Next
GetWinClass = ""
sBuf = String$(256, 0)
LRet = GetClassName(H1, sBuf, Len(sBuf))
If (LRet > 0) Then GetWinClass = Left$(sBuf, LRet)
End Function
```

```
Private Sub CleanPath(ByRef sPathIn As String)
Dim Pt1 As Long
sPathIn = Replace(sPathIn, "/", "\")
sPathIn = Replace(sPathIn, "%20", " ") '--- fix spaces.
Pt1 = InStrRev(sPathIn, ":")
If Pt1 > 2 Then sPathIn = Right$(sPathIn, (Len(sPathIn) - (Pt1 - 2)))
'--- clip "file://"
End Sub
```

```
Private Function FolderExists(ByVal sPath As String) As Boolean
On Error Resume Next
```

Returning selected items in an open folder

## Returning selected items in an open folder

```
FolderExists = (GetAttr(sPath) And vbDirectory) = vbDirectory  
Err.Clear  
End Function
```

----- end code -----

.