

Re: OLE Object– the real question

Source: <http://www.tech–archive.net/Archive/Access/microsoft.public.access.forms/2005–03/0004.html>

From: Mark (*Pseudo*)

Date: 02/27/05

Date: Sun, 27 Feb 2005 16:41:37 -0600

Michelle,

Long ago, I was given a bunch of code which was written back in the Access97 days, which uses calls to the Windows components to open the file dialog.

I'm pasting it here, but be prepared for a long bunch of code! Just copy what's between the *****start copy***** and *****end copy*****. Paste it into a new module; mine was called modFileRequestor...

For your image_click event use this code instead of what I had before:

```
GetOpenFile(, "Browse for a file")
```

As for confusing names like table "Animals" and form "Animals", I always put a three–character identifier on all my objects— tblAnimals or frmAnimals or qryAnimals, etc. Just for clarity, you might want to rename your field in the table to "ImagePath", and maybe your image control on the form to "imgPicture". You shouldn't have to change the image property from "embedded"; it won't matter since it's not being saved with the form. If you still get the error about "can't find the field..." double–check your form's recordsource to make sure the new path field is in there.

*****start copy*****

Option Compare Database

Option Explicit

'For Browse Directory dialog

Private Type BROWSEINFO

hOwner As Long

pidlRoot As Long

pszDisplayName As String

lpszTitle As String

ulFlags As Long

lpfn As Long

lParam As Long

iImage As Long

End Type

'For File Open/Save dialog

Private Type tagOPENFILENAME

lStructSize As Long

hwndOwner As Long

```
hInstance As Long
strFilter As String
strCustomFilter As String
nMaxCustFilter As Long
nFilterIndex As Long
strFile As String
nMaxFile As Long
strFileTitle As String
nMaxFileTitle As Long
strInitialDir As String
strTitle As String
Flags As Long
nFileOffset As Integer
nFileExtension As Integer
strDefExt As String
lCustData As Long
lpfnHook As Long
lpTemplateName As String
End Type
```

'For File Open/Save dialog

```
Private Declare Function aht_apiGetOpenFileName Lib "comdlg32.dll" _
    Alias "GetOpenFileNameA" (OFN As tagOPENFILENAME) As Boolean
Private Declare Function aht_apiGetSaveFileName Lib "comdlg32.dll" _
    Alias "GetSaveFileNameA" (OFN As tagOPENFILENAME) As Boolean
Private Declare Function CommDlgExtendedError Lib "comdlg32.dll" () As Long
'For Browse Directory dialog
Private Declare Function SHGetPathFromIDList Lib "shell32.dll" Alias _
    "SHGetPathFromIDListA" (ByVal pidl As Long, _
    ByVal pszPath As String) As Long
Private Declare Function SHBrowseForFolder Lib "shell32.dll" Alias _
    "SHBrowseForFolderA" (lpBrowseInfo As BROWSEINFO) _
    As Long
```

'For Browse Directory dialog

```
Private Const BIF_RETURNONLYFSDIRS = &H1
'For File Open/Save dialog
Private Const ahtOFN_READONLY = &H1
Private Const ahtOFN_OVERWRITEPROMPT = &H2
Private Const ahtOFN_HIDEREADONLY = &H4
Private Const ahtOFN_NOCHANGEDIR = &H8
Private Const ahtOFN_SHOWHELP = &H10
Private Const ahtOFN_NOVALIDATE = &H100
Private Const ahtOFN_ALLOWMULTISELECT = &H200
Private Const ahtOFN_EXTENSIONDIFFERENT = &H400
Private Const ahtOFN_PATHMUSTEXIST = &H800
Private Const ahtOFN_FILEMUSTEXIST = &H1000
Private Const ahtOFN_CREATEPROMPT = &H2000
Private Const ahtOFN_SHAREAWARE = &H4000
Private Const ahtOFN_NOREADONLYRETURN = &H8000
Private Const ahtOFN_NOTESTFILECREATE = &H10000
```

```
Private Const ahtOFN_NONETWORKBUTTON = &H20000
Private Const ahtOFN_NOLONGNAMES = &H40000
Private Const ahtOFN_EXPLORER = &H80000
Private Const ahtOFN_NODEREFERENCELINKS = &H100000
Private Const ahtOFN_LONGNAMES = &H200000
```

```
*****
*****
*****
***** Code Start *****
```

```
'This code was originally written by Terry Kreft.
'It is not to be altered or distributed,
'except as part of an application.
'You are free to use it in any application,
'provided the copyright notice is left unchanged.
'
'Code courtesy of
'Terry Kreft
```

```
Function BrowseFolder(szDialogTitle As String) As String
Dim x As Long, bi As BROWSEINFO, dwIList As Long
Dim szPath As String, wPos As Integer
```

```
With bi
.hOwner = hWndAccessApp
.lpszTitle = szDialogTitle
.ulFlags = BIF_RETURNONLYFSDIRS
End With
```

```
dwIList = SHBrowseForFolder(bi)
szPath = Space$(512)
x = SHGetPathFromIDList(ByVal dwIList, ByVal szPath)
```

```
If x Then
wPos = InStr(szPath, Chr(0))
BrowseFolder = Left$(szPath, wPos - 1)
Else
BrowseFolder = vbNullString
End If
```

```
End Function
***** Code End *****
```

```
*****
*****
*****
```

```
*****
*****
*****
***** Code Start *****
```

```
'This code was originally written by Ken Getz.
'It is not to be altered or distributed,
```

'except as part of an application.
'You are free to use it in any application,
'provided the copyright notice is left unchanged.
'

' Code courtesy of:
' Microsoft Access 95 How–To
' Ken Getz and Paul Litwin
' Waite Group Press, 1996

Function GetOpenFile(Optional varDirectory As Variant, _
Optional varTitleForDialog As Variant, Optional varFiletype As Variant)
As Variant

' Here's an example that gets an Access database name.

Dim strFilter As String

Dim lngFlags As Long

Dim varFileName As Variant

' Specify that the chosen file must already exist,

' don't change directories when you're done

' Also, don't bother displaying

' the read–only box. It'll only confuse people.

lngFlags = ahtOFN_FILEMUSTEXIST Or _

ahtOFN_HIDEREADONLY Or ahtOFN_NOCHANGEDIR

If IsMissing(varDirectory) Then

varDirectory = ""

End If

If IsMissing(varTitleForDialog) Then

varTitleForDialog = ""

End If

' Define the filter string and allocate space in the "c"

' string Duplicate this line with changes as necessary for

' more file templates.

strFilter = ahtAddFilterItem(strFilter, _
"Pictures (*.jpg,*.bmp)", "*.jpg;*.bmp")

' Now actually call to get the file name.

varFileName = ahtCommonFileOpenSave(_

OpenFile:=True, _

InitialDir:=varDirectory, _

Filter:=strFilter, _

Flags:=lngFlags, _

DialogTitle:=varTitleForDialog)

If Not IsNull(varFileName) Then

varFileName = TrimNull(varFileName)

End If

GetOpenFile = varFileName

End Function

Function ahtCommonFileOpenSave(_

Optional ByRef Flags As Variant, _

Optional ByVal InitialDir As Variant, _

Optional ByVal Filter As Variant, _

```
Optional ByVal FilterIndex As Variant, _
Optional ByVal DefaultExt As Variant, _
Optional ByVal FileName As Variant, _
Optional ByVal DialogTitle As Variant, _
Optional ByVal hwnd As Variant, _
Optional ByVal OpenFile As Variant) As Variant
' This is the entry point you'll use to call the common
' file open/save dialog. The parameters are listed
' below, and all are optional.
'
' In:
' Flags: one or more of the ahtOFN_* constants, OR'd together.
' InitialDir: the directory in which to first look
' Filter: a set of file filters, set up by calling
' AddFilterItem. See examples.
' FilterIndex: 1-based integer indicating which filter
' set to use, by default (1 if unspecified)
' DefaultExt: Extension to use if the user doesn't enter one.
' Only useful on file saves.
' FileName: Default value for the file name text box.
' DialogTitle: Title for the dialog.
' hwnd: parent window handle
' OpenFile: Boolean(True=Open File/False=Save As)
' Out:
' Return Value: Either Null or the selected filename
Dim OFN As tagOPENFILENAME
Dim strFilename As String
Dim strFileTitle As String
Dim fResult As Boolean
' Give the dialog a caption title.
If IsMissing(InitialDir) Then InitialDir = CurDir
If IsMissing(Filter) Then Filter = ""
If IsMissing(FilterIndex) Then FilterIndex = 1
If IsMissing(Flags) Then Flags = 0&
If IsMissing(DefaultExt) Then DefaultExt = ""
If IsMissing(FileName) Then FileName = ""
If IsMissing(DialogTitle) Then DialogTitle = ""
If IsMissing(hwnd) Then hwnd = Application.hWndAccessApp
If IsMissing(OpenFile) Then OpenFile = True
' Allocate string space for the returned strings.
strFilename = Left(FileName & String(256, 0), 256)
strFileTitle = String(256, 0)
' Set up the data structure before you call the function
With OFN
    .lStructSize = Len(OFN)
    .hwndOwner = hwnd
    .strFilter = Filter
    .nFilterIndex = FilterIndex
    .strFile = strFilename
    .nMaxFile = Len(strFilename)
    .strFileTitle = strFileTitle
```

```
.nMaxFileTitle = Len(strFileTitle)
.strTitle = DialogTitle
.Flags = Flags
.strDefExt = DefaultExt
.strInitialDir = InitialDir
' Didn't think most people would want to deal with
' these options.
.hInstance = 0
.strCustomFilter = ""
.nMaxCustFilter = 0
.lpfHook = 0
' New for NT 4.0
.strCustomFilter = String(255, 0)
.nMaxCustFilter = 255
End With
' This will pass the desired data structure to the
' Windows API, which will in turn it uses to display
' the Open/Save As Dialog.
If OpenFile Then
    fResult = aht_apiGetOpenFileName(OFN)
Else
    fResult = aht_apiGetSaveFileName(OFN)
End If

' The function call filled in the strFileTitle member
' of the structure. You'll have to write special code
' to retrieve that if you're interested.
If fResult Then
    ' You might care to check the Flags member of the
    ' structure to get information about the chosen file.
    ' In this example, if you bothered to pass in a
    ' value for Flags, we'll fill it in with the outgoing
    ' Flags value.
    If Not IsMissing(Flags) Then Flags = OFN.Flags
    ahtCommonFileOpenSave = TrimNull(OFN.strFile)
Else
    ahtCommonFileOpenSave = vbNullString
End If
End Function

Function ahtAddFilterItem(strFilter As String, _
    strDescription As String, Optional varItem As Variant) As String
' Tack a new chunk onto the file filter.
' That is, take the old value, stick onto it the description,
' (like "Databases"), a null character, the skeleton
' (like "*.mdb;*.mda") and a final null character.

    If IsMissing(varItem) Then varItem = ".*"
    ahtAddFilterItem = strFilter & _
        strDescription & vbNullChar & _
        varItem & vbNullChar
```

End Function

```
Private Function TrimNull(ByVal strItem As String) As String
```

```
Dim intPos As Integer
```

```
intPos = InStr(strItem, vbNullChar)
```

```
If intPos > 0 Then
```

```
TrimNull = Left(strItem, intPos - 1)
```

```
Else
```

```
TrimNull = strItem
```

```
End If
```

```
End Function
```

```
***** Code End *****
```

```
*****
```

```
*****
```

```
*****
```

```
***end copy***
```

"Michelle" <Michelle@discussions.microsoft.com> wrote in message
news:0e3a01c51d18\$1ee4d120\$a501280a@phx.gbl...

> *Mark, Thanks! I appreciate the detail steps. And it's*

> *embarrassing when I can't follow them!*

> *I'm getting an error message: run-time error '2465'*

> *Microsoft Access can't find the field 'FilePath' referred*

> *to in your expresion. I can't get from design view to*

> *form view because of it.*

>

> *On my Animals table I added FilePath as a text field. And*

> *in my Animals form (confusing names, I know) I added the*

> *picture (which I had to put a box around so I could find*

> *it!), deleted the actual picture/path and called the*

> *field: ImageControl. Should the picture type be "linked"*

> *rather than "embedded"?*

>

> *Private Sub Form_Current()*

> *Me.ImageControl.Picture = Me!FilePath*

> *End Sub*

>

> *I don't understand why this didn't work.*

>

> *So I commented it out, and then I get a compile error:*

> *expected variable or procedure, not module*

> *and remembered to name the module: GetTheFilenamemdl*

> *rather than GetTheFilename. See! I can be taught.*

>

> *Now, it's not liking*

> *Dim dlgFilePick As FileDialog*

> *I've never seen FileDialog. This must be what you*

> *mentioned as tweaking the file browser function as I'm*

> *using Access 2000. So I took a peek in the library and I*

> *have these checked (and in this order):*

> *VB for applications*

> *Microsoft Access 9.0 object library*
> *OLE Automation*
> *Microsoft DAO 3.6 object library*
> *Microsoft ActiveX Sata Objects 2.1 library*
>
> *Am I salvagable? Thanks for your help, Michelle*
>
>>-----Original Message-----
>>Hi Michelle,
>>
>>For testing, you can certainly just type in the full path
> (don't need
>>quotes) to the image (don't forget the file's extension
> like Dog1.jpg). On
>>the form, add an image and select any old image from the
> wizard. Once the
>>control is on the form, you can go to properties and
> erase the file path in
>>the "Picture" property. Now you have a blank (unbound)
> image control.
>>
>>In my test form I whipped up to refresh my memory, I just
> used the default
>>name of the control, which happened to be "Image7".
> Since the file path is
>>saved with the record, you'll always get the correct
> image for the record
>>being displayed.
>>
>>When your user clicks on the image (or blank image box if
> there's no image
>>assigned yet), the full filename with path is stored in
> the FilePath field
>>for that record and the form's recordset is requeried to
> update the image.
>>The ImageControl's OnClick event:
>>'*****code start*****'
>>Private Sub ImageControl_Click()
>> Dim strFilename As String, strFind As String
>> Dim lngID As Long
>>
>> strFilename = GetTheFilename("Browse to the file")
>> If strFilename = "Cancelled" Then
>> Exit Sub
>> End If
>> Me!FilePath = strFilename
>> lngID = Me!ItemID
>> strFind = "[ItemID]=" & lngID
>> Echo False
>> Me.Requery
>> Me.Recordset.FindFirst strFind

```
>> Echo True
>>
>> *****code end*****
>>
>> In a regular module (for the file requestor),
>> *****code start*****
>> Function GetTheFilename(strTitle As String) As String
>> Dim strFilename As String
>> Dim dlgFilePick As FileDialog
>> Dim vrtSelectedItem As Variant
>>
>> On Error GoTo Err_GetFilename
>> Set dlgFilePick = Application.FileDialog
> (msoFileDialogFilePicker)
>> With dlgFilePick
>> .Title = strTitle
>> 'Let user select only one file
>> .AllowMultiSelect = False
>> If .Show = -1 Then
>> 'The user pressed the action button.
>> GetTheFilename = .SelectedItem(1)
>>
>> Else
>> 'The user pressed Cancel.
>> GetTheFilename = "Cancelled"
>> End If
>> End With
>>
>> Exit_GetFilename:
>> Set dlgFilePick = Nothing
>> Exit Function
>> Err_GetFilename:
>> MsgBox "Error " & Err.Number & ": " &
> Err.Description, ,
>> "GetTheFilename"
>> Resume Exit_GetFilename
>> End Function
>> *****code end*****
>>
>> I'm using Access2003, so the file browser function might
> need tweaking for
>> older versions.
>> Hope this helps!
>>
>> "Michelle" <Michelle@discussions.microsoft.com> wrote in
> message
>> news:1eae01c51d08$341f2f50$a401280a@phx.gbl...
>>> I like the sound of that! I only have 5 pictures so far
>>> and they are slow to load.
>>>
>>> I want to try it! I created a new text field (FilePath)
```

> in
>>> *table Animals. My file path for the 1st picture is*
>>> *C:\My Documents\My Pictures\Dog1*
>>> *so I enter the above path in the Animals table, FilePath*
>>> *field for Dog1's record. Correct? Do I need the file*
>>> *path in quotes?*
>>>
>>> *How do I create an unbound image control on the form? I*
>>> *see the tool bar offers "image", "unbound object frame"*
>>> *and a "bound object frame"– all of which want me to*
> *select*
>>> *the image now. What am I missing to get an unbound*
> *image*
>>> *control?*
>>>
>>> *I'm confused about: Me.Image7.Picture = Me!FilePath*
>>> *Does Image7 mean something? Is that the name of your*
>>> *image in the record? If so, how does the record know*
>>> *which picture to show for which record– must be by the*
>>> *file path name...*
>>>
>>> *You said "When you click the image control, the user*
>>> *browses to the picture they want to assign to the*
> *current*
>>> *record."– that sounds cool! So I set up the unbound*
> *image*
>>> *control once in the table and then the user gets to*
> *choose*
>>> *the picture, so that the user is never in the table–*
>>> *correct? That would be excellent!*
>>>
>>> *Thanks for the guidance!*
>>> *Michelle*
>>>
>>>>-----Original Message-----
>>>>*I used to have an ID badge database that stored the*
>>>>*digital pictures in an*
>>>>*OLE field. There was an OLE control on the form which,*
>>>>*when clicked, would*
>>>>*let you browse to select the file. With only 100 or so*
>>>>*employee records, the*
>>>>*database bloated to over 200Mb. I have since modified*
>>>>*the database to get*
>>>>*rid of the OLE field and control. I'm now storing the*
>>>>*path to the picture*
>>>>*in a text field called FilePath and the database is*
> *under*
>>>>*200Kb. I've got*
>>>>*an unbound image control on the form, and in the form's*
>>>>*OnCurrent event I*
>>>>*have:*

>>>>
>>>>Me.Image7.Picture = Me!FilePath
>>>>
>>>>which will change the picture to match the picture you
>>> assigned to the
>>>>current record. When you click the image control, the
>>> user browses to the
>>>>picture they want to assign to the current record. It
>>> works with both JPG
>>>>and BMP files.
>>>>
>>>>"Michelle" <Michelle@discussions.microsoft.com> wrote in
>>> message
>>>>news:1e2501c51ce8\$2bf6eb80\$a401280a@phx.gbl...
>>>>> Sorry about that!
>>>>> I have form that I want to show a digital camera
>>> picture.
>>>>> I have the table field set up as an OLE Object.
>>>>>
>>>>> Can I load the picture from the form, or do I need to
> go
>>>>> to the table, find the correct record for the field,
>>>>> select Insert object and the browse to the correct
>>> digital
>>>>> camera picture? There's got to be an easier way for
>>> end–
>>>>> users!
>>>>>
>>>>> I see that bitmap is an option, but my digital camera
>>>>> pictures are jpg. How do I upload these?
>>>>>
>>>>> Thanks,
>>>>> Mich
>>>>
>>>>
>>>>.br/>>>>>
>>>
>>
>>
>>.br/>>>