Common Dialog Save As works in Access 32-bit but not in Acess 64-bit calling GetSaveFileName API; CommDlgExtendedError error code 1; LenB does not solve the problem

Hello,

I am writing for some help. Sorry for the length of my message but I didn't want to leave anything out. I just hope I am clear. Any help is most appreciated. Thanks in advance.

I am using a user defined common dialog box class (which I named CommonDialogAPI) to display a Save As dialog box in an Access mdb (not accdb) database. (After giving the file a name or accepting the default name I give it in code and the location to where to save the file, data (from a query) is exported and save to that file.) We have users who have Office 2010 32-bit and Office 2010 64-bit. This work without any problems for users who have Access 2010 32-bit but does not work for users who have Access 2010 64-bit.

Let me give some background:

When I run the code to open the Save As dialog box in Access 2010 32-bit, it works great and has no problems. However, when I run the same code on a computer with Access 2010 64-bit, it doesn't work as expected. The Save As file dialog box isn't displayed. However, the error code that I am getting after calling CommDlgExtendedError API is code 1. I am guessing error code 1 (or 0x0001), but I could be wrong, refers to

CDERR_STRUCTSIZE 0x0001

"The lStructSize member of the initialization structure for the corresponding common dialog box is invalid."

I am using the API called GetSaveFileName. In addition, I also used the Win32API_PtrSafe.TXT where I copied the declarations and types from.

I have made the following modifications.

1. Added the attribute PrtSafe to the Declare statements. It is required on the Access 2010 64-bit machine; otherwise you get a compile error.

2. Updated Long datatype to LongPtr datatype if it refers to a memory address.

3. Left Long datatype as is if it refers to a value.

4. Added conditional compilation #If Win54 #Else #End to distinguish if the user has 32-bit Office/Access or 64-bit Office/Access. Later, I may add a VBA7 check (which is only available in Access 2010) for users who have an earlier version of Access. For now, I just need to get this working with Access 2010 64-bit.

5. For testing purposes, I added a message box in code to show the error returned from CommDlgExtendedError API. The code is 1. I am trying to determine why the Save As file dialog box isn't shown and this error if I am not mistaken provides some incite.

Here is my question. How do I resolve the following error after calling the API CommDlgExtendedError on a 64-bit Office computer (okay on a 32-bit office computer)?

CDERR_STRUCTSIZE 0x0001

"The lStructSize member of the initialization structure for the corresponding common dialog box is invalid."

It doesn't seem to matter but I changed the following when I initialize the OpenFileName object.

```
With OpenFile
   .lStructSize = Len(OpenFile)
    ...
```

changed to

```
With OpenFile
   .lStructSize = LenB(OpenFile)
```

Here is my abbreviated code (copied and pasted) with the declare statements, type, and the call to the API from my CommondDialog class.

```
Option Compare Database

#If Win64 = True Then
'Office 2010 compatible with both 32-bit and 64-bit - for specific 64-bit office 2010 users
'updated memory addresses to longPtr
  Private Type OPENFILENAME
```

```
        lStructSize As Long
        hwndOwner As LongPtr
        hInstance As LongPtr
        lpstrFilter As String
        lpstrCustomFilter As String
        nMaxCustFilter As Long
        nFilterIndex As Long
        lpstrFile As String
        nMaxFile As Long
        lpstrFileTitle As String
        nMaxFileTitle As Long
        lpstrInitialDir As String
        lpstrTitle As String
        Flags As Long
        nFileOffset As Integer
        nFileExtension As Integer
        lpstrDefExt As String
        lCustData As Long
        lpfnHook As LongPtr
    End Type

    Private Declare PtrSafe Function GetOpenFileName Lib "comdlg32.dll" Alias "GetOpenFileNameA" _
(pOpenfilename As OPENFILENAME) As Long

    Private Declare PtrSafe Function GetSaveFileName Lib "comdlg32.dll" Alias "GetSaveFileNameA"
(pOpenfilename As OPENFILENAME) As Long

    Private Declare PtrSafe Function CommDlgExtendedError Lib "comdlg32.dll" () As Long

    Private Declare PtrSafe Function GetLastError Lib "kernel32" () As Long

#Else 'Earlier versions before Office 2010 and 32-bit office

    Private Type OPENFILENAME
        lStructSize As Long
        hwndOwner As Long
        hInstance As Long
        lpstrFilter As String
        lpstrCustomFilter As String
        nMaxCustFilter As Long
        nFilterIndex As Long
        lpstrFile As String
        nMaxFile As Long
        lpstrFileTitle As String
```

```
        nMaxFileTitle As Long
        lpstrInitialDir As String
        lpstrTitle As String
        Flags As Long
        nFileOffset As Integer
        nFileExtension As Integer
        lpstrDefExt As String
        lCustData As Long
        lpfnHook As Long
        lpTemplateName As String
    End Type


'PtrSave is required as it won't compile on a 64-bit Office computer

    Private Declare PtrSafe Function GetOpenFileName Lib "comdlg32.dll" Alias "GetOpenFileNameA" _
(pOpenfilename As OPENFILENAME) As Long

    Private Declare PtrSafe Function GetSaveFileName Lib "comdlg32.dll" Alias "GetSaveFileNameA"
(pOpenfilename As OPENFILENAME) As Long

    Private Declare PtrSafe Function CommDlgExtendedError Lib "comdlg32.dll" () As Long

    Private Declare PtrSafe Function GetLastError Lib "kernel32" () As Long
#End If


Private mstrFileName As String
Private mblnStatus As Boolean

Public Property Let GetName(strName As String)
    mstrFileName = strName
End Property

Public Property Get GetName() As String
    GetName = mstrFileName
End Property

Public Property Let GetStatus(blnStatus As Boolean)
    mblnStatus = blnStatus
End Property

Public Property Get GetStatus() As Boolean
    GetStatus = mblnStatus
```

```
End Property


Public Function SaveFileDialog(lngFormHwnd As Long, lngAppInstance As Long, strInitDir As String,
strFileFilter As String, strDefaultFileName As String) As Long

   Dim SaveFile As OPENFILENAME
   Dim X As Long

   With SaveFile
      .lStructSize = LenB(SaveFile)
      .hwndOwner = lngFormHwnd
      .hInstance = lngAppInstance
      .lpstrFilter = strFileFilter
      .nFilterIndex = 1
      .lpstrFile = String(257, 0)
      .nMaxFile = Len(SaveFile.lpstrFile) - 1
      .lpstrFileTitle = SaveFile.lpstrFile
      .nMaxFileTitle = SaveFile.nMaxFile
      .lpstrInitialDir = strInitDir
      .lpstrTitle = "Save As"
      .Flags = 0
   End With

   'update the default file name
   SaveFile.lpstrFile = strDefaultFileName & String(257 - Len(strDefaultFileName), 0) 'fixed length string
padded with nulls

   X = GetSaveFileName(SaveFile)

   If X = 0 Then
      Dim retval As Long
      retval = CommDlgExtendedError()
      MsgBox "Common Dialog errror CODE " + CStr(retval)
   End If

   If X = 0 Then
      mstrFileName = "none"
      mblnStatus = False
   Else
      'mstrFileName = Trim(SaveFile.lpstrFile)
      'revsion
      mstrFileName = Trim((Left(SaveFile.lpstrFile, InStr(1, SaveFile.lpstrFile, vbNullChar) - 1)))
      mblnStatus = True
```

```
    End If
End Function


'**************************
'This class method is unused
'**************************
Public Function OpenFileDialog(lngFormHwnd As Long, lngAppInstance As Long, strInitDir As String,
strFileFilter As String) As Long
    Dim OpenFile As OPENFILENAME
    Dim X As Long

        With OpenFile
        .lStructSize = Len(OpenFile)
        .hwndOwner = lngFormHwnd
        .hInstance = lngAppInstance
        .lpstrFilter = strFileFilter
        .nFilterIndex = 1
        .lpstrFile = String(257, 0)
        .nMaxFile = Len(OpenFile.lpstrFile) - 1
        .lpstrFileTitle = OpenFile.lpstrFile
        .nMaxFileTitle = OpenFile.nMaxFile
        .lpstrInitialDir = strInitDir
        .lpstrTitle = "Open File"
        .Flags = 0
    End With

    X = GetOpenFileName(OpenFile)
    If X = 0 Then
        mstrFileName = "none"
        mblnStatus = False
    Else
        mstrFileName = Trim(OpenFile.lpstrFile)
        mblnStatus = True
    End If
End Function
```

Lastly, I have a command button on a form that calls SaveFileDialog that passes the parameter values like.

```
lngResult = cdlg.SaveFileDialog(lngFormHwnd, _
```

```
    lngAppInstance, strInitDir, strFileFilter, strDefaultFileName)
```

I am exporting a query (named QExportToExcelCustDBUnpaid) results to an Excel file.

Here is the code and it again works in Access 2010 32-bit just not in Access 2010 64-bit.

```
Private Sub cmdSaveFile_Click()
Dim cdlg As New CommonDialogAPI

Dim lngFormHwnd As Long
Dim lngAppInstance As Long
Dim strInitDir As String
Dim strFileFilter As String
Dim lngResult As Long

Dim strDefaultFileName As String
lngFormHwnd = Me.Hwnd
lngAppInstance = Application.hWndAccessApp
'strInitDir = "C:\" 'random

strFileFilter = "Excel Files (*.xls)" & _
   Chr(0) & "*.xls" & Chr(0)

strDefaultFileName = "QExportToExcelCustDBUnpaid"
lngResult = cdlg.SaveFileDialog(lngFormHwnd, _
   lngAppInstance, strInitDir, strFileFilter, strDefaultFileName)

If cdlg.GetStatus = True Then


   DoCmd.TransferSpreadsheet acExport, acSpreadsheetTypeExcel8, "QExportToExcelCustDBUnpaid",
cdlg.GetName, False
   MsgBox "Exported QExportToExcelCustDBUnpaid to " & cdlg.GetName, vbInformation, "Tricor Main
Menu"

End If
'release resource (object variable)
If Not cdlg Is Nothing Then
   Set cdlg = Nothing
```

End If

End Sub

The problem is in the line X = GetSaveFileName(SaveFile) which returns zero. Normally, a zero is returned when the user clicks cancel, but in my case the Save File Dialog isn't being shown so I am not clicking the cancel button.

So why isn't this working? And why am I getting error code 1 after calling CommDlgExtendedError API? How to I resolve the .lStructSize if I am reading the error correctly. In the immediate (debug) window, the value of len(OpenFile) is 120 on the Access 2010 64-bit machine but len(OpenFile) is 76 on the Access 2010 32-bit machine.

If I can't get this to work, then an option is to uninstall Office 2010 64-bit and then install Office 2010 32-bit which I know will fix the problem but I wanted to see if I can resolve this issue to work with both 32-bit and 64-bit Access 2010. I believe there is a way for the code to be compatible but so far it isn't working in Access 2010 64-bit but does in Access 2010 32-bit.

Any help is most appreciated. Thanks in advance.