

Microsoft Access Wizhook Reference

Compiled by JasonM

jasonm [at] accessvba [dot] com
jason [at] ushimitsudoki [dot] com

WizHook Overview

WizHook is an internal undocumented library for the Access wizards in Access 2000 and later. You can view it and its functions in the Object Browser if you select the "Show Hidden Members" option.

I always noticed it had some interesting sounding procedure names, but there's almost no reference to it on the internet. This is my attempt at documenting it the best I can. Please understand the vast majority of this is the result of trial and error, so if you find a mistake or have a better interpretation of what's happening, please email me so I can update this document.

The biggest thing to know: **WizHook.Key = 51488399**. Some methods can be called without this key, but most of them require it. The key only needs to be set once per application session, to save space I'm not including it in any sample functions. If you don't use the key and the method requires it, the call will usually fail silently.

I ran across the key in the V-Tools 2000 download from <http://www.skrol29.com>. How he found it I do not know, but all credit for unlocking the key goes to him. His code in general is worth checking out. Most of the Script methods I try to document are based on his usage in the V-Tools oMacro module of the V-Tools 2000 application (along with the Access API declarations in the **OpenScript()** example).

Enjoy!

Last Updated: 2 December 2004

This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/2.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

AccessUserDataDir() As String	2000
Returns current user's Access data directory. Ex: C:\Documents and Settings\Jason\Application Data\Microsoft\Access\	

AccessWizFilePath(bstrWhich As String) As String	2003
Returns full path to specified Access Wizard Library.	
Example: Debug.Print WizHook.AccessWizFilePath("acwzmain.mde")	
Yields (on my system): C:\Program Files\Microsoft Office\OFFICE11\acwzmain.mde	
Tested OK with: utility.mda acwzdat.mdt acwzusr.mdt acwzlib.mde acwzmain.mde azwztool.mde	

AdpUIDPwd(pbstrUID As String, pbstrPwd As String)As String	2002

AnalyzeQuery(Workspace As Workspace, Database As Database, Query As String, Results As String) As Long	2000
<p>This is the function the Performance Analyzer uses to analyze Queries. The string Results contains the resulting information. It separates sub-strings with Chr\$(1) and ends the main string with Chr\$(2). The number of sub-strings vary according to the results of the analysis. The return value always seems to be 0.</p> <p>First sub-string is code indicating the type of response. Second sub-string is full path to database. Remaining fields vary according to first sub-string.</p> <p>Seems to be easy to match this up if you compare the results of the Performance Analyzer with the response string. If you do an in-depth analysis of the result string, please email me so I can update this documentation.</p> <p>See Also AnalyzeTable() – the code example there is easily modified for AnalyzeQuery()</p>	

AnalyzeTable(Workspace As Workspace, Database As Database, Table As String, ReturnDebugInfo As Boolean, Results As String) As Long	2000
<p>This is the function the Performance Analyzer uses to analyze Tables. The string Results contains the resulting information. It separates sub-strings with Chr\$(1) and ends the main string with Chr\$(2). If ReturnDebugInfo = True, several addition sub-strings are present, including table name and field names.</p> <p>This doesn't seem to match up as clearly with the Performance Analyzer results as AnalyzeQuery() does. If you do an in-depth analysis of the result string, please email me so I can update this documentation.</p> <p>Public Function WH_AnalyzeTable()</p>	

```

Dim dbe As PrivDBEngine
Dim db As Database
Dim wsp As Workspace

Dim IReturn As Long
Dim sResults As String
Dim sTable As String
Dim bDebugInfo As Boolean

Set dbe = New PrivDBEngine
Set wsp = dbe.CreateWorkspace("Test", "Admin", "")
Set db = wsp.OpenDatabase("test1.mdb")

sTable = "tblTest"
bDebugInfo = True

IReturn = WizHook.AnalyzeTable(wsp, db, sTable, bDebugInfo, sResults)

Debug.Print "Return value: " & IReturn
Debug.Print "Results: " & vbCrLf & sResults

Set db = Nothing
Set wsp = Nothing
Set dbe = Nothing
End Function

```

ArgsOfActid(Actid As Long) As Long	2000
---	-------------

Returns the number of arguments and ActionID expects.
An Actid is an Access Visual Basic action ID (useful in a macro).
See **NameFromActID()** and **OpenScript()**

BracketString(String As String, flags As Long) As Boolean	2000
--	-------------

Adds brackets "[]" to substrings in String, breaking String on "!" or "."
Flags:
0 – Do not add brackets
1 – Bracket sub-strings that are not legal variable names
2 – Bracket all sub-strings
Return Value:
True – Resulting string contains illegal variable names
False – Resulting string contains no illegal variable names

CloseCurrentDatabase As Boolean()	2000
--	-------------

Closes the current database. Leaves the Access application window and VBA editor open.

CreateDataPageControl(DpName As String, CtlName As String, Type As Long, Section As String, SectionType As Long, AppletCode As String, x As Long, y As Long, dx As Long, dy As Long)	
Haven't messed with this yet as I'm not too interested in Data Access Pages.	
CurrentLangID() As Long	2000
Returns current language ID. 1033 for me!	
DbcVBProject [A VBProject property]	2000
Read-only Returns the current VBProject. Doesn't need the Visual Basic for Application Extensibility reference.	
EmbedFileOnDataPage(DpName As String, FileToInsert As String) As String	2000
Haven't messed with this yet as I'm not too interested in Data Access Pages.	
EnglishPictToLocal(In As String, Out As String) As Boolean	2000
Converts format string specifier in English to Local language?	
FCreateNameMap(objtyp As Integer, bstrObjName As String) As Boolean	2003
FGetMSDE(fBlockKeys As Boolean) As Boolean	2002
FileExists(File As String) As Boolean	2000
Return value: True – If the file exists False – If the file does not exist	
FirstDbcDataObject(Name As String, ObjType As AcObjectType, Attribs As Long) As Boolean	2000
Fills Name the object name and ObjectType with the corresponding Access.AcObjectType value Not sure about Attribs. Return Value: True – found an object False – did not find object Not sure what qualifies, this seems to find the first query in alphabetical order for me.	
FIsFEWch(wch As Long) As Boolean	2002
FullPath(RelativePath As String, FullPath As String) As Integer	2000
Puts the full path of relative path into FullPath	

Return value seems to always be true (doesn't check for invalid directory)

GetAccWizRCPath

2003

Returns RC Wizard Full Path. (Microsoft Access Wizard International DLL)
(On my system: C:\Program Files\Microsoft Office\OFFICE11\1033\acwizrc.dll)

GetColumns(bstrBase As String) As String

2002

If bstrBase is a valid table or query name, lists the columns, separated by semi-colons, with a trailing semi-colon.

GetCurrentView(bstrTableName As String) As String

2002

If bstrTableName is a valid table name it seems to always return 2, no matter the current view (design, PivotTable, PivotChart, etc.)
If bstrTableName is not a valid table name it returns 0.

GetFileName(hwndOwner As Long, AppName As String, DlgTitle As String, OpenTitle As String, File As String, InitialDir As String, Filter As String, FilterIndex As Long, View As Long, flags As Long, fOpen As Boolean) As Long

2000

Putting this off until later...

GetFileName2(hwndOwner As Long, AppName As String, DlgTitle As String, OpenTitle As String, File As String, InitialDir As String, Filter As String, FilterIndex As Long, View As Long, flags As Long, fOpen As Boolean, fFileSystem) As Long

2002

Putting this off until later...

GetFileOdso(bstrExt As String, bstrFilename As String) As Long

2002

ODSO = Office Data Source Object
http://rep.oio.dk/Microsoft.com/officeschemas/html/weleodso_HV01099352.htm

GetInfoForColumns(bstrBase As String) As String

2002

Similar to GetColumns(). Returns a semi-colon delimited list of columns and attributes for a table or query.

There are 3 pieces of data for each column.

- 1 – name of column
- 2 – DAO datatype constant (dbLong = 4, dbText = 10, etc.)
- 3 – max bytes per column (?)

GetScriptString(HScr As Long, ScriptColumn As Long, Value As String) As Boolean

2000

Use OpenScript() first to get the HScr.

ScriptColumn:

- 0: Label
- 1: Comment
- 2: Condition
- 3 – 12: Arguments

Value is filled by GetScriptString()

See also example code in **OpenScript()**.

GetWizGlob(IWhich As Long)	2002
GlobalProcExists(Name As String) As Boolean	2000
Return value: True – If the current application has a global procedure matching Name False – If the current application does not have a global procedure matching Name	
HideDates	2002
IsMemberSafe(dispid As Long) As Boolean	2003
IsMatchToDbcConnectionString(bstrConnectionString As String) As Boolean [Property]	2000
Putting this off until later...	
IsValidIdentifier(Identifier As String) As Boolean	2000
Return value: True – If Identifier is a valid variable name False – If Identifier is not a valid variable name	
Key [Property]	2000
Needs to be set to a valid key to use those methods marked "Key Required" in this document. Only needs to be set once per session. Valid setting: WizHook.Key = 51488399	
KeyboardLangID() As Long	2000
Returns current language for the keyboard setting. 1033 for me!	
KnownWizLeaks(fStart As Boolean)	2000
Putting this off until later...	
LoadImexSpecSolution(bstrFilename As String)	2000
Putting this off until later...	
LocalFont() As String	2000
Returns the local font setting. I get "Tahoma".	
NameFromActid(Actid As Long) As String	2000
Returns the Access Visual Basic Action name that corresponds to ActID. Not all Visual Basic Actions have a corresponding ActID. Useful in manipulating macros, see also: ArgsOfActid() and OpenScript() Public Function WH_AnalyzeTable()	

```

Dim i As Long
i = 0
For i = 0 To 62
    Debug.Print i; ": " & WizHook.NameFromActid(i) & "(" & WizHook.ArgsOfActid(i) & ")"
Next i
End Function

```

I checked from 0 – 150,000 as values for Actid. If you check further out, please let me know so I can update this information.

ObjTypOfRecordSource(RecordSource As String) As Integer	2000
--	-------------

Returns type of recordsource:

- 0 – SQL Statement
- 1 – Table
- 2 - Query

OfficeAddInDir() As String	2000
-----------------------------------	-------------

Returns the directory for Office Add-Ins

Ex: C:\Documents and Settings\Jason\Application Data\Microsoft\AddIns\

OpenPictureFile(File As String, Cancelled As Boolean) As Boolean	2000
---	-------------

Opens the "Insert Picture" common dialog.

After the dialog is closed, File will contain the full path of the selected file and Cancelled is True if the Cancel button was clicked.

OpenScript(Script As String, Label As String, OpenMode As Long, Extra As Long, Version As Long) As Long	2000
--	-------------

Opens a macro.

Script: Macro Name Label: not sure

Open mode: 0& Extra and Version: filled by OpenScript

Return value: Hscr (handle to script), need this Hscr for other Script functions, like **SaveScriptString**, and **GetScriptString**)

Private Declare Function api_Macro_NextRow Lib "msaccess.exe" Alias "#22" (ByVal hMacro As Long, ByVal ISkipBlank As Long, IEndOfMacro As Long) As Long

Private Declare Function api_Macro_GetActID Lib "msaccess.exe" Alias "#29" (ByVal hMacro As Long) As Long

Public Function DisplayMacro(sMacroName As String)

 Dim sLabel As String

 Dim IOpenMode As Long

 Dim IExtra As Long

 Dim IVersion As Long

 Dim IHmacro As Long

 Dim IScriptColumn As Long

 Dim sValue(12) As String

 Dim IActNum As Long

 Dim IActID As Long

 Dim IMacroRow As Long

 Dim bEndOfMacro As Boolean

```

IOpenMode = 0&
IHmacro = WizHook.OpenScript(sMacroName, sLabel, IOpenMode, IExtra, IVersion)

If (IHmacro) Then
    Debug.Print "Hscr: " & IHmacro & "Macro Name: " & sMacroName
    Debug.Print "Label: " & sLabel & "Open Mode: " & IOpenMode
    Debug.Print "IExtra: " & IExtra & "IVersion: " & IVersion
Else
    Debug.Print "Failed to get handle"
Exit Function
End If

IMacroRow = 0
Do While Not (bEndOfMacro)
    Call api_Macro_NextRow(IHmacro, False, False)
    IActID = api_Macro_GetActID(IHmacro)

    If (IActID <> -1) Then
        Debug.Print "Row: " & IMacroRow & vbTab & "Action: " & WizHook.NameFromActid(IActID)
        For IScriptColumn = 0 To 12
            Call WizHook.GetScriptString(IHmacro, IScriptColumn, sValue(IScriptColumn))
            Debug.Print "ScriptColumn: " & IScriptColumn & vbTab & "Value: " & sValue(IScriptColumn)
        Next IScriptColumn
        IMacroRow = IMacroRow + 1
    Else
        bEndOfMacro = True
    End If
Loop

End Function

```

ReportLeaksToFile(fRptToFile As Boolean, bstrFileOut As String)	2000
--	-------------

Putting this off until later...

SaveObject(bstrName As String, objtyp As Integer)	2000
--	-------------

Putting this off until later...

SaveScriptString(HScr As Long, ScriptColumn As Long, Value As String) As Boolean	2000
---	-------------

Haven't tested this.
Probably saves a value to the macro. See the **OpenScript()** example code and try it there.
If you test it, please send me the details so I can update the documentation.

SetDpBlockKeyInput(fBlockKeys As Boolean)	2000
--	-------------

Putting this off until later...

SetVbaPassword(bstrDbName As String, bstrConnect As String, bstrPasswd As String) As Boolean**2000***Special thanks to Estuardo from accessvba.com for helping with this method!!*

Can set a .mdb VBA password like so, provided it does not already have a VBA password:

```
Public Sub VBAPWSet(sDBName As String, sPW As String)
  If ( WizHook.SetVbaPassword(sDBName, vbNullString, sPW)) Then
    MsgBox "Password set"
  Else
    MsgBox "Failed"
  End If
End Sub
```

This will raise an error if the password can not be set.

If you try this on the current database, it appears to work, but no password is set.

If there is already a VBA password, attempting to set it to a different value will raise an error, but setting it to the same value will not.

(Possible use: VBA password brute-force/dictionary attack)

If you try the example procedure on an ADP, Access will crash.

SetWizGlob(lWhich As Long, vValue)**2002****SortStringArray(Array() As String)****2000***The WizHook Key is not required for this procedure.*

Given an array of strings in Array(), this performs a case-insensitive sort. This is the WizHook function that most people know about.

```
Public Function WH_Test()
  Dim a(3) As String
  Dim i As Integer

  a(0) = "zulu"
  a(1) = "alpha"
  a(2) = "gamma"
  a(3) = "delta"

  WizHook.SortStringArray a()

  For i = 0 To 3
    Debug.Print a(i)
  Next
End Function
```

This will crash Access if you pass it an array with uninitialized elements.**SplitPath(Path As String, Drive As String, Dir As String, File As String, Ext As String)****2000**

Splits Path (a full path to a file) into the Drive, Dir, File, and Ext strings.

It returns an empty string if the corresponding information is not in Path.

TableFieldHasUniqueIndex(Table As String, Columns As String) As Boolean

2000

Return Value:

True - column specified in Columns has a unique index.

False – column specified does not have a unique index (or is not found)

The argument is "Columns", but I can't get it to work with multiple columns, either two columns which both have unique indexes or two columns that make up one unique index.

TranslateExpression(In As String, Out As String, ParseFlags As Long, TranslateFlags As Long) As Boolean

2000

Haven't worked it out yet...

Certain values cause Access to lock up.

TwipsFromFont(FontName As String, Size As Long, Weight As Long, Italic As Boolean, Underline As Boolean, Cch As Long, Caption As String, MaxWidthCch As Long, dx As Long, dy As Long) As Boolean

2000

Supplies the length and width (dx and dy) in twips of Caption or number of character (Cch, using average character width).

If Cch and Caption are both supplied to the function, Cch has priority and Caption is ignored.

If using Cch, then MaxWidthCch to specify how many of that characters should be considered the maximum character width.

In Access 97 this was an semi-undocumented exportable API (it was visible in the unsecured wizards, and also in the *Access 97 Developer's Handbook*):

Declare Function api_TwipsFromFont Lib "msaccess.exe" Alias "#67"

The *Access 2000 Developer's Handbook* has a re-written documented VBA alternative (adh_accTwipsFromFont in Chapter 17). This is a good reference for details about what this function expects (it's a good reference for everything Access related actually).

Public Function WH_TwipsFromFont()

```
Dim sFontName As String
Dim ISize As Long
Dim IWeight As Long
Dim bItalic As Boolean
Dim bUnderline As Boolean
Dim ICch As Long
Dim sCaption As String
Dim IMaxWidthCch As Long
Dim Idx As Long
Dim Idy As Long
```

```
sFontName = "Tahoma"
ISize = 10
IWeight = 400
bItalic = False
bUnderline = False
sCaption = "Hello world!"
```

```
Debug.Print WizHook.TwipsFromFont(sFontName, ISize, IWeight, bItalic, _
```

bUnderline, ICch, sCaption, IMaxWidthCch, ldx, ldy)

```
Debug.Print "Font Name:" & sFontName
Debug.Print "Size: " & ISize
Debug.Print "Weight: " & IWeight
Debug.Print "Italic: " & bItalic
Debug.Print "Underline: " & bUnderline
Debug.Print "Cch: " & ICch
Debug.Print "Caption: " & sCaption
Debug.Print "Max Width Cch: " & IMaxWidthCch
Debug.Print "dx: " & ldx
Debug.Print "dy: " & ldy
```

End Function

WizCopyCmdbars(bstrADPName As String)

2002

WizHelp(HelpFile As String, wCmd As Long, ContextID As Long) As Boolean

2000

Not tested ... obviously something to do with help files.

WizMsgBox(bstrText As String, bstrCaption As String, wStyle As Long, idHelpID As Long, bstrHelpFileName As String) As Long

2002

Opens a MFC-style message box. See the sample code for more info. I was not able to open up a help file, but I didn't try very hard either.

Option Compare Database
Option Explicit

```
Const MB_ABORTRETRYIGNORE = &H2
Const MB_ICONQUESTION = &H20
```

```
Public Sub WH_Test()
```

```
Dim strText As String
Dim strCaption As String
Dim wStyle As Long
Dim idHelpID As Long
Dim sHelpFile As String
```

```
strText = "This is the text in the window." & vbCrLf & "Here is more text."
strCaption = "WizHook MsgBox Text"
wStyle = MB_ABORTRETRYIGNORE + MB_ICONQUESTION
```

' If the following is less than 1, the "Help" button is not displayed

```
idHelpID = 1  
sHelpFile = "somehelpfile.chm"
```

```
    Debug.Print WizHook.WizMsgBox(strText, strCaption, wStyle, idHelpID, sHelpFile)  
End Sub
```