

Recursive Functions

Function that calls itself

Crystal Long (strive4peace)

MsAccessGurus.com

YouTube/LearnAccessByCrystal

Index

#	Title		
1	Recursive Functions	14	— ListFiles and FillDir Code
2	Index	15	Call ListFiles
3	— Allen Browne's ListFiles	16	Allen's ListFiles VBA
4	Form to Run Allen's ListFiles code	17	Allen's FillDir VBA (recursive) add files
5	Allen's ListFiles Result	18	Allen's FillDir VBA (recursive) subfolders
6	Too many results for listbox	19	— s4p Save Relationship Paths
7	query to show number of files for each batch	20	Import RelPath table and module
8	Do whatever you want with the results	21	RelPath Results
9	List objects, loop files	22	— DEMO
10	— Concepts	23	— Resources
11	Help - Dir Function	24	Links
12	Help - Collection	25	Next Access Lunchtime Meeting
13	Helper Function: Allen's TrailingSlash		

— *Allen Browne's* ListFiles

Recursive part is in the FillDir function called by ListFiles that adds on to a folder path down from another folder to make a complete path to a file

For instance:

there may be files in the specified path AND files in folders below it.

W:\(demo\Access\Contacts\Contacts_BE_SampleData.accdb

Form to Run *Allen's* ListFiles code

AllenFilesForm_Listbox

Run Allen Browne's recursive code to List Files

<http://allenbrowne.com/ser-59.html> Save to Table

Include Subfolders FileSpec:

File Path: **W:\(demo**

form by crystal, strive4peace, MsAccessGurus.com ... [Click to view web page for download with this form](#)

1. Type or Paste File Path
2. Decide Include Subfolders
3. Specify FileSpec if desired
4. Decide Save to Table (*s4p_FileList*)
5. Click Run!

Allen's ListFiles Result

AllenFilesForm_Listbox

Run Allen Browne's recursive code to List Files

<http://allenbrowne.com/ser-59.html> Save to Table

101 files listed 101 found BatchID: 1 Include Subfolders FileSpec:

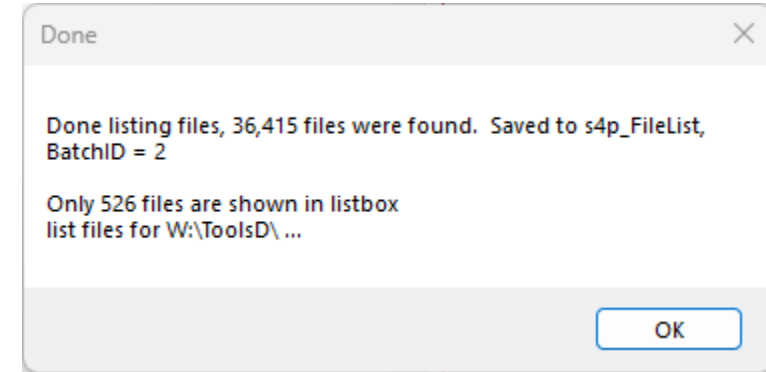
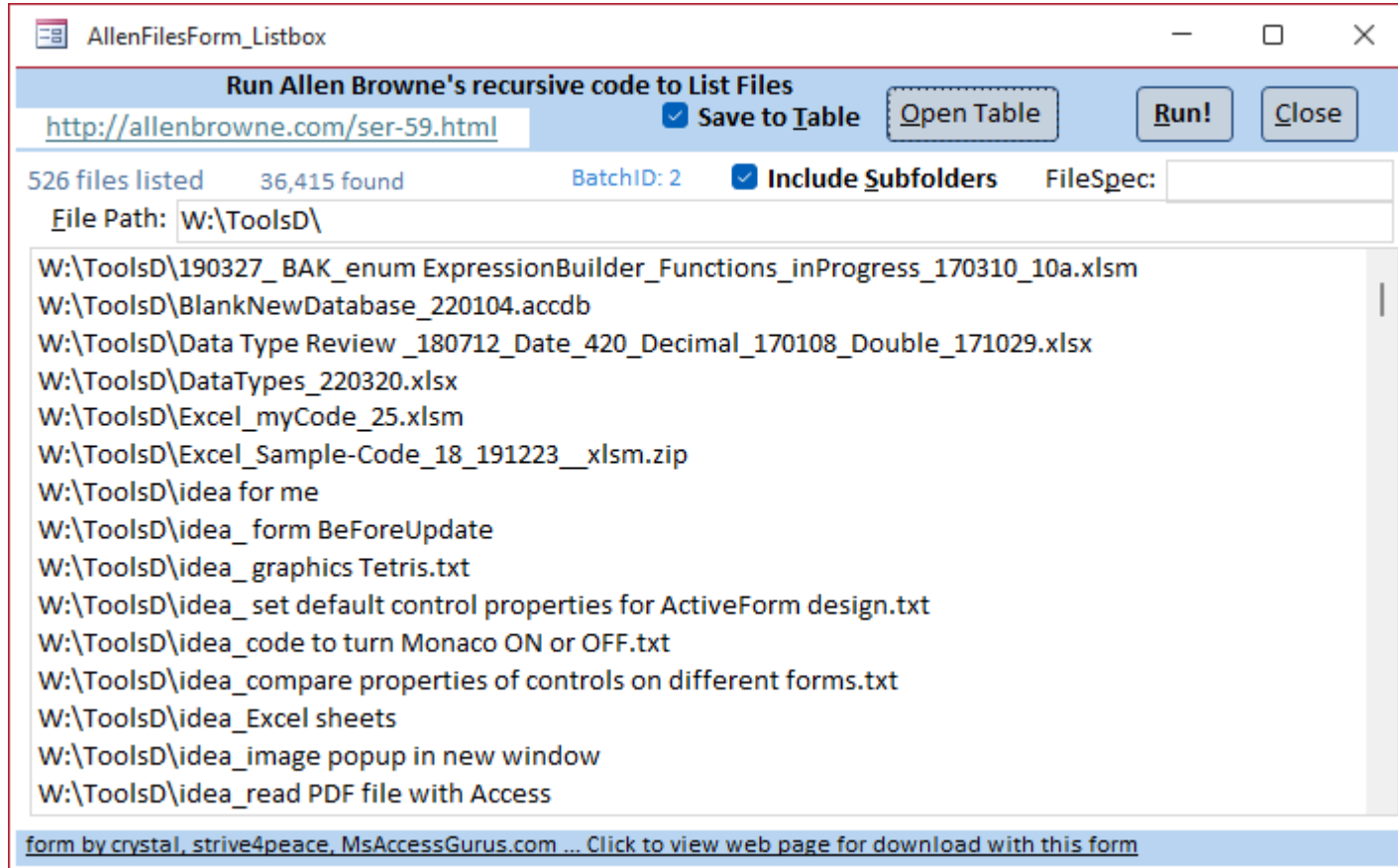
File Path: W:\(demo

- W:\(demo\links_AccessWord.txt
- W:\(demo\s_ MyPathD Demo.Ink
- W:\(demo\s_ tools AccessWord aListFields.Ink
- W:\(demo\s_ toolsD aWord ListBookmarks.Ink
- W:\(demo\Access\Contact_LastID_251123_s4p.accdb
- W:\(demo\Access\copyCOLORS_230207_0529.accdb
- W:\(demo\Access\COPYCommandButtons_ChooseColor_s4p_230606_.accdb
- W:\(demo\Access\COPY_s4p_DataDictionary_250706.acdda
- W:\(demo\Access\dAnalyzer_250705_s4p_Peter_Analyzer.accdb
- W:\(demo\Access\demo_CalendarMaker_s4p_230214_3264.accdb
- W:\(demo\Access\demo_Contact_251123_1p_LastIDs4p -OpenR.accdb
- W:\(demo\Access\demo_Expressions_s4p_250930.accdb
- W:\(demo\Access\demo_TabControl_s4p_240325_8a.accdb
- W:\(demo\Access\demo__aWord ListBookmarks_250412_open13.accdb
- W:\(demo\Access\demo__aWord_ListFields_s4p_250405_OLD.accdb
- W:\(demo\Access\demo__aWord_ListFields_s4p_250413_.accdb

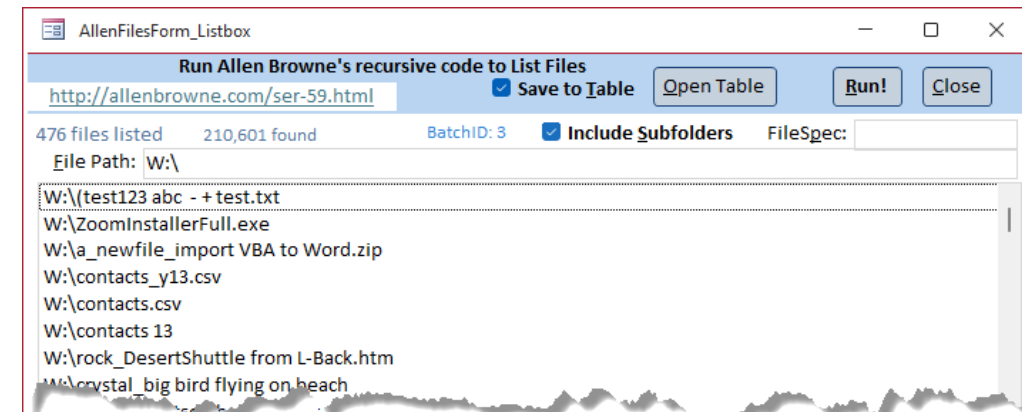
form by crystal, strive4peace, MsAccessGurus.com ... [Click to view web page for download with this form](#)

Show file names and paths.
If the "file" is a folder then,
optionally if Include Subfolders
is checked,
recursively loop each subfolder
and add more files to the
collection.

Too many results for listbox



BatchID=3: ran it again on the whole drive and got even more files



Look at StatusBar in lower left to see it is running

RUNNING get list of files for specified path ... W:\ToolsD\CHEMISTRY\Notes\

query to show number of files for each batch

BatchID	NbrFiles
1	101
2	36,415
3	210,601

Record: 1 of 3

FileListID	BatchID	FileFull	dtmAdd
1	1	W:\(demo\links_AccessWord.txt	2/23/2026 8:51:05 AM
2	1	W:\(demo\s_MyPathD_Demo.Ink	2/23/2026 8:51:05 AM
3		AccessWord_aListFields.Ink	2/23/2026 8:51:05 AM
4		Word_ListBookmarks.Ink	2/23/2026 8:51:05 AM
5		act_LastID_251123_s4p.accdb	2/23/2026 8:51:05 AM
6		COLORS_230207_0529.accdb	2/23/2026 8:51:05 AM
7		\CommandButtons_ChooseColor_s4	2/23/2026 8:51:05 AM
8		\s4p_DataDictionary_250706.accda	2/23/2026 8:51:05 AM
9		alyzer_250705_s4p_Peter_Analyzer.ac	2/23/2026 8:51:05 AM
10		_CalendarMaker_s4p_230214_3264.	2/23/2026 8:51:05 AM
11		_Contact_251123_1p_LastIDs4p-Op	2/23/2026 8:51:05 AM
12		_Contact_251123_1p_LastIDs4p.acc	2/23/2026 8:51:05 AM
13		_Expressions_s4p_250930.accdb	2/23/2026 8:51:05 AM
14		_TabControl_s4p_240325_8a.accdb	2/23/2026 8:51:05 AM
15		_aWord_ListBookmarks_250412_op	2/23/2026 8:51:05 AM
16		_aWord_ListFields_s4p_250405_OL	2/23/2026 8:51:05 AM
17		_aWord_ListFields_s4p_250413_.ac	2/23/2026 8:51:05 AM
18	1	W:\(demo\Access\demo_aWord_ShowHide_250326_s4p_a	2/23/2026 8:51:05 AM
19	1	W:\(demo\Access\demo_MomDadKid_AZ.accdb	2/23/2026 8:51:05 AM
20	1	W:\(demo\Access\demo_NavigationControl_2004.accdb	2/23/2026 8:51:05 AM
21	1	W:\(demo\Access\demo_WordDocPropActive_s4p_231207.	2/23/2026 8:51:05 AM

Record: 1 of 247117

Sort menu options:
Sort Smallest to Largest
Sort Largest to Smallest
Clear filter from BatchID
Number Filters:
 (Select All)
 (Blanks)
 1
 2
 3

Do whatever you want with the results

add more fields to the table (s4p_FileList) to parse and get information like:

- Path
- File name
- Extension
- Size
- Date created
- Date modified

and then depending on file type:

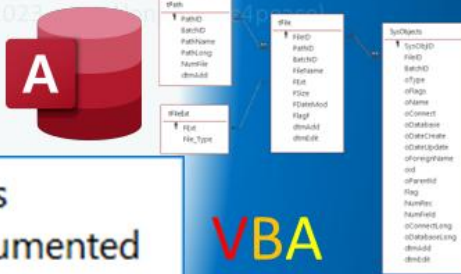
- Dimensions
- Length
- whatever you write code to figure out

List objects, loop files



List Objects, Loop Files

Find lost objects in Access databases
Documentation



Done documenting Access Objects
280,026 objects in 2,048 files documented
counted records in 39,975 tables

f_MENU_ListObjects_LoopFiles_s4p

List Objects, Loop Access Files

Folder: E:\

Recursive document databases in sub-folders too

Count Records

Reports

Batch: 1 2023-10-12, Thu 21:29

Object Type: Pattern:

include Microsoft system objects (MSys + more)

Loop files, get object info

count records in tables

48.1%

ReportCats

E:\Tools20_AnalyzerTool929\DATA\Source\
nvert_140509_COPY.accdb

ObjTyp	oName	Objects	NumRec	NumField	t
Table	t_ELEMENT		118	25	db
Table	t_ElementProperty		77	26	db
Table	t_Group		24	20	db
Table	t_NutrientCategory		7	7	db
Table	t_Property		2	15	db

Tables	Linked Tbls	ODBC Tbls	Queries	Forms	Reports	Macros	Modules	#Records
BatchID 1	run:	2023-10-12, Thu 21:29						
1	C:\MyPath\Demo\Access							Bytes: 8,482,816
	MyContacts_s4p_220407.accdb							2022-04-07, Thu 11:37
			20	17	12	1	15	43,782

Query	Form	Report	Macro	Module	#Records
qUp_Element_Neutrons	f_ELEMENTS	r_Circle_ELEMENT_ATC			
	f_GetColorValues	r_Circle_ELEMENT_ATC			
		r_Line_Border_Element			

Did you lose track of something important in your many Access databases?

I wrote this to find a module and found it!

https://msaccessgurus.com/tool/ListObjects_LoopFiles.htm

— Concepts

to better understand Allen's code

Help - Dir Function

Learn / VBA /

<https://learn.microsoft.com/en-us/office/vba/language/reference/user-interface-help/dir-function>

Dir function

Returns a **String** representing the name of a file, directory, or folder that matches a specified pattern or file attribute, or the volume label of a drive.

Syntax

`Dir [(pathname, [attributes])]`

The **Dir** function syntax has these parts:

Part	Description
<i>pathname</i>	Optional. String expression that specifies a file name; may include directory or folder, and drive. A zero-length string ("") is returned if <i>pathname</i> is not found.
<i>attributes</i>	Optional. Constant or numeric expression, whose sum specifies file attributes. If omitted, returns files that match <i>pathname</i> but have no attributes.

Settings

The *attributes* argument settings are:

Constant	Value	Description
<code>vbNormal</code>	0	(Default) Specifies files with no attributes.
<code>vbReadOnly</code>	1	Specifies read-only files in addition to files with no attributes.
<code>vbHidden</code>	2	Specifies hidden files in addition to files with no attributes.
<code>vbSystem</code>	4	Specifies system files in addition to files with no attributes. Not available on the Macintosh.
<code>vbVolume</code>	8	Specifies volume label; if any other attribute is specified, <code>vbVolume</code> is ignored. Not available on the Macintosh.
<code>vbDirectory</code>	16	Specifies directories or folders in addition to files with no attributes.
<code>vbAlias</code>	64	Specified file name is an alias. Available only on the Macintosh.

ADDING FILES to a collection

' first time, attributes not specified so Normal files
`strTemp = Dir(strFolder & strFileSpec)`

' loop

`Do While strTemp <> vbNullString`
 `colDirList.Add strFolder & strTemp`
 `strTemp = Dir`

`Loop`

Object Browser says that `vbNullString = ""`
although it can mean more

Learn / VBA /

Collection object

A Collection object is an ordered set of items that can be referred to as a unit.

Remarks

The Collection object provides a convenient way to refer to a related group of items as a single object. The items, or members, in a collection need only be related by the fact that they exist in the collection. Members of a collection don't have to share the same data type.

A collection can be created the same way other objects are created. For example:

```
VB Copy  
Dim X As New Collection
```

After a collection is created, members can be added by using the Add method and removed by using the Remove method. Specific members can be returned from the collection by using the Item method, while the entire collection can be iterated by using the For Each...Next statement.

Example

This example creates a Collection object (MyClasses), and then creates a dialog box in which users can add objects to the collection.

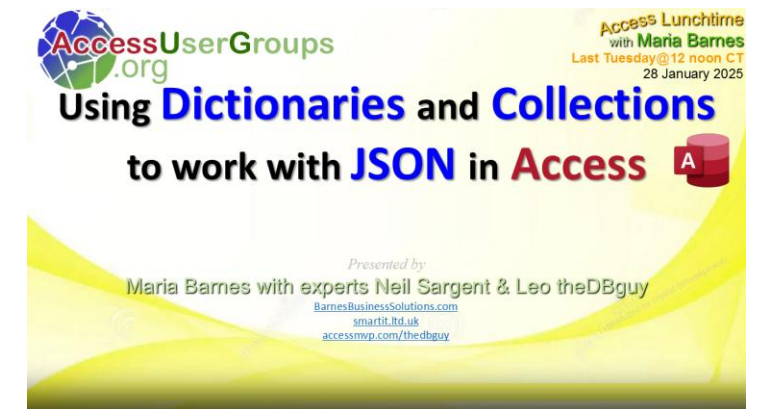
To see how this works, choose the Class Module command from the Insert menu and declare a public variable called InstanceName at the module level of Class1 (type Public InstanceName) to hold the names of each instance. Leave the default name as Class1. Copy and paste the following code into the General section of another module, and then start it with the statement ClassNameer in another procedure.

(This example only works with host applications that support classes.)

```
VB Copy  
Sub ClassNameer()  
    Dim MyClasses As New Collection ' Create a Collection object.  
    Dim Num ' Counter for individualizing keys.  
    Dim Msg As String ' Variable to hold prompt string.  
    Dim TheName, MyObject, NameList ' Variants to hold information.  
End Sub
```

VBA Collection Object

<https://learn.microsoft.com/en-us/office/vba/language/reference/user-interface-help/collection-object>



Access Lunchtime
with Maria Barnes
Last Tuesday @ 12 noon CT
28 January 2025

AccessUserGroups.org

Using Dictionaries and Collections

to work with JSON in Access

Presented by
Maria Barnes with experts Neil Sargent & Leo theDBguy
BarnesBusinessSolutions.com
smarthub.co.uk
accessmvp.com/theDBGuy

<https://youtu.be/oKRtghY9cJ0>

Helper Function: *Allen's* TrailingSlash

Public Function TrailingSlash(varIn As Variant) **As String**

If Len(varIn) > 0 **Then**

If Right(varIn, 1) = "\" **Then**

TrailingSlash = varIn

RETURN value passed if it already ends with "\"

Else

TrailingSlash = varIn & "\"

RETURN value with "\" on end if not there

End If

End If

End Function

& is the type declaration
character for Long Integer

— ListFiles and FillDir Code

by Allen Browne, modified by Crystal

Call ListFiles

Run Allen Browne's recursive code to List Files

<http://allenbrowne.com/ser-59.html>

Save to Table

Open Table

Run!

Close

the Run! button calls Allen's code to build a collection of file names and paths.

First parameter is path to look in for files and the only required argument. Second is a file specification with wildcards. Third is whether or not to include subfolders. Last is listbox control for results instead of writing to Immediate window.

' recursively list files for path

**Call ListFiles(sPath _
 , .FileSpec _
 , .Checkbox_IncludeSubfolders _
 , .lstFileList)**

' List files recursively, by Allen Browne
' <http://allenbrowne.com/ser-59.html>

Public Function ListFiles(strPath As String, _
 Optional strFileSpec As String, _
 Optional bIncludeSubfolders As Boolean, _
 Optional lst As ListBox)

Allen's ListFiles VBA

'from <http://allenbrowne.com/ser-59.html>

```
Public Function ListFiles(strPath As String, _  
    Optional strFileSpec As String, _  
    Optional bIncludeSubfolders As Boolean, _  
    Optional lst As ListBox)
```

```
On Error GoTo Err_Handler
```

'Purpose: List the files in the path.

'Arguments: strPath = the path to search.

' strFileSpec = "*.*)" unless you specify differently.

' bIncludeSubfolders: If True, results from subdirectories as well.

' lst: if you pass in a list box, items are added to it.

' If not, files are listed to immediate window.

' The list box must have RowSourceType property set to ValueList.

'Method: FilDir() adds items to collection, recursively for subfolders.

```
Dim colDirList As New Collection
```

```
Dim varItem As Variant
```

' ~ inialize variable value

```
gnCount = 0 ' ~~~ added by crystal
```

'call function that can be recursive

```
Call FilDir(colDirList, strPath, strFileSpec, bIncludeSubfolders)
```

' ~ how many files are in the collection

```
gnCount = colDirList.Count ' ~~~ added by crystal
```

'Add the files to a list box if one was passed in.

' Otherwise list to the Immediate Window.

```
If lst Is Nothing Then  
    For Each varItem In colDirList  
        Debug.Print varItem  
    Next  
Else  
    For Each varItem In colDirList  
        lst.AddItem varItem  
    Next  
End If
```

'----- 's4p ~~~ added by crystal

' write collection to a table

If gbSAVEtoTABLE <> False Then

```
    Call SaveCollectionToTable_s4p(colDirList)
```

```
End If
```

Exit_Handler:

```
Exit Function
```

Err_Handler:

```
MsgBox "Error " & Err.Number & ": " & Err.Description
```

```
Resume Exit_Handler
```

```
Resume
```

```
End Function
```

Allen's FillDir VBA (recursive) *add files*

in calling procedure: Dim colDirList As New Collection 'nothing is in collection until **FillDir** adds file names preceded by path

'from <http://allenbrowne.com/ser-59.html>

Private Function FillDir(colDirList As Collection, _

ByVal strFolder As String, _

strFileSpec As String, _

bIncludeSubfolders As Boolean)

'Build up a list of files, and then add add to this list

', any additional folders

Dim strTemp **As String**

Dim colFolders **As New** Collection

Dim vFolderName **As Variant**

'Add the files to the folder.

strFolder = TrailingSlash(strFolder) *strFolder is the path*

strTemp = Dir(strFolder & strFileSpec) *get first file name in folder*

Do While strTemp <> vbNullString *do while there is a file name in folder* **Next** vFolderName

colDirList.Add strFolder & strTemp *Add item to collection*

strTemp = Dir *get next file in folder*

Loop

If bIncludeSubfolders **Then**

'Build collection of additional subfolders.

strTemp = Dir(strFolder,vbDirectory)

Do While strTemp <> vbNullString

If (strTemp <> ".") And (strTemp <> "..") **Then**

If (GetAttr(strFolder & strTemp) And vbDirectory) <> 0 & **Then**

colFolders.Add strTemp

End If

End If

strTemp = Dir

Loop

'Call function recursively for each subfolder.

For Each vFolderName **In** colFolders

Call FillDir(colDirList, strFolder & TrailingSlash(vFolderName) _

, strFileSpec, **True**)

Next vFolderName

End If

End Function

Allen's FillDir VBA (recursive) *subfolders*

'from <http://allenbrowne.com/ser-59.html>

```
Private Function FillDir( colDirList As Collection, _
    ByVal strFolder As String, _
    strFileSpec As String, _
    bIncludeSubfolders As Boolean)
    'Build up a list of files, and then add add to this list
    ' , any additional folders
    Dim strTemp As String
    Dim colFolders As New Collection
    Dim vFolderName As Variant

    'Add the files to the folder.
    strFolder = TrailingSlash(strFolder)
    strTemp = Dir(strFolder & strFileSpec)

    Do While strTemp <> vbNullString
        colDirList.Add strFolder & strTemp
        strTemp = Dir
    Loop
```

in calling procedure: Dim colDirList As New Collection 'nothing is there until this procedure fills the collection with files

If bIncludeSubfolders Then

```
    'Build collection of additional subfolders.
    strTemp = Dir( strFolder, vbDirectory) Normal Files + Folders
    Do While strTemp <> vbNullString
        If (strTemp <> ".") And (strTemp <> "..") Then
            If (GetAttr(strFolder & strTemp) And vbDirectory) <> 0& Then
                colFolders.Add strTemp strTemp is now the folder path
            End If
        End If
        strTemp = Dir
    Loop
    'Call function recursively for each subfolder.
    For Each vFolderName In colFolders
        Call FillDir( colDirList, strFolder & TrailingSlash( vFolderName ) _
            , strFileSpec, True)
    Next vFolderName
End If
End Function
```

— s4p Save Relationship Paths

RelPath	BaseTable	RelTable	Level	RelField	NoteRP	ParentTable	ParentField
c_Category—c_Contact—c_Contact	c_Category	c_Contact	2	CID_	self-join	c_Contact	CID
c_Category—c_Contact—c_eAddress	c_Category	c_eAddress	2	CID		c_Contact	CID
c_Category—c_Contact—c_List	c_Category	c_List	2	CID_Li		c_Contact	CID
c_Category—c_Contact—c_List—c_List	c_Category	c_List	3	ListID_	self-join	c_List	ListID
c_Category—c_Contact—c_List—c_Member	c_Category	c_Member	3	ListID		c_List	ListID
c_Category—c_Contact—c_Member	c_Category	c_Member	2	CID		c_Contact	CID
c_Category—c_Contact—c_Phone	c_Category	c_Phone	2	CID		c_Contact	CID
c_Category—c_Contact—c_URL	c_Category	c_URL	2	CID		c_Contact	CID

Import RelPath table and module

to see RelPath in action, you need to put it into a database.

import:

- s4p_RelPath *table*
- s4p_mod_RelPath_Recurse *module*

into your database.

Compile, Save, and run **DoRelPath_All_s4p** OR **DoRelPath_OneTable_s4p**

RelPath Results

RelPath	BaseTable	RelTable	Level	RelField	NoteRP	ParentTable	ParentField
c_AdrType—c_Address	c_AdrType	c_Address	1	TypIDadr		c_AdrType	TypIDadr
c_Category—c_Contact	c_Category	c_Contact	1	CatID		c_Category	CatID
c_Category—c_Contact—c_Address	c_Category	c_Address	2	CID		c_Contact	CID
c_Category—c_Contact—c_Contact	c_Category	c_Contact	2	CID_	self-join	c_Contact	CID
c_Category—c_Contact—c_eAddress	c_Category	c_eAddress	2	CID		c_Contact	CID
c_Category—c_Contact—c_List	c_Category	c_List	2	CID_Li		c_Contact	CID
c_Category—c_Contact—c_List—c_List	c_Category	c_List	3	ListID_	self-join	c_List	ListID
c_Category—c_Contact—c_List—c_Member	c_Category	c_Member	3	ListID		c_List	ListID
c_Category—c_Contact—c_Member	c_Category	c_Member	2	CID		c_Contact	CID
c_Category—c_Contact—c_Phone	c_Category	c_Phone	2	CID		c_Contact	CID
c_Category—c_Contact—c_URL	c_Category	c_URL	2	CID		c_Contact	CID
c_Contact—c_Address	c_Contact	c_Address	1	CID		c_Contact	CID

— DEMO

demonstrate in Access

— Resources

Links

Download this presentation + ACCDB databases for ListFile and RelPath

https://msaccessgurus.com/presentation/Recursive_Functions.htm

List files recursively by Allen Browne

<http://allenbrowne.com/ser-59.html>

Dir function

<https://learn.microsoft.com/en-us/office/vba/language/reference/user-interface-help/dir-function>

Collection object

<https://learn.microsoft.com/en-us/office/vba/language/reference/user-interface-help/collection-object>

List Objects, Loop files – names of Access objects

https://msaccessgurus.com/tool/ListObjects_LoopFiles.htm

AL: Using Dictionaries and Collections to work with JSON in Access by Maria Barnes, Neil Sargent, Leo theDBGuy (1:03:47)

<https://youtu.be/c7PwDuXPLeA>

Next Meeting: Interfacing with the Outlook calendar by Maria Barnes

<https://accessusergroups.org/calendar/lunchtime-2-2026-03-31/>

Access Lunchtime Playlist on YouTube:

https://www.youtube.com/playlist?list=PLwlULo1M2EWvapx3zAhd3jhX_ITigxM6M

Next Access Lunchtime Meeting



Access Lunchtime
with Maria Barnes
Last Tuesday @ 12 noon CT

Next Access Lunchtime meeting:

Interfacing with the Outlook calendar

- Classic Outlook
- New Outlook

with Maria Barnes

- user calendar
- shared calendar

Tuesday 31 March 2026, 12 noon Central

join us!

<https://accessusergroups.org/calendar/lunchtime-2-2026-03-31/>