

 Search**Main Page**[Home Page](#)**Microsoft Office**[MS Access](#)[MS Excel](#)[MS PowerPoint](#)[MS Word](#)**Operating Systems**[Windows 10](#)[Windows 7](#)[Windows Vista](#)[Windows XP](#)**Miscellaneous**[Adobe Photoshop](#)[A+ Certification](#)[iPad](#)[iPhone](#)[Networking](#)[PHP](#)[MS-Access](#) / [Getting Started](#)

Working with Attachment Fields

Are You Ready For Love?

Russian & Ukrainian Girls! Join
Now The Online Dating Site For
Singles

[online-internet-dating.com](#)



As mentioned earlier, Access 2007 includes a new data type - Attachment - that you can use with ACCDB files in Access. This type can store zero or more files that are associated with an individual record. Remember the students and classes example? Say that you want to store the class syllabus and homework assignments with the class. The Attachment data type enables you to save the file as part of the database without the bloat of an OLE Object.

Attachment fields are a special type of multi-valued field in which multiple fields are included in the nested recordset. The fields defined by the Attachment data type are described in the following table:

Field Name	Description
FileData	The file itself is stored in this field.
FileFlags	Reserved for future use.
FileName	The name of the file in the attachment field.
FileTimeStamp	Reserved for future use.
FileType	The file extension of the file in the attachment field.
FileURL	The URL for the file for a linked SharePoint list. Will be Null for local Access tables.

Navigating Attachments

Because attachment fields are a type of multi-valued field, you can navigate them by enumerating through the nested recordset for the field. The following code

shows how to print a list of attachments that are included with each record in a table.

```

Sub ListAttachments()
    Dim dbs As DAO.Database
    Dim rst As DAO.Recordset2
    Dim rsA As DAO.Recordset2
    Dim fld As DAO.Field2

    'Get the database, recordset, and attachment field
    Set dbs = CurrentDb
    Set rst = dbs.OpenRecordset("tblAttachments")
    Set fld = rst("Attachments")

    'Navigate through the table
    Do While Not rst.EOF

        'Print the first and last name
        Debug.Print rst("FirstName") & " " & rst("LastName")

        'Get the recordset for the Attachments field
        Set rsA = fld.Value

        'Print all attachments in the field
        Do While Not rsA.EOF

            Debug.Print , rsA("FileType"), rsA("FileName")

            'Next attachment
            rsA.MoveNext
        Loop

        'Next record
        rst.MoveNext
    Loop

    rst.Close
    dbs.Close
    Set fld = Nothing
    Set rst = Nothing
    Set dbs = Nothing
End Sub

```

Adding, Saving, and Deleting Attachments

To load binary data in an Access database in the past, you could either use the OLE Object data type and automate a form by using the Bound OLE Object control or you could use the AppendChunk method of the Field object. Attachment fields make this much more elegant and save space because they are compressed in the database.

Adding Attachments

Using the Field2 object, you can insert or save attachment fields. The Field2 object makes it easy to insert an attachment into a field using a new method called LoadFromFile.

The following code demonstrates inserting a file into an attachment field. The strPattern argument in the function enables you to add all files in the directory specified by strPath that match a given pattern. This might be useful for loading all .bmp files in a folder, but not the .gif files.

```
Public Function LoadAttachments(strPath As String, Optional strPattern As i
String = "*.*") As Long
    Dim dbs As DAO.Database
    Dim rst As DAO.Recordset2
    Dim rsA As DAO.Recordset2
    Dim fld As DAO.Field2
    Dim strFile As String

    'Get the database, recordset, and attachment field
    Set dbs = CurrentDb
    Set rst = dbs.OpenRecordset("tblAttachments")
    Set fld = rst("Attachments")

    'Navigate through the table
    Do While Not rst.EOF

        'Get the recordset for the Attachments field
        Set rsA = fld.Value

        'Load all attachments in the specified directory
        strFile = Dir(strPath & "*.*")

        rst.Edit

        Do While Len(strFile) > 0

            'Add a new attachment that matches the pattern.
            'Pass "" to match all files.
            If strFile Like strPattern Then
                rsA.AddNew
```

```

        rsA("FileData").LoadFromFile strPath & "\" & strFile
        rsA.Update

        'Increment the number of files added
        LoadAttachments = LoadAttachments + 1
    End If
    strFile = Dir
Loop
rsA.Close

rst.Update
'Next record
rst.MoveNext
Loop

rst.Close
dbs.Close

Set fld = Nothing
Set rsA = Nothing
Set rst = Nothing
Set dbs = Nothing
End Function

```

Saving Attachments

To save an OLE Object field value to the computer required writing code for the Bound OLE Object control on a form. Using an Attachment field, you can now save your attachments to the computer without the need for a form. The Field2 object includes a new method named SaveToFile that makes this easier. The following code demonstrates saving an attachment to a specified location.

```

Public Function SaveAttachments(strPath As String, Optional strPattern As
String = "*.*) As Long
    Dim dbs As DAO.Database
    Dim rst As DAO.Recordset2
    Dim rsA As DAO.Recordset2
    Dim fld As DAO.Field2
    Dim strFullPath As String

    'Get the database, recordset, and attachment field
    Set dbs = CurrentDb
    Set rst = dbs.OpenRecordset("tblAttachments")
    Set fld = rst("Attachments")

    'Navigate through the table

```

```

Do While Not rst.EOF

    'Get the recordset for the Attachments field
    Set rsA = fld.Value

    'Save all attachments in the field
    Do While Not rsA.EOF
        If rsA("FileName") Like strPattern Then
            strFullPath = strPath & "\" & rsA("FileName")

            'Make sure the file does not exist and save
            If Dir(strFullPath) = "" Then
                rsA("FileData").SaveToFile strFullPath
            End If

            'Increment the number of files saved
            SaveAttachments = SaveAttachments + 1
        End If

        'Next attachment
        rsA.MoveNext
    Loop
    rsA.Close

    'Next record
    rst.MoveNext
Loop

rst.Close
dbs.Close

Set fld = Nothing
Set rsA = Nothing
Set rst = Nothing
Set dbs = Nothing

```

Deleting Attachments

The following code shows you how to delete an attachment from a table. The `strRemoveFile` argument is the name of the file to remove. Specify the `strFilter` argument to add a filter to the table prior to deleting attachments.

```

Function RemoveAttachment(strRemoveFile As String, Optional strFilter As
String) As Long
    Dim dbs As DAO.Database
    Dim rst As DAO.Recordset2

```

```

Dim rsA As DAO.Recordset

Dim fld As DAO.Field2

'Get the database
Set dbs = CurrentDb

'Open the recordset. If the strFilter is supplied, add it to the WHERE
'clause for the recordset. Otherwise, any files matching strFileName
'will be deleted
If Len(strFilter) > 0 Then
    Set rst = dbs.OpenRecordset("SELECT * FROM tblAttachments WHERE "
& strFilter)
Else
    Set rst = dbs.OpenRecordset("tblAttachments")
End If

'Get the Attachment field
Set fld = rst("Attachments")

'Navigate through the recordset
Do While Not rst.EOF

    'Get the recordset for the Attachments field
    Set rsA = fld.Value

    'Walk the attachments and look for the file name to remove
    Do While Not rsA.EOF
        If rsA("FileName") Like strRemoveFile Then
            rsA.Delete

            'Increment the number of files removed
            RemoveAttachment = RemoveAttachment + 1
        End If
        rsA.MoveNext
    Loop

    'Cleanup the Attachments recordset
    rsA.Close
    Set rsA = Nothing

    'Next record
    rst.MoveNext
Loop

rst.Close

```

```

dbs.Close

Set fld = Nothing

Set rst = Nothing

Set dbs = Nothing

End Function

```

[\[Previous\]](#) [\[Contents\]](#) [\[Next\]](#)

Write better VBA code faster!

✓ Knowledge and productivity		Access
✓ Code Builders		Excel
✓ Code Explorer		Word
✓ Code Library		Outlook
		AutoCAD
		and more ...


Code VBA


Download

In this tutorial:

1. [Using DAO to Access Data](#)
2. [Data Access Objects](#)
3. [New Features in DAO](#)
4. [Referring to DAO Objects](#)
5. [The DBEngine Object](#)
6. [Using Transactions](#)
7. [The Errors Collection](#)
8. [The Databases Collection](#)
9. [The CurrentDb\(\) Function](#)
10. [Opening an External Database](#)
11. [Closing and Destroying Database Object References](#)
12. [DAO Property Types](#)
13. [Setting and Retrieving Built-In Object Properties](#)
14. [Setting and Retrieving SummaryInfo Properties](#)
15. [Creating Schema Objects with DAO](#)
16. [Creating Indexes](#)
17. [Creating Relations](#)
18. [Creating Multi-Value Lookup Fields](#)
19. [Database Encryption with DAO](#)
20. [Setting Encryption Options](#)
21. [Managing Access \(JET\) Security with DAO](#)
22. [Creating Security Objects](#)
23. [Creating and Deleting Groups](#)
24. [Managing Passwords](#)
25. [Data Access with DAO](#)

26. [Modifying a QueryDef](#)
27. [Filtering and Ordering Recordsets](#)
28. [Navigating Recordsets](#)
29. [BOF, EOF](#)
30. [Navigating Recordsets with Multi-Value Lookup Fields](#)
31. [Bookmarks and Recordset Clones](#)
32. [Finding Records](#)
33. [Working with Recordsets](#)
34. [Using Arrays with Recordsets](#)
35. [Working with Attachment Fields](#)
36. [Append Only Fields](#)

© 2013 [SourceDaddy](#) - [Privacy Policy](#) - [Legal](#) - [Contact](#)