

## Seminar for 12/17/99: Topics on File I/O

### Text Files

Text data files are usually to be imported into Access in order to transform the raw data into recordsets. The data from vendors or suppliers commonly arrives as text, since the text file is the “lingua franca” of data processing. Most applications have the facility of reading in and interpreting text documents.

In other cases, the data may arrive in some other form, such as a spreadsheet or legacy database table format and there are times when it is convenient to export this to text, and then import the text into Access.

### Access Import Wizard:

This program scans your input text file and makes a determination of what the text format represents. For simple files, the Import Wizard does a creditable job. If the text file is a bit more complicated, then the overhead of using it becomes more burdensome than writing a custom routine to read in the file – especially if your task is the periodic loading of data. Such a custom import routine becomes a necessity if the data format is not so straightforward. For example, some numeric formats arrive in “main-frame”-style packed decimal, or records may be separated into fields by special hexadecimal characters.

### Export Wizard:

Again, for simple output requirements, the wizard is fine, but there are times when you may want to output a file with custom specifications, if it is to ultimately be input to a proprietary processing program, such as StarView. More commonly, however, you might want to create HTML table data or XSL files from an Access table. In such a case, you apply your own processing “metadata” (such as HTML tags) to your output text file.

### Text File Types:

The three types of text file you’re likely to run into for raw data are:

- **SEQUENTIAL:** This is the sort of file put out by Notepad, for instance. There are no assumptions made about record structure in this type of file. It is organized around the “line”, which is a string of characters ended by an end-of-line indicator, which is usually a CR(Carriage Return) followed by LF(Line Feed), or the Ascii notation chr(13) chr(10). When Sequential files are used to store records, each record fills up a line and is separated from the next record by CR + LF. The fields within the record are designated as DELIMITED TEXT: The fields are separated from one another with a comma (or other character), text fields are surrounded by double quotes. The length of any given record is not assumed, and all lines may be of different length.
- **RANDOM:** This type of file consists of fixed-length records, usually (but not necessarily) separated from one another with CR, LF. In order to read this file successfully, you must know the length of a record, in characters. No assumptions at all are made about the divisions of this fixed-length unit into fields.
- **BINARY:** “Anything goes”. No breaking up into “lines” is assumed, no record length is assumed. To deal with a file like this, you designate a buffer memory variable of some convenient length, (say 256 characters), and then use character manipulation code to figure out what is in it. Normally, this sort of file is not used to output structured record data. We’ll be concentrating on the other two formats.

## Sequential Files

For the purposes of field recognition, here is a typical Sequential or “Delimited” text file as it appears in Notepad:

```

SequentialFile.txt - Notepad
File Edit Search Help
"CDA","Greater Wenatchee Reg Landfill & Recycling Ctr","Chelan","Asbestos",38,"County",""
"CDA","Greater Wenatchee Reg Landfill & Recycling Ctr","Benton","Asbestos",1,"County",""
"CDA","Greater Wenatchee Reg Landfill & Recycling Ctr","Grant","Asbestos",3,"County",""
"NKO2","Olympic View Landfill","Western WA","Asbestos",2000,"County",""
"EWS2","Sudbury Road Landfill","Benton","Asbestos",3,"County",""
"EWS2","Sudbury Road Landfill","Franklin","Asbestos",.3,"County",""
"EWS2","Sudbury Road Landfill","Spokane","Asbestos",.16,"County",""
"EWS2","Sudbury Road Landfill","Asotin","Asbestos",.31,"County",""
"EWS2","Sudbury Road Landfill","Oregon","Asbestos",.11,"State","Union County"
"EWS2","Sudbury Road Landfill","Oregon","Asbestos",3,"State","Umatilla County"
"EWU","Whitman County Landfill","Lincoln","Asbestos",0,"County","120 lbs"
"EWU","Whitman County Landfill","Idaho","asbestos",13,"State",""
  
```

## Random Files

This is the same data as a fixed-length text file. In this case, with a CR and LF terminating each record:

```

RandomFile.txt - Notepad
File Edit Search Help
CDA      Greater Wenatchee Reg Landfill & Recycling Ctr   Chelan      Asbestos
 38      County
CDA      Greater Wenatchee Reg Landfill & Recycling Ctr   Benton      Asbestos
 1       County
CDA      Greater Wenatchee Reg Landfill & Recycling Ctr   Grant       Asbestos
 3       County
NKO2     Olympic View Landfill                             Western WA  Asbestos
2000    County
EWS2     Sudbury Road Landfill                             Benton      Asbestos
 3       County
EWS2     Sudbury Road Landfill                             Franklin    Asbestos
 0.3     County
EWS2     Sudbury Road Landfill                             Spokane     Asbestos
 0.16    County
EWS2     Sudbury Road Landfill                             Asotin     Asbestos
 0.31    County
EWS2     Sudbury Road Landfill                             Oregon     Asbestos
 0.11    State      Union County
EWS2     Sudbury Road Landfill                             Oregon     Asbestos
 3       State      Umatilla County
EWU      Whitman County Landfill                             Lincoln    Asbestos
 0       County      120 lbs
EWU      Whitman County Landfill                             Idaho     asbestos
 13      State
  
```

## The Sample Application

The Access sample application, “FileIO.mdb” shows examples of how to read in these text files, and how to append their data to an Access tables. Alternatively, it takes an Access Table, runs a query on it, and outputs the query result in the two different formats.

## Methods Used

For both examples, random and sequential, the types of OPEN and CLOSE commands are used for Input and Output.

For Sequential access, we use the READ and WRITE methods, for Random, the GET and PUT methods.  
(This by no means exercises all the appropriate statements or commands for these access methods, but they are the most common.)

## Type Statement

The Random file, since it is of fixed length, and the fields within the record are otherwise not defined, means that the program which reads or writes the file must impose such a structure upon the record. One way of doing it is by defining a user type, which functions as a record layout. It's put in the Declaration section of the module.

```
Option Compare Database
Option Explicit

'-----File specification for I/O using a type statement, which
'-----constitutes a user-defined type

Type waste
    ID As String * 10
    FacName As String * 50
    WhereFrom As String * 15
    WasteType As String * 20
    EstAmt As String * 10
    Cty As String * 30
    Comments As String * 30
    Endit As String * 2           'CR, LF
End Type
```

## **Reference: Here's the rest of the code:**

```
Public Sub ImportRandom()
Dim InRan As waste
Dim thisfile As String

'---prepare a table for the random file
Dim currdB As Database
Set currdB = CurrentDb
Dim RanTab As Recordset
Set RanTab = currdB.OpenRecordset("RandomTable", DB_OPEN_TABLE)
'---zap it if necessary
If Not (RanTab.BOF And RanTab.EOF) Then
    RanTab.MoveFirst
    Do While Not RanTab.EOF
        RanTab.Delete
        RanTab.MoveNext
    Loop
End If

'---open a random file for input
Forms!form1!Text4.SetFocus
thisfile = Forms!form1!Text4.Text
Debug.Print thisfile
Open thisfile For Random As #1 Len = Len(InRan)
Do While Not EOF(1)
    Get #1, , InRan 'get a record from the file
    '----add the record to the table
    RanTab.AddNew
        RanTab![ID] = InRan.ID
        RanTab![FacName] = InRan.FacName
        RanTab![WhereFrom] = InRan.WhereFrom
        RanTab![WasteType] = InRan.WasteType
        RanTab![EstAmt] = InRan.EstAmt
        RanTab![Cty] = InRan.Cty
        RanTab![Comments] = InRan.Comments
    RanTab.Update
Loop
Close #1
MsgBox "just filled up your random table!"
```

End Sub

---

```
Public Sub ImportSequential()  
  
'---prepare a table for the sequential file  
Dim currdB As Database  
Set currdB = CurrentDb  
Dim SecTab As Recordset  
Set SecTab = currdB.OpenRecordset("SequentialTable", DB_OPEN_TABLE)  
  
'---zap it if necessary  
If Not (SecTab.EOF And SecTab.EOF) Then  
    SecTab.MoveFirst  
    Do While Not SecTab.EOF  
        SecTab.Delete  
        SecTab.MoveNext  
    Loop  
End If  
  
'---get the file name  
Dim thisfile As String  
Forms!form1.Text11.SetFocus  
thisfile = Forms!form1.Text11.Text  
  
'dimension the working text fields  
Dim thisid, thisfac, thiswhere, thistype, thisamt, thiscty, thiscomment As String  
Dim one, two, three, four, five, six, seven As String  
  
'---open the input table for sequential access  
Open thisfile For Input As #1  
  
Do While Not EOF(1)  
    '---read the record  
    Input #1, one, two, three, four, five, six, seven  
    Debug.Print one, two, three, four, five, six, seven  
  
    '---append the record to the table  
    SecTab.AddNew  
        SecTab![ID] = one  
        SecTab![FacName] = two  
        SecTab![WhereFrom] = three  
        SecTab![WasteType] = four  
        SecTab![EstAmt] = five  
        SecTab![Cty] = six  
        SecTab![Comments] = IIf(seven = "", " ", seven)  
    SecTab.Update  
Loop  
  
Close #1  
SecTab.Close  
  
End Sub
```

---

```
Sub ExportRandom()  
  
'----dim corral  
Dim thisid As String, thisfac As String, thiswhere As String, thistype As String  
Dim thisamt As Double, thiscty As String, thiscomment As String  
  
'-----Get the file name and path for the text from the form's "text1" control  
Dim thisfile As String  
Forms!form1.Text1.SetFocus  
thisfile = Forms![form1]!Text1.Text  
MsgBox "the file/path name is " & thisfile  
  
'---open an output file  
Dim WasteRecord As waste ' Declare variable record as user-defined type "waste"
```

```

Open thisfile For Random As #1 Len = Len(WasteRecord)
' Close before reopening in another mode.

'---Open the recordset you want to output as text
Dim currdb As Database
Set currdb = CurrentDb
Dim myset As Recordset
Set myset = currdb.OpenRecordset("Select * from waste_1099 where type_of_waste = 'asbestos'")

'---Step through the recordset, and output each record to text
myset.MoveFirst 'get the first table row
Dim counter As Long
counter = 1
Do While Not myset.EOF
    '---use temp variables to store input record fields, test for nulls
    thisid = myset![ID_Number]
    thisfac = myset![Facility_Name]
    thiswhere = myset![Where_is_waste_from]
    thistype = myset![Type_of_Waste]
    thisamt = myset![Estimated_Amount]
    thiscty = myset![Cty_St_Ctry]
    thiscomment = IIf(IsNull(myset![Comments]), "", myset![Comments])

    '---store field contents in your user-defined type
    WasteRecord.ID = thisid
    WasteRecord.FacName = thisfac
    WasteRecord.WhereFrom = thiswhere
    WasteRecord.WasteType = thistype
    WasteRecord.EstAmt = CStr(thisamt)
    Debug.Print thisamt, WasteRecord.EstAmt
    WasteRecord.Cty = thiscty
    WasteRecord.Comments = thiscomment
    WasteRecord.Endit = Chr(13) & Chr(10) 'cr, lf
    '---write the record to text
    Put #1, , WasteRecord
    counter = counter + 1

myset.MoveNext 'get the next table row
If myset.EOF Then Exit Do
Loop
Close #1 'close the file
MsgBox "done!"
End Sub

```

---

```

Public Sub ExportSequential()
'---Open the recordset you want to output as text
Dim currdb As Database
Set currdb = CurrentDb
Dim myset As Recordset
Set myset = currdb.OpenRecordset("Select * from waste_1099 where type_of_waste = 'asbestos'")
'
'---get the file name
Dim thisfile As String
Forms!form1.Text8.SetFocus
thisfile = Forms!form1.Text8.Text

'dimension the working text fields
Dim thisid, thisfac, thiswhere, thistype, thisamt, thiscty, thiscomment As String

'---Open the file for sequential output
Open thisfile For Output As #1

'---Step through the recordset, and output each record to text
myset.MoveFirst 'get the first table row
Dim counter As Long
counter = 1
Do While Not myset.EOF
    '---use temp variables to store input record fields, test for nulls
    thisid = myset![ID_Number]

```

```
thisfac = myset![Facility_Name]
thiswhere = myset![Where_is_waste_from]
thistype = myset![Type_of_Waste]
thisamt = myset![Estimated_Amount]
thiscty = myset![Cty_St_Ctry]
thiscomment = IIf(IsNull(myset![Comments]), "", myset![Comments])

'---write to the file -- NOTE: the "Write" method inserts a CR, LF
Write #1, thisid, thisfac, thiswhere, thistype, thisamt, thiscty, thiscomment

counter = counter + 1
```

```
myset.MoveNext      'get the next table row
If myset.EOF Then Exit Do
Loop
Close #1            'close the file
myset.Close
MsgBox "done!"
End Sub
```