Updating The Size of an Existing Random Access File – Jim Brossman

Have you ever created a random access file in a freeware, shareware or commercial program only to find out that some time in the future you have to add a field to the file or increase size of a field? I have and always wondered how to make the update transparent to the user. Fortunately, my son is a software engineer for a major software corporation and showed me what he does. He places a field at the beginning of each record that contains the number of bytes in a record or the length of the record. Then at each update, he retrieves the record length from the first record and then knows what version he is working with and how to update the original file to one that will work with the updated program. The reason to place the record length in each record is if the file ever becomes sorted, the record length number will always be the first field in the first record assuming you do the sorting on the correct columns. You could place the number in the first record by itself and data in the next record but I find that cumbersome to deal with.

Shown in the code is a demo that will update a name file that originally contained only the last and first name. Now I want to have the ability to add a city to each record. Basically what is done is open the file with the new record length and fields, look at the first 5 bytes and see if it matches your new record length. If it doesn't, you know that:

1. There is no record length in the file and that the file is the original record length and using the select case in the UpdateNameFile sub you know how to update the record and fields for the new file size.
2. Or you know what version the file is and you can update to the newest version.

So each time you add a field you will need to add a new case in the select case section and you will modify each existing case to update to the latest record length and fields. You must also modify your file opening sub to the new length and fields and change NFRL at the beginning of the program.

This method can be used to update field lengths as well as adding fields. I use 5 bytes for the record length because if the original file happens to contain a number in the first field, it is unlikely that it would match a record length which will be 3 or four bytes at most. You could use a program version instead but it is unlikely that the program will become confused because of a number in the first record. In the demo I used an array to move the data from the old file to the new one. You could also open a temporary file of the correct format and move the data from the old file to the new temp, kill the old file and name the temp file to the original name. In fact, if you are shortening the record length for some reason, this is the preferred method otherwise you'll end up with data remaining at the end of the new file.

The demo program creates a file that contains only the first and last name and I want the ability to add a city to each record. Because I tend to use subroutines for a lot of things, I need several global variables, one of which is Name File Record Length or NFRL for the newest version and the length should be defined at the beginning of your program. In your program, you won't be killing the file at the end so the first time you run your program, the UpdateNameFile sub will see that your file isn't up to date, modify it, notify the user that the file was updated and continue. The next time you start the program, it will again go to the UpdateNameFile sub but it finds the correct NFRL and does nothing.

The program I had a problem with has the ability to back up and retrieve files. If a customer installs a program update and then retrieves some backup files that were saved with an older version, another section of the program calls the Update sub and automatically updates the backup files when they are retrieved and suggest to the user to back them up again because of the change.

```
'A method to keep track of and modify Random Access Files versions.
    'Jim Brossman 1/2/2010

    'Data to create an original RAF
    data Smith, Harry
    data Connley, James
    data Dewalt, Rose

    'Create the original data file with a record length of 45
    'and two fields.
    open DefaultDir$+"\NameFile.dat" for random as #NameFile len = 35
    field #NameFile, 15 as lastname$, 20 as firstname$
    for x=1 to 3
        read lastname$
        read firstname$
        put #NameFile, x
    next

    'Reset the field length and print contents of the Name File
    Field #NameFile, 35 as a$
    for x=1 to 3
        get #NameFile,x
        print a$
    next
    close #NameFile
    '----------------------------
    'At the opening of your program, these are the parts you need.
    'You will need an array big enough for the number of records.

'It might be multi dimensional, especially if you are changing the len
gth
    'of one of the existing fields.
    Dim NameArray$(100)

    'I like to have the update section and the opening of the random

'access file as subroutines so the new (N)ame (F)ile (R)ecord (L)ength

'needs to be global as well as the number of records in the name file.
    'I also need a flag to indicate the file has been updated.
    global NFRL, NUMBERRECORDS, FLAG

    'Set the New File Record Length at the beginning of your program.
    NFRL=50
    call UpdateNameFile  'Call the sub to see if updating is required.

    'I notify the user that their file has been updated.

'After the first time your program is run this notice will not appear.
    if FLAG=1 then
        Notice "Update Complete!"+chr$(13)+
"The Name File has been updated for this version."
        FLAG=0 'Reset the flag if necessary.
    end if
    '----------------------------

    'Show in this demo that the file has indeed been updated
    call OpenNameFile
    for x=1 to NUMBERRECORDS
        get #NameFile,x
        print NFRL;lastname$;firstname$;city$
    next
    close #NameFile
    'and then remove the demo file from your system.
    kill DefaultDir$+"\NameFile.dat"
    end



    'This sub opens the Name File with the new record length and
    'number of fields.
    sub OpenNameFile
        open DefaultDir$+"\NameFile.dat" for random as
 #NameFile len = 50
        field #NameFile, 5 as NFRL, 15 as lastname$, 20 as
 firstname$, 10 as city$
        NUMBERRECORDS=lof(#NameFile)/50
    end sub

    sub UpdateNameFile 'Update the name file to any new version.
        'This is to see if the name file exists.
        call OpenNameFile
        close #NameFile
        if NUMBERRECORDS>0 then
'If Name File exists and has records then

'open for a length of 5 because that is where I expect to find
```

```
            'the NFRL if it is there.
            open DefaultDir$+"\NameFile.dat" for random as
 #NameFile len = 5
            field #NameFile, 5 as recordlength
            get #NameFile, 1
'Get the first 5 characters in the Name File.
            close #NameFile
            select case recordlength
                 case 50 'Latest version of NFRL

'Up to date so do nothing.  The next time you change the
                'record length, this section will look similiar to

'the case else section and you need to add a new case with
                'the new record length that does nothing.
            case else
'recordlength does not equal any of the of the previous NFRLs.
                'Open with the original record length.
                open DefaultDir$+"\NameFile.dat" for random as
 #NameFile len = 35
                field #NameFile, 35 as a$
'Set the field to the entire record
                records=lof(#NameFile)/35
'Get the number of records in the old file.
                for x=1 to records
                    get #NameFile, x 'Get the entire record and
                    NameArray$(x)=a$ 'Put it into the array.
                next
                close #NameFile

'city$=""  Normally city$ would be empty but for this demo it will equ
al
                city$="Some City" 'Some City.
                'Open the file with the new record length.
                open DefaultDir$+"\NameFile.dat" for random as
 #NameFile len = 50

'Five as the NFRL, 35 as the original data and 10 as the new city$.
                field #NameFile, 5 as NFRL, 35 as a$, 10 as city$
                for x=1 to records
                    a$=NameArray$(x) 'Get the old data
                    put #NameFile, x 'and put all in the new file.
                next
                close #NameFile
                FLAG=1 'Set the flag to 1 if the file was updated.
        end select
```

```
    end if
end sub
```