

APPLE THREE USERS OF NORTHERN CALIFORNIA
VOLUME 5 NUMBERS 11, November 1, 1988
Converting Databases from
Apple /// to Another Machine.

by Hendrik Van der Straeten

As I said in a previous article, published in our Oct. issue, A/// is a fine machine, except it's not very fast. If you have to work with very large datafiles in a database (I personally use PFS), you can go away, have a cup of coffee while the A/// is searching the PFS-file. Unlike a printbuffer, you have to wait a long time before the report is printed.

There are many ways to speed up the programs (that maybe the subject of another article), but there are limits. In many cases I have to wait half an hour or more before the report from a large PFS-file is prepared and printed. I need faster searchings and reports for certain PFS-files. So I decided to convert my PFS-database to a Macintosh database. Here's how it's done. The principles are simple.

1. - What database do you use on the A///? In this case PFS-file. Is it possible to get from your database an ASCII-file with all the fields and the data ? Yes, it's possible for all databases on the A///. Solution for PFS : print to disk.
2. - Do a little magic with this file (stripping the fieldnames).
3. - Transfer this file to your Mac.
4. - Prepare the file for conversion to the Mac database. I use DBase Mac. In the following weeks I'll try other databases and conversions. If you are interested, send me a note and let me know what you want.
5. - Import your file in DBase Mac.
6. - Make changes in the database to fit your desired layout.

These principles are universal and in many cases you can exchange files between two databases on different computers and different operating systems : SOS , Prodos, MS-dos, Macintosh...

Let us go now step by step.

1. - A/// - PFS database and its output.

First, a few interesting things you have to know:

For my convenience I use on my A/// a Unidisk. I can put the files to transfer on the 3.25 inch. A Mac can read this disk. It is also possible to connect the Unidisk of the A/// immediately to your Mac (how? I'll tell you later). If you don't have a Unidisk, you have to transfer the files by a cable.

A fieldname in a PFS-file is always terminated by a colon (:). After the colon, normally you find a space. Otherwise, there are two possibilities: either you write or buy another Pascal program to prepare the outputfile, or you change your design of your PFS-file and add to the fieldlength one character.

A fieldname and its contents have to be on one line or you get some trouble.

There are solutions for it, but we assume here that all is on 1 line and there are no carriage returns in the contents of a field.

Change the design of your PFS-file and add a new field with the fieldname "help:". Put this new field on top of your design or at a place where you can reach it quickly.

Done? Now put in this new field a character that does not occur in the contents of the whole database. I use the percent sign (%) but you may use another. You can also take a short string, for instance "\$%\$". This character or string has to be unique and will be replaced later by a record delimiter in the future database.

You have to put the same character or string in the "help:" field of every record. Take care that there is a space between the colon of "help : " and your character or string. PFS puts by itself this space if your fieldlength is long enough. It is quite a task, but let POWERKEYS or THE DESKTOP MANGER (use the point on the numeric keypad and add an ENTER in your MACRO!) help you.

Select the PRINT function from the main menu and enter the desired filename.

Select the PRINT FORMS option. Now a blank form appears and you are asked to indicate which forms you want to print. You can now make a selection if necessary. Press ENTER and the PRINT OPTION menu appears. Answer on "print item names (Y/N):" with Y. Put after "Print to:" the file name where you want your outputfile. I took : ".U1/ SIGNAL" because I want to print to my Unidisk. Take a legal SOS-name. Take a short file name to avoid difficulties later on.

Now you are asked "lines per page:". If you are going to convert "x" fields per record, the answer is "x". (As you see, it's also possible to convert a part of a record). If you want all your fields : x = the total of fields you have in one record. Let's say that you have 20 fields in one record and you only need 6 fields in your DBase Mac. So put "lines per page:"6".

The number of copies is always 1. When all of the options contain the information, press ENTER and PFS will ask you to specify which items are to be printed to your file ".U1/SIGNAL". A blank form reappears. For each item (field) you want to print, enter the character "x" after the fieldname. The output of every field will be on a new line. You put now after 6 field names the "x". The last field has to be the added "help:" field. (If this field is not the last, you must redesign again your PFS-file). The "help:" field that contains your unique string will serve as a good recognized record delimiter for future use. Items are printed in the order as they appear on the screen. When the print specifications have been entered, press ENTER once again.

After the last form has been printed to disk the screen indicates the total of printed forms. Remember this total because it lets you verify at the end that all records are transferred. On your Unidisk (or on your hard disk) is now a file called "SIGNAL". It is a DATA file. There are differences between DATA files and ASCII files. For transformation, we need ASCII files. ASCII files have no internal format, they just are streams of bytes (known as ASCII characters).

If you add the suffix '.ASCII' to the name of your outputfile (.U1/SIGNAL.ASCII), Pascal creates automatically an ASCII file instead of a DATA file. But why not incorporate this option in a Pascal-program? It's a good exercise. So read on.

2. - Transformation in ASCII file and stripping the fieldnames.

Now the little magic. Because an empty field in the PFS database gives no output (no blank line) in the printed DATA file, we first printed all the fieldnames. Now we have to strip these fieldnames from the contents of the fields. For all empty fields, we should have a blank line.

Because we are A/// enthusiasts, let this work be done by the A///. The output file has to be an ASCII file. Let me give you an example of a UCSD Pascal program that does the job.

Suppose you have the following DATA file :

DATA file (as output from PFS): 1 record, last field = "help:"

```
field1: abracadabra
field2: 1234#
field15: firstname
field17:
field20: 123-456-789
help: %
```

And you want the following ASCII file with the fieldnames stripped : ASCII file (as input for your MACINTOSH) : 1 record, last string = "%"

```
abracadabra
1234
firstname

123-456-789
#%
```

These 6 lines are multiplied by the total of forms printed by PFS. Here is a Pascal program that does the desired job. It's a simple program, but does no checks! So, be careful!

Program STRIPPER ; {This program strips the first string terminated by a ':' from a}. {PFS data file created by the PRINT to DISK option in de PFS FILE menu}. {You have also to translate the datafile to an ASCII file}. {The result of the whole operation is an ASCII file}

```
var in file    :   text;
    out file   :   text;
in name, out name, dummy : string;
begin
write ('What is the original file ? Enter filename : --> ');
readln (in name);
reset (in file, in name);
write ('What is the destination file ? Enter name : --> ');
readln (out name);
out name := concat (out name, '.asci');
rewrite (out file, out name);
repeat
readln (in file, dummy);
delete (dummy,1,(pos(':',dummy))+1);
{after the ':' there must be a blank character }
{in the PFS-field - or you have to modify this program}
writeln (out file, dummy);
until (EOF(in file));
close (in file, lock);
close (out file, lock);
end.
```

By adding the suffix "ASCII" to the filename, Pascal creates automatically an ASCII file.

Once again I must warn you this program does no checks. If you want a good and safe program, write your own or buy one from the TAU library (Third Apple Users - 1113 Wheaton Oaks Drive - Wheaton IL 60187; ask for TAU-078 NVASCI by Bloom).

My "destination filename" was SIG. My ASCII outputfile from the above program was : U1/SIG.ASCI. Rename it for your convenience :.U1/SIG.

It is this ASCII file you have to transfer to your Mac. The Pascal program does a lot of work and it is quit time consuming. Why don't you take a break while the program is working?

3. File transfer from the A/// to the Macintosh.

There are different methods to exchange files between A/// and Mac. The simplest and most practical is by exchanging 3-1/4" diskettes and let the Mac read these disks (the files are in SOS or Prodos format).

Another method is to connect the two machines by a cable and to use on both machines a communication program (Access /// on the A///, Red Ryder or MS-Works on the Mac).

I use the first method here. The ASCII file to transfer is on the 3-1/4" disk and has the name : "SIG".

Start your Mac and use the program PASPORT or APPLE FILE EXCHANGE. Insert the disk and watch the Menu appear. There is an option "TEXT to TEXT" in the menu "Prodos to Mac". Disable the other possibilities. Start the transfer. A ASCII-file is created on the Mac with the same name "SIG". Quit the program.

4. - Preparing the file on the Mac.

From now on we work only on the Mac. It is impossible to use this file (or another ASCII file) to immediately import in dBase Mac. This file, named "SIG", is the foreign text file to change in the way dBase Mac can recognize it.

dBase Mac recognizes only 6 possible characters as field delimiters : comma (,), slash (/), period (.), colon (:), Tab, and Space. I use here the comma (,).

dBase Mac recognizes the following record terminators : a single carriage return, a single linefeed, a carriage return followed by a line feed, a line feed followed by a carriage return. I use the carriage return.

Take now your favorite word processor. I use MS-Word but all the others are good. Start MS-Word. Click CLOSE to close the current window. Click OPEN and choose the file SIG. The file SIG appears in the window.

First, count the lines, (Review the last page and count the lines on it. Add the total lines of all the previous pages to the number. Can it be divided by 6? Yes? So everything is OK; if not, you've lost some lines and have to restart the whole process from the beginning on the A///).

You notice very quickly every record delimiter. It's the "%" sign. Every new paragraph is the field delimiter. Change now every "end paragraph" (use in

MS-Word the characters "^p") by the colon (:). Take the "change all" option. The contents of the window change completely. Notice that now the record delimiter is : comma-%sign-comma (,%). Change now every occurrence of ',%, ' by a new line (this is in MS-Word :^n). The total of changes is the total of the records minus 1.

Look at the end of the document: you'll see : comma-%sign (,%). Cut these two characters from the document and add manually a CR. Every new record begins now on a new line. That's all you have to do. Do it carefully, it's important!

In most cases these are the basic principles to prepare an ASCII file for importing in a foreign database. To help dBase Mac recognize the file as a data- base file, let us create the fieldnames to use in dBase Mac. You can change these fieldnames later if you wish. Suppose you want to call the five fieldnames : a, b, c, d and e. At the top of your ASCII file in your word processor, add a new line and type : a,b,c,d,e. (Do not type the point after the 'e!'). dBase Mac will use these names (or characters) as fieldnames. Save the file as TEXT ONLY. Do not forget to take a different filename. Your ASCII file will look like this :

```
a,b,c,d,e
abracadabra,1234,firstname,,123-456-789
```

5. - Import your foreign text file in dBase Mac.

From this foreign text file, dBase Mac will create a "foreign structure" file and import the data to a dBase Mac file. The steps to do it are described in the manual of dBase Mac, so I'll not explain it here. It's easy but takes a little time if your text file is big.

6. - Make changes in the database to fit your desired layout.

That is now your job. Refer to the manual of dBase Mac.

POSTSCRIPT.

This is the whole story. If there is interest, I'll show you how to exchange files between an A/// and other machines. We are now working on file exchanging between PFS - A/// and 4th Dimension on a Macintosh.

Any questions or suggestions? let me hear from you. You can reach me at the following address :

AUBAN (Apple /// users of
Belgium and The Netherlands)
Hendrik Van der Straeten
Mechelsebaan 116
2850 KEERBERGEN
BELGIUM