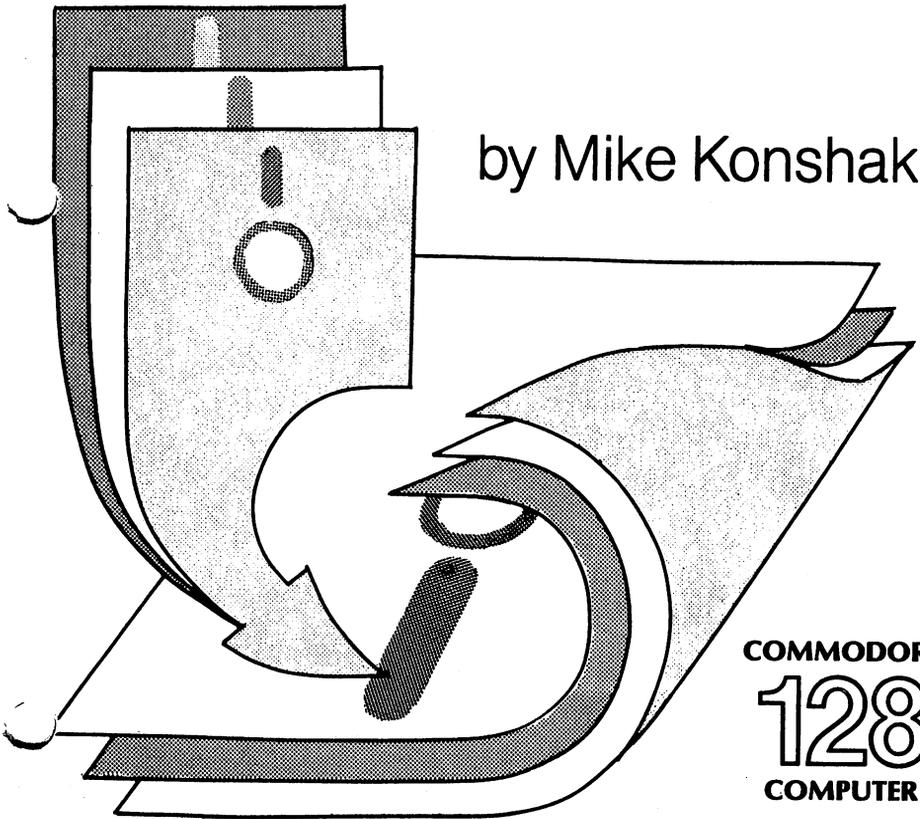


michaelsoft.

dfile. 128

DBMS File Utility Programs

by Mike Konshak



COMMODORE
128
COMPUTER



FORWARD

Hello, I'm Mike Konshak, the author of dfile128. I would like to thank you for purchasing these programs and for being a loyal user to dfile128. This utility disk contains many useful programs which increase the capacity and flexibility of dfile128 by allowing you to manipulate and modify your record files. These programs were created because they were requested, even demanded, by loyal users of dfile128. Of course, if you do not have the dfile128 Database Management System, these special utilities will not do you much good since they only perform operations on dfile128 record files.

dfile128 record files are created and edited while they reside in the upper 64K of memory in a Commodore 128 personal computer. Consequently, the number of records that are possible are limited by the size and number of the fields within a record. For records collected by most people, dfile128 is quite sufficient and capable of storing the user's data. In some cases however, a user may find it necessary to create several record files with identical structures in order to hold all their records.

Having multiple files is not a real problem with most folks since dfile128 is so easy to use. However, if you need to print out reports using multiple files, you end up with multiple reports. This is not very convenient, especially if you are performing calculations on the records. Even printing labels would be a major chore since you would have to reload each set of records. To solve this dilemma you will find MERGE FILES on the disk. This program will merge or combine up to 30 compatible record files into one huge record file.

Now that all of your files are combined, how do you print them out since dfile128 cannot read a single file as large as can be created with MERGE FILES? PRINT FILES will accomplish this task, since it will print record files based on defined formats directly from the disk. The PRINT FILE utility also operates on non-merged files to save you time in not having to load a file into the computer's memory. MERGE FILES and PRINT FILES together, are so helpful that their benefits alone more than makes up for the cost of the disk.

Everyone who uses databases from time to time discovers that they would like to create a file that contains some of the information in another file, but the file has a structure inconvenient to their current needs. The user is then presented with the dull task of retyping in the data all over again. I am personally always looking for ways to make my life more efficient and do not like to spend a lot of time duplicating my efforts.

CLONE FILE will save you a considerable number of wasted hours by cloning or copying the data you need out of one record file into a new record file. The new file can be a totally different structure than the old which will allow you to have more or less fields than the old file. You may even combine the data from several old fields into one of the new. Great! CLONE FILE allows you to use poor judgement when you first create a file since you have the capability to restructure the file later.

Now that should be enough flexibility for any database! But, many users wanted to use record files created by dfile128 with wordprocessors to create form letters (very infrequently), or wanted to transfer or export their files into a format compatible with other database management systems. For this reason, XPORT FILES resides on this disk also. XPORT FILE will strip all of the non-essential structure data found in a dfile128 record file and leave just the records. In some cases you might desire to use CLONE FILE beforehand to only leave the fields that you want to use in your word processor, or use MERGE FILE to combine the records before exporting.

Finally, I have provided a READ FILE program which will read the structure of record and format files. The structures and contents of the file may be printed to the screen or to your printer. READ FILE will become very handy in understanding your formats and become very useful when designing new formats. Not only will READ FILE read dfile128 files, it will also read any sequential file created by other programs.

If you look at the directory of the disk you will not see the utility programs listed by the names I have just described but as numbered programs. This is to insure that you always load the boot program first which will load some very important machine language routines needed by the other programs.

Many routines and operations within the programs are similar or identical to those found within the dfile128 database management system program. Since these routines are described within the DBMS system manual there will be many instances where I will refer to that manual instead of repeating everything again. Keep the system manual handy when running these programs.

It is possible you might have a need not covered by the dfile128 Utilities found on this disk. Please contact me if this is the case. dfile128 was created by listening to the needs of the average computerist, and most likely your needs are similar if not identical to those of others.

As with all the programs from michaelsoft, this disk is intended to be easily copied so that you will always have backups in case of an accident. I believe that if I am fair to you, you will also be fair to me. The prices of our software are kept low to allow everyone to be able to afford them. Please do not make copies for friends. You would be not only be cheating yourself, since you paid for your copy, but would also be cheating our family out of deserved and necessary income.

Thanks for your support,



Mike and Becky Konshak
4821 Harvest Court
Colorado Springs, CO 80917 USA
303/596-4243

LOADING dfile128 UTILITIES

The Commodore 128 personal computer provides many ways to begin using the dfile128 utility programs. The most obvious way is by inserting the UTILITY disk into Device 8, Drive 0, and typing in the following:

```
dload"dfile util" <Return>
run <Return>
```

Typing run"*" also works on the Commodore 128 for this and other programs where the intended program is first on the disk directory. Pressing the SHIFT and RUN/STOP key simultaneously will start the disk even easier.

The utility disk has a boot file record which also will automatically load the starting program when you reset the computer or when you first turn it on. Be sure the disk drive is already on before you insert the disk to protect your programs from current surges within the drive.

When you see the following menu, press the appropriate key to load the operation that you wish to use. Keep your Utility disk in the drive until your choice has been loaded.

Select dfile128 File Utility Program

```
Press PRINT Record File from Disk
MERGE Compatible Record Files
CLONE an Existing Record File
XPORT an Existing Record File
READ the Structure of dfile128 Files
Quit dfile128 Utility Programs
```

Press the Appropriate Key

Make your selection by pressing a single key highlighted in reverse print on the screen. It is not necessary, and in some instances not permitted, to use the SHIFT key to enter the appropriate selection. Once your choice has been successfully loaded, you will be presented with another menu which looks almost like the previous one. The exception being that the program title is at the top of the screen. You can continue into the program or transfer to another Utility if you have changed your mind at this point. Remember to put your Utility disk in device 8, drive 0 before transferring to another utility program.

You might reference the system manual as to the expected responses to prompts within the dfile128 series of programs.

TABLE OF CONTENTS

Loading dfile128 Utilities.....	4
Selecting a Utility Program.....	4
PRINT FILE Program Description.....	5
MERGE FILE Program Description.....	6
Guidelines for Merging Record Files.....	7
Utility Programs Main Menu.....	8
Selecting Files for Merging.....	8
Merging Operation.....	9
CLONE FILE Program Description.....	10
Guidelines for Cloning.....	10
Breaking Apart Existing Files.....	11
Cloning a Sample File.....	12
XPORT FILE Program Description.....	14
READ FILE Program Description.....	15
Record File Structure.....	16
Report Format File Structure.....	16
Mailing Label Format File Structure.....	16
Calculated Report Format File Structure.....	17
Examples of File Structure Printouts.....	18
Appendix - Printer Codes for Compressed Print.....	20

PRINT FILE

PRINT FILE works exactly like the print routines found in the dfile128 DBMS program except that it prints records directly from the disk instead of the memory. This allows printing of very large files created by MERGE FILE. Any dfile128 compatible record file may also be printed with PRINT FILE. Pre-existing formats may be used or new report and label formats may be created. As in the system program, printing may be directed to the screen as well as the printer.

You will discover that printing directly from the disk is actually faster with PRINT FILE than with the dfile128 DBMS since you would not have to wait for the records to be loaded into memory. This assumes that the records are already in the current form and sorted order on the disk and do not need editing. A mailing label list that gets printed weekly would be a good example. Report titles may be changed directly by PRINT FILE without having to load dfile128, in the case where you might want to alter a date.

All the printing formats, Labels, Reports and Calculated Reports, may be created or changed as needed from within the PRINT FILE utility program. Unformatted lists are also available. The only drawback to PRINT FILE is its inability to randomly select individual records one after another for printing since the file is searched from start to finish. For this capability you must use dfile128. Keep in mind that dfile128 cannot load most merged files.

To use the print utility, you must first identify which record file you will be printing. After that, the routines are the same as you would find in dfile128. The records will be printed in the same sorted order as was last saved by dfile128. Refer to the system manual for an explanation of the various menus and routines that will appear.

As with dfile128 you are given four printer options. Print option 4 prints to the screen while the other three are used for obtaining hardcopy printouts from various printer and interface combinations. Option 2 should be chosen for most non-Commodore printers using serial interface adapters. Since you have the ability to print 132 character wide reports with dfile128 and PRINT FILE it might be wise to consider a non-Commodore printer (or one with a 15 1/2 inch carriage) since most Commodore printers cannot print in a compressed character mode (12-15 CPI). When using option 2, set your interface in pure ASCII mode, not in the 1525 emulation mode.

The appendix at the back of the manual lists the codes that will invoke compressed print for various popular printers. These codes may be sent to the printer by dfile128 or PRINT FILE when you access the SEND PRINTER COMMANDS routine at the PRINT OPTIONS MENU.

MERGE FILES

The world and our personal environment never stands still and what was adequate yesterday may not be applicable today. Too often we have drastically underestimated the growth of our expenditures, or the increased number of members joining our church or club. Our collections and possessions may expand to a point that a record file that was previously created is no longer large enough to store all the records. Most users of dfile128, having the foresight to anticipate growth, or having started out with a large number of records, have opted to create several compatible record files in which to store record data. In most cases the record files were categorized to contain specific groupings of record data, typically alphabetically or by month or date. An example of such files would be a club member list or church directory.

Let us assume that you are the secretary of a computer club consisting of 490 members. You anticipate the growth of the club to reach approximately 1500 members because you live in a large city and your user's group is based on a popular computer. When you first created your dfile128 record file, you discovered that dfile128 only permitted a total of 400 possible records. This was based on the number and length of the fields that you needed to adequately store the information about each member. Being devilishly clever, you set up four record files and designated, by the name of each file, to contain alphabetical categories of members. You entered each member's name, address and equipment information and ended up with the following files:

Name of Record File	# Records Possible	# Records in File
members a-g	400	150
members h-m	400	110
members n-s	400	135
members t-z	400	105
Total of (4) Files:	1600	490

Although there are currently only 490 members in the club there is now room for growth and you will not have to change your database for quite a while. Separating the members into categories as shown above is a good idea, because you would be able to sort each file within the category to put the records in alphabetical order. If reports were printed, listing all the members, there would be four separate reports, because of the four individual files. This is not totally acceptable in some cases which is the main the reason for merging files.

MERGE FILES will merge up to 30 identically structured dfile128 record files. Merging identically structured files provides the advantage of having one uninterrupted printing session which saves a considerable amount of time in not having to load up individual files one-at-a-time. Another advantage during printing is that all the data in several files may be used in calculations or listed in one report. Once you start the printing process you can just walk away from the computer (and

the noisy printer). The resulting merged file will most likely be too large to be loaded up into dfile128 for printing, but may be printed using PRINT FILE. Merged files may be exported, using the XPORT FILE for importing into other programs and databases.

To merge files, keep the following points in mind:

- 1) One or two single disk drives may be used or a dual drive.
- 2) If one disk drive is used, the source files to be merged must all reside on the same disk.
- 3) The total number of blocks that the source files consume cannot exceed the number of blocks free on the destination disk.
- 4) Once merged, the file may be too large to be loaded by dfile128. If the number of records combined into the merged file is less than the number of possible records of one of the files, you will be able to load the merged file into dfile128 for editing and sorting. You might have to use the CONVERT DATAFILE program on the system disk or the CLONE FILE program on the utility disk to recalculate the number of possible records, since merged files change the number of possible records to equal the number of actual records in the merged file.
- 5) Records will be loaded in the original order that they were saved onto the disk, and the files will be loaded in the order as they were typed in. Consider this point if you are merging files that have records put into files separated by alphabetical categories to insure that your reports will be printed alphabetically also.
- 6) All files to be merged must have the same identical structure or at the very least, the same number of fields. Field lengths and names may differ from file to file. The merged file will retain the structure of the first file that is merged.
- 7) Once merged, your files may be printed or exported using PRINT FILE or EXPORT FILE since these programs operate directly off of the disk drive, but cloning of files by CLONE FILE may not be performed since this program, like dfile128 itself, operates out of arrays in memory.
- 8) One (1) to 30 files may be merged together. The capability to merge only one (1) file permits using MERGE FILE as a backup program for making copies of merged record files onto separate disks. You will need two drives to make backups of single files that are to be on separate disks. dfile128 will copy single non-merged files simply by saving (or writing) the files as many times as you desire. If non-merged files are to be copied by MERGE FILE and are still destined to be loaded into dfile128, then the number of records possible will have to be recalculated by running CONVERT DATAFILE or CLONE FILE.
- 9) Record files may be merged with themselves to create large dummy files for experimenting with reports and testing of other programs.

When you first enter MERGE FILE, by pressing M at one of the other programs's main menu, you will be confronted with the following selections:

dfile!28 Multiple File Merge Utility

Press Merge Compatible Record Files
 Scan Records in File
 Directory \$
 Quit Program

or Transfer to:

PRINT Record File from Disk
 CLONE an Existing Record File
 XPORT an Existing Record File

Press the Appropriate Key

Pressing P, C or X transfers you to the other utility programs mentioned while Q quits the program altogether. Pressing D or \$ will obtain the directory on the disk. You will be asked for the device number (8 or 9) of the drive containing the disk to be read for its directory. The other utility programs have similar main menus.

Pressing S allows you to SCAN a record file either before merging or after merging to help you see the data within the record file. Use this routine to check compatibility of files to be merged or to see the final result of your efforts.

Pressing M actually starts the merging process by asking which device and drive contains the source files to be merged. Because the program is capable of using all types of drives produced by Commodore you will have to enter device 8 or 9 and drive 0 or 1. Remember, all the source files must reside on the same disk. You will then be asked for the device and drive numbers of the destination disk. This may be the same as the source disk if only one drive is available.

You will be prompted as to the number of files to be merged and the record file names of the source files. You will then enter the name for your merged or destination record file. The screen will look something like this (using the club member files mentioned previously):

```

Source Files are on Device? 8      press <RET> or
                                Drive ? 0      change number
Merging will be on Device? 9
                                Drive ? 0

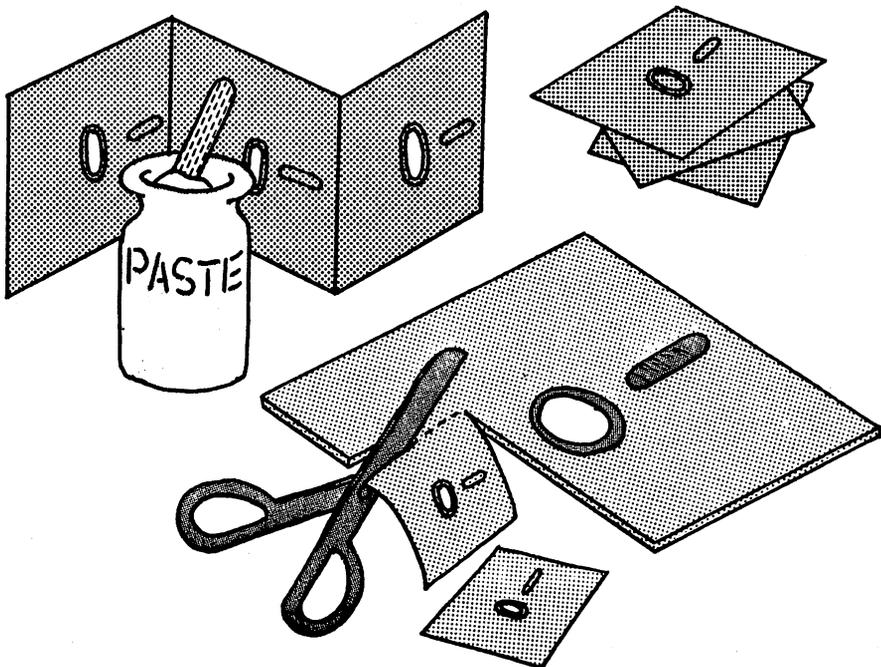
Number of Files to Merge ? 4
Source File Name #1 ? members a-g
Source File Name #2 ? members h-m
Source File Name #3 ? members n-s
Source File Name #4 ? members t-z

Merged File Name? members
  
```

As you can see, the merged file name will be simply members since all the categories can now be combined. Entering 0 for the number of files or a blank (null) for the names of files will exit you to the starting menu.

Once the above information has been entered, the program will perform the following operations:

- 1) Determine if the source files have the same number of fields and are structurally compatible. The merged file will use the same field names and field lengths as were in the first file.
- 2) Count the number of total records in each file to determine how large the merged file will be.
- 3) Read the records, starting with the records in the first source file, and write the records into the merged file.
- 4) Display which record number of the source file is being read and which record number is being written in the merged file. The ending record number in the merged file will correspond to the total number of records present in all the source files.
- 5) The files will be closed and you will be given the opportunity to merge additional files or to transfer to the other utility programs.



CLONE FILE

Home and business records typically have similar data which is used in various different functions. Many times identical data is required within several files even though the files may be used for different purposes. A good example would be a church membership which requires member's names to appear in separate files such as mailing lists, financial contributions, and sunday school records. Items on a warehouse inventory file may also show up on a bill of materials for a product in the manufacturing plant. It becomes apparent that some of the data in a particular file may be needed for another purpose in another file. Creating new files for new applications and typing in data which is already present in other files is a wastes considerable time and costs more money in man-hours. Even if you are creating files for home use, I am sure you have better things to do.

At some time you might also discover that record file structures that you had created had more fields than were really necessary (or never used) and you would like to eliminate one or more of these fields. This would also allow more records possible within the file. You might also discover that you needed to add additional fields because you needed more information within the records. Creating new record files will solve your problem but you would still have to re-type in all your records again. Not good, undesirable, and no fun.

CLONE FILE will cure all the previously described situations by copying record data from an existing record file into a new file. The ability to clone or transfer record data from one file to another will pay for the cost of dfile128 Utilities many times over (if only time is saved you would at least be able to spend that time with your spouse or family).

In transferring or cloning records, you may:

- 1) Clone all or part of the records in an existing record file into a new file with identical structures.
- 2) Clone all or part of the records in an existing record file into a new file which has a totally different structure (number of fields, names and length of fields). The new file structure may have less or more fields than the old file.
- 3) Combine up to three (3) record fields of an existing record file structure into one record field in the new record file structure.

CLONE FILE will allow you to create a new file, for example, having 10 fields, the clone the records from a file having only six (6) fields (the inverse is also possible). The program will allow you to combine up to three (3) of the old record fields into one of the new record fields and will allow you to change the order of the fields as they appear within the records.

In cloning records, several logical operators (=, >, <, <>, and ??) may be used to determine which records should be transferred. Up to three operators, record fields and record data may be used to search for the proper match to insure that

you only transfer record data that you desire. The logical search routines operate identically to the search routines for printing records in PRINT FILE or dfile128. Refer to the DBMS system manual for guidelines on logical searches.

CLONE FILE can be used to break apart an existing file that ran out of memory space, for example a video tape library that grew larger than was expected. You would be able to create several new record files having the same structure, but dedicated to separate groupings of tape categories, in order to allow more growth. Examine the following groupings that were created from totally filled old files:

Old File	Possible New File Groupings (Catagories)
members	members a-g, members h-n, members m-s, members t-z
computerclub	club pet, club vic, club c64, club 128, club amiga
authors	fiction, nonfict, mystery, scifi, history, romance
videotape	video g, video pg, video r, video x, video music
yearly sales	jan sales, feb sales, mar sales,dec sales

CLONE FILES can be used to help extract essential fields for use in wordprocessing to create form letters (in conjunction with the XPORT FILE program to convert the file after cloning). Let us assume a form letter requires a First and Last, followed by Address, then the City, State and Zip (three lines). The record file that we want to use, mail list, has eight (8) fields and we need to transfer all of the records into a new file containing only three (3) fields. Here is what the old and new files will look like:

Old Structure		New Structure		
#	Field	#	Field	Contains Old Field #'s
1	Last (name)	1	Name	2 (First) + 1 (Last)
2	First (name)	2	Address	4 (Address)
3	Code	3	CitySTZip	5 (City) + 6 (ST) + 7 (Zip)
4	Address			
5	City			
6	ST			
7	Zip			
8	Phone			

After determining the structure, you might transfer ALL the records or FIND (search) records with common items. Even though not every field was transferred to the new file, we are still able to use the fields in the old file to search by in order to only obtain those records that we really want in the new file.

Creating a cloned file is much the same as creating a new record file in dfile128 whereas you must choose the number of record fields and the title and lengths of the fields within within the new file. Choosing the fields that you want transferred from the old file into the new is much the same as formatting a label format whereas each row of the label can contain up to three (3) record fields.

When CLONE FILE is first entered you will be asked for the name of the existing record file which is to be cloned. Once the file name has been entered, the program will load the entire record file into memory. Because it is necessary to read records resident within memory for cloning purposes, it is not possible to clone merged files from the disk.

Let us use the file mail list again to clone a new file with a different structure. The record file structure will be displayed on the screen as shown (40 column mode):

```
Field Name (Length) for mail list
1 Last Name 15   5 City 23
2 First Name 10  6 State 2
3 Code 5       7 Zip 5
4 Street      8 Phone 12
```

```
384 Records Possible in Old File
4 Records Currently Present
```

Clone Structure as Above? Yes No Exit

Pressing Y will create a new file with the exact structure as the old file. The program will advance directly to the Transfer Record Data to New File routine.

If N is pressed, which is our choice at this time, we will have to create our new file by designing the number of fields, etc. Before we get too specific. Lets describe what the new file will look like.

Our goal is to create a new simplified address file which will be used to send mail internationally. The new file must combine the data found in several fields of the old file and must provide an additional field so that we may enter the country as part of the address. The following is our intended structure:

#	Field Name	Contents	Length (chrs)
1	Name	First Name + Last Name	32
2	Street	Street Address	32
3	CitySTZip	City + State + Zip	32
4	Country	Country	16

It appears we will be able to use most of the fields in mail list toward the new file. Actually, all the fields may be filled from the old file except for the country field. Lets continue with the program by answering the following prompts as shown:

Number of Fields per Record? 4

```
Field #1
Title ? Name
Length ? 32
```

Keep entering the field title and length data as prompted for fields 2 thru 4 as we defined earlier. When the initial structure is defined you will be presented with:

Your selections will allow 468 Records
 Press Accept Reject Change Field Data

You have the opportunity at this time to correct spelling errors or make slight modifications to the fields. Pressing A will advance us into the routine which selects which old fields get transferred to the new. Old field values will be pre-printed for you in most cases after the prompts. If the number shown is not the one you wish, just over type the prompted number with the desired one.

Up to Three of the Old Fields May be
 Combined into one of the New Fields.

Enter Fields to be Cloned into New
 File as Prompted. Enter 0 to Leave
 New Field Empty.

New Field 1 Name
 Old Field 1 ? 2
 2 ? 1
 3 ? 0

New Field 2 Street
 Old Field 1 ? 4
 2 ? 0
 3 ? 0

New Field 3 CitySTZip
 Old Field 1 ? 5
 2 ? 6
 3 ? 7

New Field 4 Country
 Old Field 1 ? 0
 2 ? 0
 3 ? 0

Since Country is a new field that does not have a counterpart in the old file, 0's were entered to leave the field blank. Once all the data is entered the program will ask:

Is This the intended Clone? Yes No

New File	Old File
1 Name	= First Name Last Name
2 Street	= Street
3 CitySTZip	= City State Zip
4 Country	=

If everything is as you expected it then press Y to continue to the next routine which will transfer the records. Refer to the dfile128 system manual for help with the logical operators.

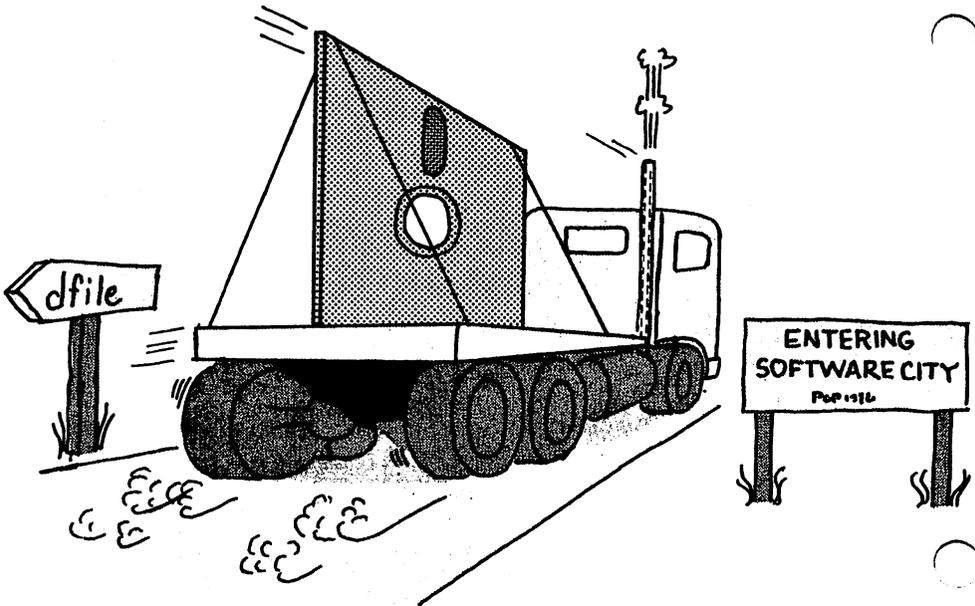
XPORT FILE

This XPORT FILE program is designed to export or transfer dfile128 record files into the standard form expected by other programs using sequential files. All non-record data information within the file that is used by dfile128 is stripped from the file leaving just the record data.

Since programs vary in design and purpose, it is very important to review the manual of the program that is to be the recipient of your records. It is there that you should discover the operations necessary to successfully import the modified dfile128 record file. You might find also, that the application that you intended to use does not require every field that was in the file. Use CLONE FILE to separate out the required fields.

XPORT FILE is very simple to use. The only decision that has to be made is whether or not to add an additional carriage return between each record in the file. Your destination program may or may not require the extra space. In order to minimize confusion, the exported file may contain the same name as the original because the special file name identification characters are also removed during exporting.

As with most of the programs on the dfile128 utility disk, you are able to scan the results from the program's main menu. Take a look at your exported file to get a better understanding of the records. You can use the READ FILE structure program to get a hardcopy printout of your files.



READ FILE

One of the basic requirements in creating formatted reports that operate off of files, like those produced by dfile128, is an understanding of the structure of the file. The structure consists of the length of various fields and variables, as well as the relationship of the various factors. Sometimes, just knowing what data is in a file is sufficient, in order to assist in debugging an improperly programmed or formatted file.

READ FILE allows the user to peek at the sequential files created by dfile128, as well as any sequential file that was created by other programs not related to dfile128.

The sequential files are written (or saved) to the disk using special codes. Each record file and format file type has a unique code which precedes the file name. It is possible for every program to create files with the same name. The only difference, which only becomes apparent when you view the disk directory, is the special characters that precede the file names.

The following codes, which precede the file name sample file, are written by the respective routines:

- * Record file. df] sample file
- * Report format file rp] sample file
- * Mailing label format file. ml] sample file
- * Calculated report format file. cr] sample file
- * Exported or Non-dfile128 file. sample file

Every sequential file in the Commodore operating system is further identified on the directory with a SEQ which follows the file name.

Upon entering the READ FILE program you will be presented with the following menu:

Record File or Format File Selection

```
Dfile Record Files
Report Format Files
Mailing Label Format Files
Calculated Report Format Files
Non-dfile128 Sequential File
$ Disk Directory
Quit Program
```

Press the Appropriate Key

Pressing \$ will display the directory of the disk in device 8, drive 0. When viewing the directory of files created by dfile128, you will notice the file names preceded by two characters and a right bracket (ie. df] ...). These special codes allow the use of identical names for files, saving you considerable typing, while keeping the names separate for the disk operating system.

Pressing D, R, M, C or N will initiate a prompt for the name of your sequential file. If the file is from dfile128, just enter the names without the prefixes. If the file is an exported file or a non-dfile128 file, then type in the name exactly as it appears on the directory.

After entering the name of the sequential file, the program will search out the data from the disk in device 8, drive 0, and display the structure of the file. The light on the disk drive will stay on until you actually return to the beginning menu, so do not remove the disk until then. The routines are very simple to operate. Just follow the prompts at the bottom of the screen. You will notice that you are able to print the structure of your files by pressing the appropriate keys.

Record File Structures

Each of the various files created by the dfile128 system will have different information contained within a structure display. Record files contain the following information:

- * Number of records possible in file.
- * Number of records currently in the file.
- * Number of fields within each record.
- * Title of each field.
- * Length of each field.
- * Data contained in each record in fields.

The actual string and numerical data, as they appear on the disk, may be viewed by selecting the Non-dfile128 Sequential File option at the main menu. This will show you the exact sequence of the contents the record file as it resides on the disk. The data will be viewed serially, without any format or labels. If you use this option, the file name must be entered exactly as it appears on the disk directory (ie. mail list is actually df] mail list on the directory).

Report Format File Structure

Report format files, used for the purpose of producing printed reports in custom forms, will contain the following:

- * Number of lines in report title.
- * Data for each title line.
- * Number of columns in the report.
- * Position of each column.
- * Header data for each column.
- * Which record fields appear in each column.
- * Was totalling of the last column chosen.

Mailing Label Format Structure

Label format files are used to print dfile128 records on labels. The custom label structure file contains the following:

- * Number of rows on label.
- * Which record fields appear in each row.

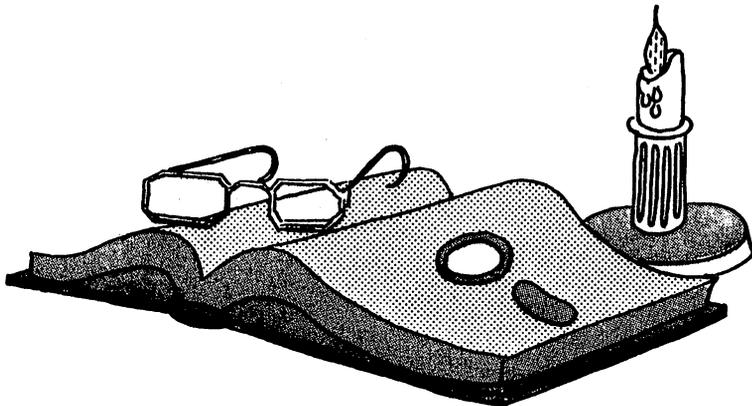
Calculated Report Format File Structures

Calculated reports, simulate the type of reports produced by spreadsheets. These format file structures contain the following data:

- * Number of lines in report title.
- * Data in each title line.
- * Number of columns in report.
- * Position of each column.
- * Header data for each column.
- * Contents of each column (Field, Rec#, Equation).
- * Equation(s) for column, if chosen.
- * Justification of each column.
- * End-of-column operation (Total, Average, NOOP).

READ FILE is especially useful when dealing with calculated reports because it allows you to use information found in other report formats to help you design new report formats.

On the following two pages are examples of printouts of various file structures within dfile128. Examine the examples carefully, especially if you are having difficulties with report formatting.



EXAMPLE PRINTOUTS OF FILE STRUCTURES

Structure for dfile128 ** mail list **

Records Possible in File: 392
Records in Current File : 4
Fields in each Records : 8

Table with 3 columns: #, Title of Fields, Length. Rows include Last Name (15), First Name (10), Code (5), Street (32), City (23), State (2), Zip (5), Phone (12).

Structure of Report Format ** mail list **

Lines in Report Title: 2

Title 1 : Mail List Records
Title 2 : January 23 1984

Columns in Report : 5

Table with 3 columns: #, Pos Header, Contents. Rows describe field positions and counts like 1..Last/First Name (Fields:1+2+0).

No Totaling of Last Column

Structure of Mail Format ** mail list **

Rows on Label: 3

Table with 2 columns: #, Contents of Label (Record Fields). Rows show field groupings like 2+1+0, 4+0+0, 5+6+7.

EXAMPLES OF FILE STRUCTURES (cont)

Structure for Record File ** grocerystore **

 # Records Possible in File: 754
 # Records in Current File : 19
 # Fields in each Record : 6

#	Title of Fields	Length
1	Item.....	16
2	Stock#.....	6
3	Unit.....	4
4	Quantity.....	4
5	Cost\$.....	8
6	Retail\$.....	8

Structure of Calculated Report Format ** grocerystore **

 # Lines in Report Title: 2

 Title 1 : Ma and Pa's Corner Store
 Title 2 : Smalltown - U.S.A.

Columns in Report: 10

#	Pos	Header	Contents/Equation	Just	EOC
1	1	..Rec#.....#.....F.....	n
2	7	..Grocery Item....f1.....l.....	n
3	23	..Stock#.....f2.....l.....	n
4	31	..UN.....f3.....l.....	n
5	35	..QTY.....f4.....r.....	n
6	40	..Cost\$.....f5.....c.....	a
7	48	..Price\$.....f6.....c.....	a
8	56	..TTL CST.....e..f4*f5.....c.....	t
9	66	..TTL PRC.....e..c5*c7.....c.....	t
10	76	..%MUP.....e..(f6-f5)/f5.....p1.....	a

Note that the alphanumeric characters within the contents of columns are entered in lowercase letters. Uppercase characters are not permitted in these areas.

APPENDIX - Printer Codes

dfile128 and PRINT FILE both provide routines which will allow the sending of printer codes to printers. These codes are used to set the printer into modes which allow more capability in your printouts. Popular modes are compressed print, italics, expanded characters, different fonts, etc. Most Commodore printers do not allow compressed print modes. You must inspect your printer's manual to see what your printer is capable of.

Codes are sent by entering numbers corresponding to the CHR\$() commands listed under the options for your printer. If your manual tells you to send a CHR\$(27)CHR\$(15), you would only enter 27 and 15 when prompted.

Codes to put some popular printers into a compressed mode are as follows:

Printer	12 CPI	17 CPI
Gemini, Star Micronics	27, 66, 2	27, 66, 3
Okidata 82A		29 (16.5 CPI)
Okidata 92A	28	29
Epson RX80 F/T	27, 77	15
CITOH Prowriter	27, 69	27, 81

Some printers, especially daisy wheel printers, provide a switch that will easily change the print pitch from the normal 10 CPI to 12 and 17 CPI. This would obviously be the way to go if you have this feature.

NOTE: Do not use codes that skip over perforations since dfile128 handles this during printing.

Use the following space for printer codes for your printer.

DFCopy

Tailor your Datafile files to meet your particular needs.



By PAUL TURNER

I'm fascinated by Mike Konshak's Datafile series of programs. In my opinion, they're among the most powerful and easiest-to-use data management programs available.

In using Datafile, I've often needed special-purpose files composed of only some of the records in my Datafile files—for instance, a file of zip codes copied from an address file. So I wrote my own program, called DFCopy, to create them. (Naturally, the DF comes from Mike.) Although DFCopy uses routines similar to Mike's, it is in no way intended to infringe on his "claim to fame." Instead, I intend that it work in conjunction with the DF series of programs to provide additional data-management flexibility.

As I indicated, DFCopy creates a new file by copying records from a Datafile file that already exists. It copies only those records from the source file that meet certain specifications you input. It doesn't alter the source file in any way, nor does it alter the records it copies. It just moves them.

The beauty of DFCopy is that it eliminates any need to alter a source file's structure to tailor its data to special needs or to reenter data that you've already typed into the source file.

DFCOPYING

To use DFCopy, first enter LOAD "DFCOPY",8 and then RUN. The first screen that appears is an introduction. Next, the program asks for the name of the Datafile file you want to use as a source from which to copy data. Your

response doesn't need to include the special characters, DF], that precede the filename. If you press return without entering a name, the program goes back to the introductory screen.

Assuming you have entered the name of a source file, the computer reads that file into memory and displays a list of its field names and lengths, then asks you to specify the field you want to work with. A return without specifying a field brings you back to the screen for naming the source file.

Assuming you want to proceed, enter the field number. Then a list of logical operators will appear. DFCopy will use the operators that you specify for identifying data in the source file to be copied to the new file.

You'll find the operators listed in Table 1, along with their results for a sample run of DFCopy. The table assumes that the source file contains addresses and that its zip-code fields are the ones to be copied. It assumes further that all the zip codes in the source file are a full five digits long.

Back to the operation screen, enter the numbers for the logical operators you want to use (or zero to return to the field-selection screen). Any operators you choose will be listed at the top of the screen for reference.

Then you'll be asked to input the constant data—that is, the data strings DFCopy should search for in the source fields. The maximum length for each string is XXX. A return with no input of constant data brings you back to the field-selection display.

Table 1. Sample logical operations for selecting data to be copied from a source file.

Number	Logical operator	Meaning	User input	Zip codes extracted from source file
1		None; exit		
2	=	Equal to	019##	01900-01999
3	>	Greater than	019##	02000-99999
4	<	Less than	019##	00000-01899
5	<>	Not equal to	019##	00000-01899 02000-99999 (01900-01999 aren't selected)
6	> =	Greater than or equal to	019##	01900-99999
7	< =	Less than or equal to	019##	00000-01999

Refer again to the table, and note that the data string you input can be shorter than the length of the fields it'll be compared to. Just fill out the string with spaces. (That's what the # symbols in the table represent.) I coded the comparison this way to provide maximum flexibility in record selection.

After you've specified the operators and data strings you want to use, the program lists them for your final okay. An R for reject retrieves the operations-available screen; an A for accept starts construction of your new file. If DFCopy doesn't find any fields that meet your criteria, a message to that effect appears and the program returns to the field-selection screen.

If DFCopy does find some fields that meet your criteria, it asks for a name for the new file. A return with no name input brings back the field-selection

screen. If you do name the file, DFCopy writes the file to disk. Then the field-selection screen comes back, giving you an opportunity to create another special file from the same source file.

Two special files can be built from a source file before you must read the source file into memory again. This limit lessens the chance of the computer going into "garbage collection."

DATAFILE 3.6

Those who've upgraded to Datafile 3.6 (see *RUN*, January 1987) and want to use the DOS 5.1 Wedge should remove the REM statements in lines 110, 120 and 300 of DFCopy. Also, be sure that the wedge is on the same disk as DFCopy. ☐

Paul Turner is a systems accountant and analyst for a manufacturing firm.

michaelsoft™

A COTTAGE INDUSTRY OF HOME SPUN SOFTWARE

4821 Harvest Court
Colorado Springs, CO 80917 USA
(303) 596-4243