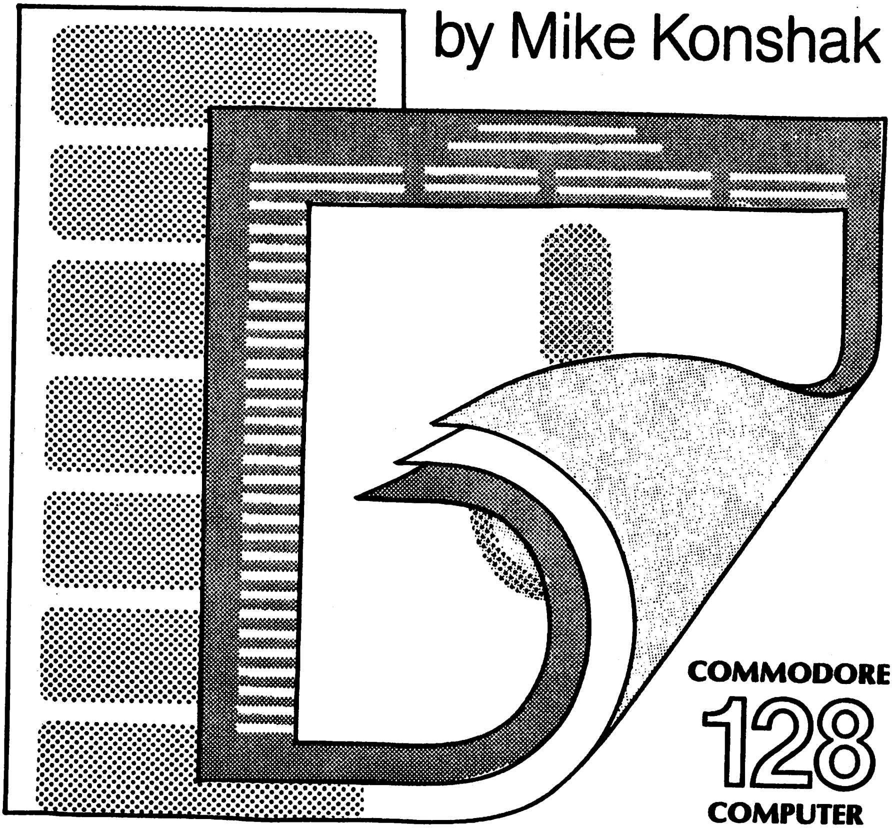
michaelsoft™

dfile™ 128

Database Management System



by Mike Konshak

COMMODORE
128
COMPUTER

michaelsoft™
A COTTAGE INDUSTRY OF HOMESPUN SOFTWARE

dfile128 ERRATA
Compiled Version

The dfile128 database management system has been significantly improved since the printing of this manual. The program has been compiled into machine code in order to speed up the operation of various routines such as sorting. Consequently, several changes to the manual need to be mentioned.

page 6 - Loading and running may also be accomplished by first turning on the disk drive (Drive 0/Device 8), inserting the dfile128 system disk, then turning on the computer. The program will automatically be booted into the computer. DO NOT turn the disk drive on or off with the dfile128 disk inserted into the drive with the door closed.

page 6 - Note 5 - Because the program is in machine language, it is impossible, at this time, to re-enter the program without losing the file data held in memory. The premature termination of the program by the operating system or by bad operator input should not occur since most errors have been trapped and accounted for.

page 27 - Causes for ??ERROR?? - Add Note 9 - All alpha characters (a-z) used as operands and operators in equations must be entered using lower case (non-shifted) characters. Any capital or shifted characters will cause ??ERROR??s in the result of a calculation.

page 31 - Converting DATAFILE record files - A program has been provided on the dfile128 system disk which will convert files created by DATAFILE into file128 format. CONVERT DATAFILE will restructure the number of possible records and will capitalize the first character of EVERY word in the record file. Numbers will be ignored.

FORWARD

Hello, I'm Mike Konshak, the author of dfile128. I'd like to thank you for purchasing this program and to welcome you to our family of users. I hope you will find your money well spent and your needs fulfilled with your ownership of dfile128.

dfile128 is a fully featured easy-to-use database management system. This program was developed by listening to the inputs of several thousand users, like yourself, and implementing those features which most of you found desirable to have. dfile128 has many similarities to, and is patterned after, the DATAFILE database system which I wrote and was originally published by RUN magazine (CW Communications/ Peterborough) for the Commodore C64 computer. dfile128 is a highly evolved program which uses the many capabilities found in the Commodore C128 computer. The additional memory provided by the C128 allows more flexibility and record storage space than was provided by its predecessor, the C64. 80 column displays are possible, with the proper monitor, which increases the ease of use in data entry, allowing more information to be printed on the screen. Since dfile128 stores records in memory for quick access, the new BASIC 7.0 eliminates excessive garbage-collection times that the operating system may invoke. This new BASIC also provides many new powerful commands which make a program like this possible. Sequential files are used to store records and formats on floppy disks, as opposed to relative files. Records and formats must only be loaded into memory once, which minimizes the amount of time spent waiting on the disk drive.

I sincerely believe that you will find this the easiest database program to use and the lowest cost of any database program available while still providing the many capabilities necessary to organize your records. Although most commercial databases will offer an increased number of records possible, because they are disk-based and use relative files, it has been shown that most folks do not have an enormous number of records or desire speed and ease of use over capacity. For those of you who have records created by DATAFILE, I have made dfile128 files compatible with DATAFILE files. It would be unkind to ask you to re-type all that stuff in again.

dfile128 is intended to be easily copied so that you will be able to make backups to insure the integrity and also the access of your records. Please do not give copies to friends. dfile128 is priced so that anyone can afford it. The only way the price can stay so low is if everyone, not just you, shares in the costs.

Thanks,

Mike Konshak

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

TABLE OF CONTENTS

Introduction to dfile128

Database management programs are used to help organize and catagorize large numbers of similar data. A collection of data can be considered similar if all the data in the collection can be described using similar terms. An address list of relatives may be considered similar because each relative, or record, has a Name, Street Address, City, State, Zip Code and Phone Number. These items describing the record are called fields. An inventory list of valuables in the home are considered similar because each item of value (a record) could have the same descriptions (fields) such as Item of Value, Location in Home, Original Cost and Replacement Value.

The nice thing about databases is that you are able to customize a record file to meet your specific needs. There are many programs that specialize in specific applications, but databases allow one to 'write' their own program or application. A record file is created with a structure of a certain number of fields. Each field is given a Title and Length. The length of a field is determined by the number of characters of the record data that will be in that field. For example, let us create a record file called LIBRARY which contains a collection of books. We want to keep track of:

1) The TITLE of the book.
2) The AUTHOR of the book.
3) The TOPIC or subject matter of the book.
4) A brief REVIEW of the text.

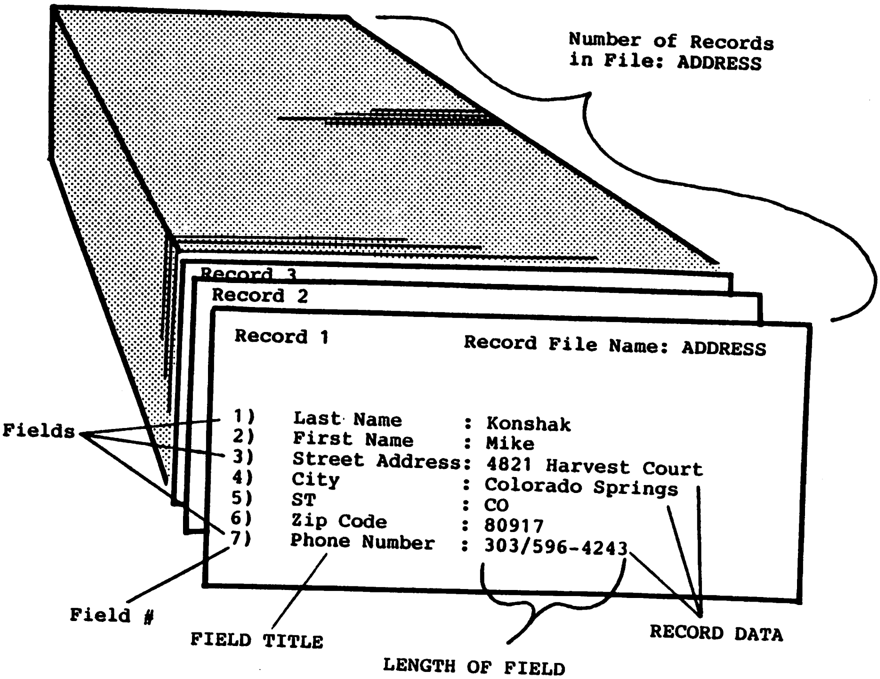
As you can see, the four items above describe any book in our LIBRARY file, so we will create the file with four (4) fields each titled or named TITLE, AUTHOR, TOPIC and REVIEW. It will also be necessary to determine the length of each field. Since dfile128 holds the entire record file in memory, it is important to keep the length of the fields as short as possible. This will provide the greatest number of records possible in the file. You might consider using abbreviations where applicable.

Once a record file is created and the structure defined, you will then need to start adding records to the file. As you build the file with more and more records you will want to write or save the file to your disk for storage. Once this is accomplished you will be able, at a later date, to recover your file by reading or loading it into the computer's memory. Anytime a file is in memory, you may edit it in case you desire to modify or delete a record or even to sort the records in alphabetical order. You can casually browse through your record file by viewing the records one at a time also.

At some time, a hard copy or printout of your file may be necessary. dfile128 provides several options for printing your records. Some options require that you design a format so that the records will be printed in a manner suitable for your application. Mailing Labels, Text Reports and Calculated Reports are some of the options available and, once a format is designed, it will be saved into a special format file on your disk. This way you will only have to design a format once.

Index Card Illustration

A database or record file is very similar to an index card file. Before the computer entered our homes index cards played a major role in organizing our collections, possesions and family affairs. The illustration below shows a record file named address with each record of the file occupying one card.



Record 3

Record 2

Record 1 Record File Name: ADDRESS

```
1)  Last Name      : Konshak
2)  First Name     : Mike
3)  Street Address: 4821 Harvest Court
4)  City           : Colorado Springs
5)  ST             : CO
6)  Zip Code       : 80917
7)  Phone Number   : 303/596-4243
```

EQUIPMENT CONFIGURATIONS

dfile128 configures itself, sometimes with your help, to the type of equipment on your system. 40 column composite color monitors, like the Commodore 1701/1702 are supported as well as 80 column RGB monitors. The 80 column monitors provide the nicest screen displays and provide the greatest amount of room on the screen for displaying record data. Because the C128 computer can only run in FAST mode with an 80 column monitor, using 40 column monitors will develop a noticable time delay during some of the routines. Prompts and data are given in various colors on the screen and audio warnings are provided to flag mispressed keys.

Single or dual disk drives are supported whether they be the 1541 or 1571 disk drives. The 1571 is faster in loading programs than the 1541, but both take the same amount of time in writing or reading sequential record files. Combined dual drives are also considered and should work properly with the system when they become available.

All Commodore or ASCII printers (with suitable interfaces) work with dfile128. Several printer configurations are provided and although there are recommended options for some printers, you may have to try several to find the one that best suits you. Non-Commodore graphic printers provide the best value and capabilities for printing out your reports. Since dfile128 allows up to 136 character wide reports, it would be real handy to be able to put your printer into a compressed character mode. Most Commodore printers do not provide this capability. Multiple up labels are supported (up to 4 across a page) so feel free to purchase any type of labels that fit your printer and pocketbook the best.

It is not necessary to keep your record and format files on a disk separate from the program, although nothing would prevent you from doing so. With a 1541 you should be able to store up to 4 or 5 record files (full) including the dfile128 program on one disk. The 1571 allows a larger number of complete files because both sides of the disk are used. You should make several backups, for your own use, of the program and files to make sure you can always retrieve your important records. Copies of record files used for insurance purposes should be kept in a safe-deposit box and updated every so often. Always write or save your record files at least twice on separate disks after every session involving the updating of records. Keep them in separate locations. There are several methods of copying disks other than by loading and saving programs or files one-by-one. A few methods are provided on the 1541 or 1571 demonstration disks, the 1571 having the most applicable programs for the C128 computer. There are several commercial backup programs for the C64 that copy disks very fast, but it requires that you GO64 with your C128.

RUNNING dfile128

Turn on all your equipment, computer, monitor, disk drive(s) and printer, making sure that you have the 40/80 key set for your monitor type. Insert your dfile128 disk into drive device 8. Type in the following:

```
DLOAD"dfile128    press  <RETURN>
RUN   <RETURN>
```

The screen will clear and display **CLEARING...** followed by the **MAIN MENU.** The current revision of your copy of dfile128 will be shown at the top of the screen. The revision will only be shown the first time that the main menu is displayed.

Notes:
1) Data to be typed in and keys to be pressed will be identified in the manual by **BOLD** printing, **CAPITAL** letters, or letters surrounded by inequality symbols like **RETURN** and **A** or **<RETURN>** and **<A>**.

2) Text and prompts printed on the screen in reversed letters will be identified in the manual by **BOLD** printing or by being surrounded in brackets like **PRESS ANY KEY** or **[PRESS ANY KEY]**.

3) Choices from menus are normally chosen by a single, non-shifted key. Pressing **RETURN** is not necessary. Choosing a key not shown on the menu will cause the computer to beep at you.

4) Routines used for entering formatting information, data for searching records, file names, etc., use **INPUT** statements which will **NOT** allow commas, quotations, semi-colons and colons. The use of the **UP/DOWN** cursor keys will cause screen editing to be dislocated as well as the **CLEAR/HOME** keys. If a flashing cursor is present, your data must be followed by **RETURN** to be accepted.

5) If your program crashes for some reason, caused by bad input, disk or printer I/O failures, etc., you may re-enter the program safely without losing your record data by typing:

```
close3:close4      RETURN
goto 360           RETURN
```

(This should put you back at the main menu.)

6) 40 column screens will be shown in the manual, although the program will automatically adjust for 80 columns if an RGB monitor is present.

7) All menus may be safely exited by pressing **E** which will send you to the previous menu. When entering data, pressing **RETURN** without any entry, or entering a 0 in some cases, will normally allow you to escape a particular routine.

MAIN MENU

```
      [           dfile  Main Menu            ]
           Create new file  Quit program
           Add record to current file
           Modify record in current file
           Delete record in current file
           View file  on screen
           Sort records by field
           Print records
           Read old file from disk
           Write new File to Disk
           @ Disk Drive Commands $
      [     Press the Appropriate Key       ]
         There are 0 records in memory
```

This menu is the heart of dfile and is where you will spend most of your time. It is here at the MAIN MENU where all the entry and editing of records occur and where most operations begin. You have 12 options here, Q, C, A, M, D, V, S, P, R, W, @, and $. Pressing any key not shown will result in a error beep. As with most of the routines and prompts in dfile, the choices are self explanatory, and once you have tried any option, you will most likely not need any assistance from this manual.

QUIT Program

Before a file is created or Read (loaded) into the computer, only a few options are permitted. Quit, Create, Read, or @ and $ are permissable. Pressing Q will terminate the program safely, allowing the opportunity to Write (save) a file, if any changes were made to it. You should always QUIT the program at the MAIN MENU only.

DISK COMMANDS

Pressing @ at the MAIN MENU will invoke the DISK COMMANDS MENU where the following appears:

```
      [         Disk Commands Menu          ]
           Directory $
           Header (New) a blank disk
           Scratch a SEQ file
           Rename a SEQ file
           Change Default Drive# 0 Dev# 8
           Exit to Main Menu
      [    Press the Appropriate Key      ]
```

The disk directory of the disk in the default drive may be displayed on the screen by pressing D or $. The dollar sign, $, found above the 4 key will be seen more often, especially when other prompts require the use of the D key. Most menus which load or save files will give you a directory option. Press E to exit this menu back to the previous one.

HEADER (NEW) A BLANK DISK

Pressing H at the DISK COMMANDS MENU will allow you the option to NEW or header a blank disk. You will be beeped at and warned that you are about to erase a disk. Insert the disk into the default drive and press Y to continue. Enter a 16 character, or less, NAME, followed by a comma, and a two (2) character ID as shown:

```
    [Disk name,ID]? record files,01    <RETURN>
```

SCRATCH A SEQUENTIAL FILE

This routine, entered by pressing S at the DISK COMMANDS MENU, will scratch any sequential file on the disk in the default drive. Insert the disk containing your file to be scratched and enter the file name as prompted. You must enter the name of the file exactly as it appears when viewing the directory. All dfile record and format files are preceeded by two special characters, a right bracket, followed by a space (example: the file mail list appears on the directory as df] mail list). This technique is used to allow related files to have the same name within dfile.

RENAME A SEQUENTIAL FILE

When responding to the prompts for the OLD file name and the NEW file name, be sure to enter the file names as they appear or will appear on the directory. It is advisable to only use lower case (non-shifted) characters in a file name.

CHANGE DEFAULT DRIVE

If you have more than one disk drive, or a dual drive, you may desire to choose one of the drives as your default drive. Anytime dfile accesses the disk drive for loading or saving files or reading the directory, it will always use the default drive. The current drive status is displayed at the DISK COMMANDS MENU. Although most folks call thier drives 8 or 9, these are actually Commodore's device numbers. On single drives the drive number is actually 0, and on dual drives, 0 and 1. You may use a device number from 8 to 11. Once you leave dfile, your computer will still default to drive 0 and device 8.

CREATING A NEW RECORD FILE

Before any records may be entered for manipulation and storage, the computer must have its memory configured for your particular application. When you create a record file, you are defining the structure of your file, within which all your records must conform. You must carefully evaluate the needs of your application so that you will be able to store the minimum required data for each record. It will be very difficult to change your mind later, if indeed, you required more or less fields. This will become more apparant as you start to enter records into the file that you have created. Most of you will not want to re-type in your records if you find your file structure not quite what it should be.

RULES FOR CREATING RECORD FILE STRUCTURES:

1) The number of fields have no restrictions, but it would be advisable to have no more than 15 fields. If you are using a 40 column monitor and your fields are long and many, you may experience scrolling of the screen in the various routines, losing important data and instructions.

2) Field title names cannot contain quotation mark, commas, colons or semicolons. You will receive an ?Extra ignored error if, for example, you put a comma in a title called City, ST.

3) Field lengths cannot be longer than 160 characters.

In order for you to have the largest number of records possible in your file, it is important to keep the number of fields and the length of fields to a minimum. Since dfile is a memory-based data management program, it can only hold so much. 60,000 bytes of memory are allocated for record storage space on the C128 computer. All of your intended file must fit into that space. If your records require more data and it looks like you are going to exceed the space allotted, you might consider creating two or more identical record files.

To keep track of club members, the record file members will have the following structure:

Number of fields: 7

Field	Title	Length
1	Last Name	15
2	First Name	15
3	Street Add	30
4	City ST	22
5	Zip	7
6	Phone	12
7	Date Joined	8

After entering the create routine from the MAIN MENU and entering the above data when prompted, you will be told as to how many records are possible with your chosen structure. In the above example file, members, your choices result in:

```
Your Selections will allow 418 records
Accept, Change Field Data, or Reject
```

If your structure is satisfactory and the number of records possible is sufficient, then press A to accept the structure, continuing on to the MAIN MENU. If you desire to change the title of a field or a length of a field, press C to change the data. You will walk through the selection cycle, changing data as required, pressing RETURN to accept each displayed value. If the whole thing is totally outrageous, press R to reject the mess and start over. For instructional purposes and future examples, let's assume that the above structure was just fine.

ADD RECORDS TO CURRENT FILE

The current file is the one which is located in memory, not on the disk. After creating a record file structure or after reading an old file from the disk, the memory will be configured into hundreds of 'buckets' into which each individual record will reside. By adding records you will be filling each of the buckets.

If you have a 40 column monitor, you will experience some slight delays in the response of the cursor. The FAST mode of the computer is utilized when an 80 column RGB monitor is present and the delays will not be noticed.

When adding (or modifying) records, all text and numeric characters on the keyboard are acceptable, including all punctuation (commas, colons, etc.). Any keypress which the routine cannot handle is rejected by a beep (such as cursor left, up/down and HOME). Beeps will also occur if you try to enter more characters of data into a field than is allowed based on your original structure. The DELETE key must be used to correct your typing errors.

Pressing RETURN without any entry on the first field will exit the record adding process. If the data in a field is repititious, such as the day's date, entering the equal sign (=), will force the current record field to equal or clone the previous record's field data. That way, once a date is entered at the beginning of a day's sitting, you will not have to re-type in that particular field. This is also handy for numerous identical last names or addresses.

If you will be adding a large number of records at one sitting, you would be advised to write (or save) your file from time to time, just in case the power goes out, or the computer locks up for some reason. Every half or full hour would be safe.

MODIFY RECORDS IN CURRENT FILE

Modifying allows you to correct, change, update, alter, etc. your records. When entering this routine you will be asked if you wish to modify a particular record number or ALL of the records in the file. Unless you know the record number or have changes to make on every record, it would be more convienient to modify records starting from the VIEW RECORDS option.

As each record is incremented, each field of that record will be displayed showing the current data or value. If you wish to leave that field as is, just press RETURN to accept the data.

If you desire to erase a particular field you must enter only a Minus sign (-) or hyphen to eliminate the data. The minus sign will not appear when the field data is displayed again.

If you want to clone or equal the previous record's field data, enter an equal sign (=). This will save a considerable number of keystrokes when entering identical and repititious data.

If you wish to accept part of the data and change the rest, use the cursor right key to preprint the data up to the place where you wish to make the change. At that point make your corrections. If you have not changed the position of the characters (changing only one letter for example), the cursor right key will pre-print the rest of the field for you.

At the end of each record, press N to advance to the next record or press E to exit the routine.

DELETING RECORDS IN THE CURRENT FILE

When entering this routine frome the MAIN MENU, you will be asked to you wish to delete a particular record number or ALL the records. If you can not remember the record number, go to to the VIEW RECORDS options and delete the record there.

Although you may have chosen A for ALL the records, you will not actually delete or erase a record from the file without a little more trouble. Each record will be displayed one at a time. If you do NOT want to delete the record, press the SPACE BAR where you will be prompted to advance to the next record or to exit the routine. If you wish to delete the displayed record, press the SHIFT key and the D key simultaneously and the record will be erased.

Once any record has been deleted, the record file will not be in the same order as before. Deletion of records is accomplished by replacing the record to be deleted with the very last record in the file, decrementing the total number of records by one. To put your records back into order, you must use the SORT RECORDS option.

VIEW CURRENT FILE ON SCREEN

This routine offers the user the most flexiblity in viewing, scanning and editing the current file in memory. As each record is displayed you will be given eight (8) prompts as follows:

```
Next    Last    Jump    Find
Modify  Delete  Print   Exit
```

Pressing N advances you to the next higher record number. When you get to the end of the file the screen will start again at the first record. Holding the key down will advance and flash each consecutive record.

L is the inverse of N. L displays the last previous record or next lower record number. When you reach the beginning of the file the screen will display the very last record in the file. Holding the key down displays the file in decrementing order.

Pressing J allows you to jump forward or backward in the file directly. J invokes a prompt which asks for the record number that you wish to go directly to.

F allows you to find records having common data. Two (2) fields may be searched using the ?? operator described on page 16.

M allows you to modify the record shown on the screen. The parameters are the same as descibed in the Modify records option at the MAIN MENU. When you finish modifyinf a particular record it will be displayed revealing the changes made.

D allows deletion of the record from the VIEW option. Again the same rules apply as before. Expect the records to be out of order after deletion.

P is very handy if you have a printer connected to your system. Pressing P will print the individual record shown on the screen providing a snap-shot or hard copy of your record. This is by far the easiest way to get a printout in dfile.

As always E exits you back to the MAIN MENU.

SORT RECORDS BY FIELD

Upon first entering this routine the screen will display the names and numbers of the fields in the current file. You have the capability to sort your file by one (1) or two (2) fields, all in ascending order. The fastest sort will be accomplished by only choosing one field to sort. Two field sorts take considerably longer, depending on the order and number of records (anywhere from a few seconds to a whole hour).

Choose any field number to be your primary sort field, enter 0 to exit the routine. If the file will only be sorted on one field, enter 0 for the value of the secondary sort field. If a two-field sort is desired enter the appropriate field number. If you have a large file this would be a good time for a coffee break.

It is possible that when you sort your file that it will not be in the order that you had expected. Nine times out of ten it will be caused by inconsistant data within the sorted fields, especially where numeric data is concerned. This is because dfile stores all data as strings, not as actual numbers. Consequently the value of each field, when compared for sorting, is detèrmined by the position of each character. For example, one user has twelve disks in his library called Games 1, Games 2, etc. through Games 12. In the following columns, the first column is a list of the disks in the correct order. The second column the list after sorting. The third column has the disks renamed so they will produce the desired sort.

Original Names	Original Sort	New Names and Sort
Games 1	Games 1	Games 01
Games 2	Games 10	Games 02
Games 3	Games 11	Games 03
Games 4	Games 12	Games 04
Games 5	Games 2	Games 05
etc.	etc.	etc.

Note that the number in column three all have two characters, and the first character is a 0. This positions equally the values to be compared during the sort. All numbers should have the same number of characters in order to be sorted correctly. Dates are also bothersome. If you want the date sorted by year, month, then the day, you must enter date data accordingly such as: 84.11.31 , 85-03-25 , or 86/06/01 .

WRITE NEW FILE TO DISK

Your file currently held in memory is considered to be your NEW file. Pressing W at the MAIN MENU will invoke a prompt asking for the name of the file to be saved or written to the disk. If a file was previously read or written, the last file name used will be pre-printed for you on the screen. Just press RETURN to accept the name or make the appropriate adjustments.

A maximum of 12 characters are allowed for a file name in the dfile system. This allows for four (4) reserved characters to help keep related format files separate.

Anytime a record file is written onto a disk where a file with the same name resides, a backup file is created. The following steps describe the process using the sample file members:

1) The previous backup file named df] members .bak is scratched from the disk.

2) The old file on the disk which was named df] members is renamed to df] members.bak by adding .BAK to the end of the name.

3) The current new file is saved under the name df] members.

READ OLD FILE FROM DISK

Anytime a record file is loaded into memory from the disk drive, the memory space allocated to record storage and all variable values are reset to thier initial state, starting with a clean slate, so to speak. What this means is that your current file in memory will be destroyed along with default disk drive and printer values as well as format configurations. You will be forewarned if you have made any changes to your existing and current file and have not saved the updates. The memory will be reconfigured based on the criteria found during the reading of the file.

Do not enter the first four reserved characters as part of the name as it is shown on the directory, just the part of the name to the right of the bracket.

PRINT RECORDS

Pressing P at the MAIN MENU starts an entry into the part of dfile which provides the main product of a database management system, the hardcopy printout. After all, you can not carry your computer around everywhere you go to show others the fruits of your labors. The print functions comprise a major part of dfile and consume most of the available memory space. This is necessary to provide the many powerful and flexible features desired by many users.

Because there are so many printers that will attach to the C128 computer, most are non-Commodore printers which require interface adapters, dfile can be configured to your special system. Upon leaving the MAIN MENU for the first time you will be presented with four (4) printer configurations. After choosing one (such as 4 for printing to the screen), there will be two additional options on the menu. It appears as follows:

```
[   Printer-Interface Configuration  ]
       Current Option is 4
  Press  1  Cardco A, C= 1525
         2  Printers w/ Graphic Interfaces
         3  C= 1526, C= MPS801/802/803
         4  Print Output to Screen
      or Send Printer Commands
         Exit to continue
[   Press the Appropriate Key       ]
```

The current option will change each time you press 1 - 4. Pressing E exits this menu to the PRINTOUT SELECTION MENU where you will choose one of several printing methods.

SENDING PRINTER CODES

Pressing S from the PRINTER-INTERFACE CONFIGURATION menu will deliver you to a routine which will allow you to send ASCII codes to your printer to set it up for various print modes and styles. Most Commodore printer do not give the user much flexibility in the areas of compressed print and various fonts of letter styles. You will have to check your printer manual for possible codes that may be sent. Look for statements that print numbers within CHR$() commands, such as:

PRINT#4,CHR$(15)	which makes a commodore printer print all double-wide or enhanced characters.
PRINT#4,CHR$(27)CHR$(66)CHR$(2)	which invokes a compressed print mode (12 CPI) on Star Printers.
PRINT#4,CHR$(27)CHR$(81)	which invokes 17 CPI pitch on a CITOH Prowriter printer.

Commands that look like PRINT#4,ESC"W"1 must be converted to thier ASCII equivalents such as PRINT#4,CHR$(27)CHR$(87) CHR$(1). When sending printer command codes, only the numbers within the parenthesis are entered, such as 27, 66 and 2. Enter only the numbers one at a time, each number entered should be followed by pressing RETURN. One (1) to four (4) numbers may be sent, the sequence being terminated by entering an asterisk (*) as the last code. The printer will typically do a line feed at this time, which indicates something was sent to it. You can test your change by pressing T at the end of the routine and you may also choose to send another code by pressing A. E will exit you to the PRINTOUT SELECTION MENU.

PRINTOUT SELECTION MENU

Once a printer configuration is chosen, you will be shown the following display immediately after pressing P at the MAIN MENU.

```
[    Printout Selection Menu       ]

 Printing/Format Functions:
     Unformatted List (easy)
     Reports        Formats: rp] *
     Mailing Labels        : ml] *
     Calculated Reports    : cr] *

 Configuration Functions:
     Labels : 5 Rows, 34 Chrs, 1 Up
     Printer: Option 4, Device# 3

     Directory $
     Exit to dfile Main Menu
 [    Press the Appropriate Key     ]
```

This screen allows you to choose the type of printout that will best suit your needs and to re-configure the printer or change label sizes and type. The disk directory may also be displayed from this menu.

UNFORMATTED LISTS

This is by far the easiest printout option because it does not require the user to create or design a format. Records and fields will be printed much in the same way as you might fill out 3 x 5 cards. An unformatted report resembles the following:

```
[ Record # 1 ]--------------------
Name............Mike
Age.............38
Birthday........05/28

[ Record # 2 ]--------------------
Name............Becky
Age.............27
Birthday........06/27

[ Record # 3 ]--------------------
Name............Sarah
Age.............11
Birthday........09/10
```

The name of each field is printed along with the field data and record number. The program will automatically skip over the perforations by calculating how many complete records will fit onto one page. The dashed line is printed across the entire page to be used as a guide for cutting the paper in case you wish to attach the record data to a card. This type of printout uses a considerable amount of paper so it would be best to define a report format for printing large files.

RECORD SELECTION MENU

After choosing an unformatted list option or upon entering a print format into the computer from the disk, you will be given the option to select which records will be in the printout.

```
[       Record Selection Menu        ]
 There are 22 records in file members
      All records in file
      Select individual records by #
      Find Records with Common Fields
      Exit to Print Options Menu
[     Press the Appropriate Key      ]
```

Pressing A will print the entire file using the current format. You will first be asked the starting record #. Record 1 is the default starting position.

Pressing S allows you to print records randomly one at a time. You must already know the record number beforehand of the records that you will want to print.

Pressing F will invoke a routine which will search for records based on your entered data (described in the next section).

Pressing RETURN during printing stops the process after a record. You may continue printing or exit the routine as prompted.

FINDING OR SEARCHING FOR RECORDS

When searching for specific records, you will be required to input four items necessary to perform the search. They are:

Field # — The number of the record field which will be looked at or compared to.

Operator — The logical operator which will determine the type of comparison.

= Field data is equal to search data.
> Field data is greater than search data.
< Field data is less than search data.
<> Field data is not equal to search data.
?? Field data partially matches search data.

Data — The actual search data which will be compared against the field data within the records.

Continue? — Do you wish to continue or expand the search by adding additional search items which will ANDed or ORed to the previous search data. END stops all further search selections.

A good example of using the search process would be to consider a fictitous file containing all the students in a grade school. The file contains all grade levels (1 thru 6), boys and girls (M or F), and the latest test scores (0 thru 5 where an F=0 and an A=5. The structure of the record file is:

Field #	Field Name	Record Data
1	Name	Bill, Mary, etc.
2	Sex	M or F
3	Grade	1 thru 6
4	Score	0 thru 5

Using the above file, we could search for all the boys (M) to be considered for honor roll in grades 5-6 with A's (5) on thier test scores. The search criterium would then be:

Search Field	Operator	Data	Continue?
2 (sex)	=	M	AND
3 (grade)	>	4	AND
4 (score)	=	5	END

If we desired a printout of all the girls whose last name begin with K thru R, the search parameters would be:

Search Field	Operator	Data	Continue?
1 (name)	>	J	AND
1 (name)	<	S	AND
3 (sex)	=	F	END

The operator defined by two question marks (??) operates the same way as the Find option in the VIEW RECORDS mode. If the record field data contains the value of the search data anywhere within the field then a match is considered good. Example: If the search data is Brown then the following records would be found; Brown, Browne, Browning and McBrown.

PRINT FORMAT OPTIONS MENU

Whenever you choose... M for Mailing Labels
... R for Reports
... C for Calculated Reports

...at the PRINTOUT SELECTION MENU, you will be confronted with the following screen:

```
[        Print Format Options       ]
 [Current Format Type] Labels
 [Format File Name   ] No Formats Present

      Create New Print Format
      Print With Current Format
      Load Old Print Format
      Save Current Print Format
      Directory $
      Exit to Printout Selection Menu
[      Press the Appropriate Key      ]
```

If you have entered this routine for the first time and a print format is not present then two of the options, Print and Save, will not be displayed.

If you have a pre-existing format on the disk then it may be loaded by first pressing L and then entering the format file name as prompted.

If you have never created a print format for the particular type of printout or desire to change one that is already loaded you must press C to enter the respective format definition routines. Designing or creating formats will be discussed in subsequent sections.

Once a print format is present in memory, pressing P will advance you to the RECORD SELECTION MENU to begin the print process. This menu was discussed earlier.

CREATING MAILING LABEL FORMATS

Before records from your file can be printed out on labels, you must first define a label format specifically designed for your file. Formats tell the computer which of your record fields to print in each row of the label. Besides the actual data that is to be printed you must also define the size of your label.

The size of a label is specified by the number of rows and the number of characters in each row. dfile defaults to a standard label size which measures 15/16 of an inch wide and 3 inches long. This popular label size can be printed with 5 rows and 34 characters. There are various sizes of labels available and also various types. By types I mean the number of labels across a page, sometimes refered to as one up, two up, etc. dfile will print up to 30 rows and 136 characters/row on labels up to 4 across a page (4 up). Obviously, you cannot have every parameter mentioned, but you should be able to design a format around any size or type of label. I prefer using one up labels, but some printers do not have adjustable tractors so you must print the labels two (or more) up.

After defining the number of rows you will be asked for which fields will be printed in each row. Up to three record fields may be printed in each row and in any order. You will be required to enter at least one record field number for the first field in each row that will print data. If you desire to skip a row (print a blank line) you must enter a minus sign (-) instead of a number in all three fields for that row.

When asked for field 1-3 in each row enter the record field number behind the prompt. A zero will be preprinted for you the first time. If you only want one record field in a row, enter the desired record field number for field 1 and zeros (0) for field 2 and field 3. For example, let us use the address record file called members for creating a label format. members is a good candidate since it needs to combine multiple fields in several rows. members has the following structure:

Field	Title	Length
1	Last Name	15
2	First Name	15
3	Street Add	30
4	City ST	22
5	Zip	7
6	Phone	12
7	Date Joined	8

Knowing the record file structure, we will put, in 5 rows, the Date Joined in row 1 so the member will know when to pay his dues, skip row 2, First Name and Last Name in row 3, Street Add in row 4, and City ST and Zip in row 5. The rows would look like the following:

Row	Field 1	Field 2	Field 3	Will print
1	7	0	0	Date Joined
2	-	-	-	
3	2	1	0	First + Last Name
4	3	0	0	Street Address
5	4	5	0	City ST + Zip

As you can see, it is not important which position the record fields are in, because you can rearrange them when you create the label format.

Again let me remind you that the default label size is 5 rows, 34 characters long and one up. If you are using a different size or type of label, you must alter the label configuration by pressing L at the PRINTOUT SELECTION MENU.

Label formats are handy if you wish to review all your records on the screen. Use PRINTER OPTION 4 and one of the multiple up options and change the number of characters to get as many records across the screen as possible.

When you finish defining your label format you will be given the opportunity to review or change the format. If you pressed N then you will be asked to SAVE the label format. You should immediately save your label format in case you might forget later. Press Y and you will be prompted with the current record file name. Either press RETURN to accept the file name as is or make the appropriate changes. Use no more than 12 characters for your label format file name. Any label formats with the same name will be scratched and replaced with the new format file. Label format files appear on the disk as ml] members.

CREATING REPORT FORMATS

Just like labels, a report format must be designed around your record file, if you wish to print the record data in an organized and well presented columnar report. Reports differ from labels in that the record field data is printed in columns instead of rows. This report format routine is reccomended for reports containing mostly text although the last column may be used for totalling numbers. Use Calculated Reports for reports that require a lot of calculations. In every report, you have the capabilities, and must define, the following parameters:

The WIDTH, in CHARACTERS of the report. Up to 136 characters may be printed across the page depending on the capabilities of your printer. You might have to put your printer into a compressed print mode to print more than 80 characters across the page.

The NUMBER of TITLE LINES at the top of the page. Up to four (4) titles may be printed. The report titles will only be printed on the first page.

The TITLE for each TITLE LINE. Titles can be as long as the width of the report. Titles cannot contain commas, colons or semicolons. Examples of titles would be Company Name, Date, Report Information, Sources, etc.

The NUMBER of COLUMNS in the report. Up to 16 columns may be chosen. At least one column must be printed.

The HEADER for each COLUMN. Header names appear at the top of every column and are printed at the top of every page. Headers identify the contents of the column and may be any name you choose. Header names cannot be wider than their respective columns nor can they contain commas, colons or semicolons.

The POSITION of each COLUMN. The starting location of each column must be chosen and will be a number from 1 to the width of the report. You will have to determine beforehand how wide that you expect each column will be based on the length of the record field data that will be printed in the column. Recommended starting points are preprinted on the screen for you and are calculated by evenly dividing the width of the report by the number of columns. For example, if a report has a width of 80 characters and 4 columns are chosen, the starting positions for each column will be 1, 21, 41 AND 61. Each column will actually be 19 characters wide and separated by 2 blank spaces. You should change the default values to match the actual column width needed to properly print your record field data (do not forget about the two spaces between columns). You may have to go back and adjust the width of the report and the position of the columns after you see your first printout in order to properly fit your data. A sheet of graph paper comes in very handy in determining the width and position of columns.

The CONTENTS of each COLUMN. Up to three (3) record fields may be printed in each column, similiar to the way fields are put in rows when formatting a label. Each field within a column will be separated with one space and the entire contents of that column will be printed left justified. At least one record field must be chosen for each column and must be entered in the first field position. Enter zeros (0) for the extra fields in the columns if you do not wish to print three record fields. The record field titles will be printed on the screen for easy reference. Only one record can be printed in each row of a report.

TOTAL the LAST COLUMN? The only calculation that the report format will perform will be to total or sum the contents in the last column of the report. The record field data defined for the last column must be numeric and must be entered into the first field position. If non-numeric data is present (like the dollar sign) .at the beginning of the record field the program will produce a result of zero (0) for that record. The total of the column will be printed at the bottom of the report. The values of the column will be printed right justified with two (2) decimal places, such as 125.50. To choose the totalling option, choose 1 when prompted at the end of the formating process. 0 turns off totalling. If you desire more complex calculations, such as multiplication of record fields and columns within a row, use the Calculated Reports format routines.

At the end of the report format definition process you will be given the opportunity to go back and review or change the format parameters. If you respond with a N then you will be asked to SAVE the format. If you have made any changes at all to an existing format or have created a new format press Y at this time. Use the preprinted file name or type in a new name. Use a maximum of 12 characters in the name of the report format file. If any report format files reside on the disk with the same name you have chosen the file on the disk will be scratched and the new file will take its place.

To use the file members as an example, you might create a report format using the following parameters:

```
REPORT FORMAT FILE: members  (appears on the disk as rp] members)
NUMBER OF CHARACTERS (wide): 80
NUMBER OF TITLE LINES: 2
TITLE 1: Club Member Address List
TITLE 2: May 31 1985
NUMBER OF COLUMNS: 5
CLM 1: POSITION= 1   HEADER= First/Last Name  FIELDS= 2 1 0
CLM 2: POSITION= 19  HEADER= Street Address   FIELDS= 3 0 0
CLM 3: POSITION= 41  HEADER= City and State   FIELDS= 4 5 0
CLM 4: POSITION= 60  HEADER= Phone Number     FIELDS= 6 0 0
CLM 5: POSITION= 74  HEADER= Joined           FIELDS= 7 0 0
TOTAL LAST CLM: 0
```

Enter the above values after the respective prompts. You will notice that after you have entered a value it will be preprinted for you when you go back to review the format parameters.

CREATING CALCULATED REPORT FORMATS

dfile provides the capability of using your record file data to produce 'spreadsheet like' reports. Many applications are possible when using this option because dfile allows considerable flexibilty in designing and creating your calculated reports. Calculations are performed on only one record at a time within a row of the report, however the data and results in each column of the row may also be used in calculations. The entire contents of columns which include all the record data within a column may be totalled (summed) or averaged. Various methods of justification are provided as well as many mathematical tokens to expand the capabilities of your printed document. The following is a more comprehensive description of the capabilities provided using calculated reports:

1) A HEADER consisting of up to four (4) TITLE LINES may be printed at the top of the first page of your report.

2) Up to sixteen (16) COLUMNS may be defined in your report. The number of columns possible will be dependent on the width of each column (determined by the column position), and the number of characters possible across the page provided by your printer.

3) Columns may contain:
   a) The RECORD NUMBER.
   b) The contents of a FIELD or data within a record.
   c) The contents of another COLUMN within the current row.
   d) An EQUATION, the results of which will be printed within the column.
   e) A RUNNING TOTAL of the data found in a previous column.

4) Equations will perform operations using the following OPERANDS:
   a) Numerical data found as the contents of a FIELD within a record.
   b) Numerical data found as the contents of a previous COLUMN within the row.
   c) Numerical CONSTANTS or values such as PI, 100, 3.14159, 469.65, 2.6769E-3, 8.965E10, etc.

5) Equations may perform operations using one of 47 different OPERATORS such as +, -, *, /, ^, sin(), log(), sqr(), etc. Included in the list are conversion OPERATORS which will perform unit conversions such as inches to millimeters, mms(), and degrees to radians, rad().

6) End-of-Column operations or EOC may be defined as follows:
   a) No Operation or NOOP which does not print any values at the end of a column.
   b) Calculate the TOTAL or SUM of all the data printed in the column.
   c) Calculate the AVERAGE of all the data printed in the column (the total divided by the number of records printed).

7) Column JUSTIFICATIONs may take the following forms:
   a) Left (for alphanumeric data).
   b) Right (for alphanumeric data).
   c) Currency (numeric data, right justified, 2 decimal places).
   d) Decimal (numeric data, right justified, 0 to 8 places).
   e) Percent (multiplies contents of column by 100).

EXAMPLES:

```
Justification                       Numbers        Text
------------------------------   ------------   ------------
l - Left                         [123        ]  [ABC        ]
r - Right                        [        123]  [        ABC]
c - Currency, 2 places fixed     [     123.00]  [        .00]
d,4 - Decimal, 0 - 8 places      [   123.0000]  [      .0000]
p,1 - Percentage, value * 100    [      123.0]  [         .0]
```

Note that text printed with numeric justifications produce a result of 0.

Now that you are aware of the reporting capabilities, let us demonstrate the report format of three report examples:

REPORT EXAMPLE 1

```
                      Class Record
                Miss Appleby's Third Grade
                  Main Street Elementary
      ---------------------------------------------------
      NAME           1QTR  2QTR  3QTR  TOTAL   AVRG  %ABSNT
      ---------------------------------------------------
      Albrecht J     4.0   3.0   4.0   11.0    3.67    0.0
      Baker M        2.0   3.0   2.0    7.0    2.33    1.5
      Donnely S      1.0   4.0   4.0    9.0    3.00    2.0
      Gray A         4.0   4.0   4.0   12.0    4.0     0.0
      ---------------------------------------------------
                     AVG   AVG   AVG           AVG     AVG
      ---------------------------------------------------
                     2.7   3.5   3.5           3.25     .9
      ---------------------------------------------------
```

REPORT EXAMPLE 2

```
                   BOB's LIGHT BULB SUPPLY
                Current Inventory and Assets
      ----------------------------------------------------
        #  BULB   QTY   COST  PRICE  TTLCST   TTLRET  %MUP
      ----------------------------------------------------
        1  15W     75    .22    .44   16.50    33.00   100
        2  25W    112    .28    .49   31.36    54.88    75
        3  40W    125    .32    .66   40.00    82.50   106
        4  60W    256    .40    .75  102.00   192.00    88
        5  100W   178    .49    .98   87.22   174.44   100
      ----------------------------------------------------
                         AVG    AVG   TOTAL    TOTAL   AVG
      ----------------------------------------------------
                         .34    .66  277.48   536.82    94
      ----------------------------------------------------
```

REPORT EXAMPLE 3

```
                    Trigonometric Table
            ---------------------------------------
            Deg  Radians      Sin     Cosin Tangent
            ---------------------------------------
             15    .2618     .2588    .9659   .2679
             30    .5236     .5       .8660   .5774
             45    .7854     .7071    .7071   .9999
             60   1.0472     .8660    .5     1.7320
             75   1.3090     .9659    .2588  3.7320
            ---------------------------------------
```

In the first REPORT EXAMPLE 1, Miss Appleby used a file which contained her class records and had the following fields in each of her records:

```
    Record file: CLASS RECORD  (4 records shown)

    Field # 1) Student
            2) Days Absent
            3) QTR 1 Grade
            4) QTR 2 Grade
            5) QTR 3 Grade
            6) ...
            7) ...
```

Miss Appleby has other fields in her file which will not be used in this report. She wants to produce a report which will calculate the overall grade-point-average of each student, and the entire class, as well as to print out the percentage of absenteeism. She designed a 52 character wide report with 7 columns which printed field data from her records in columns 1, 2, 3, and 4. Columns 5, 6, and 7 contain equations that calculate the data printed in the previous fields and columns. The following is Miss Appleby's report structure (her titles and header are shown on the report example):

Col	Pos	Contents	Field/Equation	Just	EOC
1	1	Field Data	f1	L	NOOP
2	14	Field Data	f3	D,1	AVG
3	20	Field Data	f4	D,1	AVG
4	26	Field Data	f5	D,1	AVG
5	32	Equation	c2++c4	D,1	NOOP
6	40	Equation	c5/3	D,2	AVG
7	46	Equation	f2/180	P,1	AVG

As you can see, column 1 contained data from field 1 (f1) of the record file and is printed LEFT justified with no end-of-column (EOC) calculations. Columns 2, 3 and 4 also contained record data but were justified with one (1) DECIMAL place. The EOC of those columns were to AVERAGE the values found in each column. The equation in column 5 (c5) summed the values printed in columns c2 thru c4. The ++ operator means SUM THRU and will add or total all the data printed in a row from the first column defined (c2) thru the second column defined (c4). The equation could have been written c2+c3+c4 or f3+f4+f5. Column 6 (c6) calculates the average grades by dividing c5 by the number of quarters (3) and is justified with two (2) decimal places. Finally, column 7 (c7) divided the number of days absent found in field 2 (f2) of the record file (not printed on the report) by the number of school days (180) in the year. Column 7 (c7) was justified using PERCENT and one decimal place. When PERCENT is chosen it will automatically multiply the value to be placed in a column by 100 to obtain the the normal form of percentage.

Confusing? Maybe at first, but you will understand the concepts more as you try reports of your own. let us look at another example.

In REPORT EXAMPLE 2, Bob desired to have a report which calculated the current total value of his stock of light bulbs. He also wanted to calculate the value based on his retail price and to determine the percent of markup from cost to retail. Bob's record file and report structure took the following forms:

Record File: INVENTORY (5 records shown)

Field# 1) Bulb Type
2) Quantity In Stock
3) Cost Each
4) Retail Price

Col	Pos	Contents	Field/Equation	Just	EOC
1	1	Record #	#	R	NOOP
2	6	Field Data	f1	L	NOOP
3	12	Field Data	f2	R	NOOP
4	18	Field Data	f3	C	AVG
5	25	Field Data	f4	C	AVG
6	32	Equation	f3*f2	C	TOTAL
7	41	Equation	f4*f2	C	TOTAL
8	50	Equation	(c7-c6)/c6	P,0	AVG

Bob started his 54 character wide, 8 column report by printing the RECORD NUMBER in column 1 (c1). He did this by entering a pound sign (#) for the contents of the column. The program will automatically print the record number for Bob. Field data from the record file were printed in columns 2 thru 5 and equations were defined for columns 6 thru 8. Columns 4 thru 7 were CURRENCY justified which defaults to two (2) decimal places. Columns 6 and 7 mutliplied thier respective cost (f3) and retail price (f4) by the quantity (f2) of bulbs in stock and had thier EOC operation as TOTAL. The last column (c8) subtracted the total cost (c6) of each of the light bulbs from the total retail price (c7), then divided that RESULT by the total cost (c6) to obtain the percent of markup. Column 8 is justified with percent and 0 decimal places.

You will notice in all the examples shown that the WIDTH of each column is determined by the character position across the page which defines the STARTING POSITION of the column. Each column is separated by two (2) characters. If you subtract the starting position of the column from the starting position of the next column minus 2, you will get the width of the column. Calculate the widths by using the following formulas:

Col N Width = (Col N+1 Start Pos) - (Col N Start Pos) - 2
or Last Col Width = (Report Width) - (Last Col Pos)

Any data or calculation result which is longer than the width of a column will be truncated and some characters of the data will be lost. Choose your column width based on the longest data necessaryb to print or in the case on numeric data, the largest number of digits (as found after totalling the column). Make sure you provide a space for the + or - sign when printing numbers even though the sign may not be printed. Take particular care to leave enough space for record numbers.

In our final example, REPORT EXAMPLE 3, a simple trigonometric table was generated in order to demonstrate some of the more advanced math operations and conversions. The record file and report structure appears as follows:

Record File: DEGREES (5 records generated)

Field# 1) Angle

Col	Pos	Contents	Field/Equation	Just	EOC
1	1	Field Data	f1	R	NOOP
2	6	Equation	. rad f1	D,4	NOOP
3	15	Equation	. sin c2	D,4	NOOP
4	25	Equation	. cos c2	D,4	NOOP
5	34	Equation	. tan c2	D,4	NOOP

The record file was created with only one field which contains a selection of common angles which will be used to 'SEED' the calculations. Column 2 (c2) takes the angle found in the field (f1) and converts it from DEGREES to RADIANS. Columns c3 thru c5 then use this value found in c2 to calculate thier respective trig' functions. Column c2 thru c5 are justified with 4 decimal places and none of the columns have any EOC operations performed.

You will notice on the report example that no total lines were printed at the bottom of the report. This is because NOOP was chosen for ALL of the columns.

It is necessary, when performing calculations involving angles with a computer, to convert the angles into RADIANS instead of DEGREES. You will be given unexpected results if you fail to make the conversion. The BASIC interpreter was designed to use RADIANS since the math processor handles radians easier than degrees.

I have attempted to make conversions like the above as painless as possible (and to cut down on the number of equations) by inventing a new set of MATH TOKENS that specifically perform conversions. Tokens are the operators that perform functions in your BASIC computer, such as LIST, RUN, GOTO, IF-THEN, INT(), CHR$(), etc. What I have done is given you a few more math related tokens to use when writing your equations. Besides shortening your formulas they will also give you more power in calculating your record data. The tokens and other operators and thier functions are listed in the TABLE OF LEGAL OPERATORS. You might glance over the various operators before you define your report format to help give you a better idea of what can be accomplished in your report.

The next few pages list the LEGAL parameters and definitions for OPERATORS and OPERANDS. They are rules you must follow when defining equations. You should try writing simple equations at first until you get the feel of how the routines work before starting more complex or multiple operator equations.

RULES FOR LEGAL OPERANDS

OPERANDS are the variables (data) being operated on (or calculated) by OPERATORS such as +, -, sqr, int, etc. Operands are denoted in the TABLE OF LEGAL OPERATORS as X and Y. The value created by a calculation is called the RESULTANT. To help clarify and define these terms we will use the following simple equations as examples:

A= B*C   Z= X+Y   res= .sinX   res= X^Y   res= res/200

In the above equations, anything to the LEFT of the equal sign (=) is the RESULTANT of the equation. The variable A, Z and res are therefore resultants. In all the equations that you will define using dfile, res will be your resultant.

The items to the RIGHT of the equal sign will always consist of two (2) OPERANDS and one (1) OPERATOR. The middle term being the OPERATOR. B, X, Y, res, 200 and the period (.) are all OPERANDS. *, +, sin, ^, and / are all OPERATORS. The following is an expanded and more detailed description of what may be used as LEGAL OPERANDS:

RECORD FIELDS: f1, f4, f12, etc.

An equation using record fields may be used in any of the following forms: res=

f2*f3 f4^f2 f4^3 100*f5 c3+f3 .absf1 .sinf4 res*f2 f5+f5

The following conditions must apply:

1) The number of the field defined must exist in the record file being printed.
2) The same field may be used more than once in any one column or in a number of column equations in either operand position.
3) Fields may be used with any operator in an equation except the SUM THRU (++) operator.
4) The field referenced must contain numeric data. Alpha characters (A-Z, $, %) present in the data will produce a result of 0, an unreliable answer or result, or an ??error?? condition when used in a calculation. ??error?? will be printed as a result if the equation cannot be resolved or may cause the program to crash.

COLUMNS: c2, c8, c16, etc.

An equation using columns may take the following forms: res=

c4*c5 c6+f5 c3^4 c2/res .cosc2 .invc4 c1+c1 .psic12

The following conditions must apply:

1) The column number referenced must be less than the number of the column containing the equation. If, for example, you are defining an equation for column c6 you cannot use c6, c7, c15, etc. as operands.
2) Columns may be used in either operand position with any operator.

CONSTANTS: PI, 100.50, 1.234e+6, -45.5, etc.

An equation using constants may assume any of the following: res=

```
2+2  f4*250  32^c3  180/pi  res*256  .sin-30  -12*res  4^2

1.2345e+6*f1  c12/3.14159e-3  -5.66666e+4*-7.33333e-2
```

The following rules must apply:

1) Constants must be within a range operable to the computer.
2) Constants may occupy either operand position.
3) Constants may have a positive (+) or negative (-) value.
4) PI (3.14159265) is the only predefined constant.

OTHER OPERANDS: . and res

The period (.) is used as the first operand when only one operand is being operated on, normally in the case when a higher level operator is being used (higher level operators are always identified with a three character code), as in: res=

```
.sin30  .invf1  .deg45  .sqr4  .cot90  .absc12  .kms1.234e+3
```

The resultant (res) of a previous calculation within the current column must be used as an operand if more than one operator is used. Up to three (3) operators or equations may be used to complete a calculation within a column. The first operator may use any two legal operands, but the second and third operators must have res as one of the operands in either position.

Case in point. Given a mathematical equation written normally:

Z= ((X+Y)*100)/1000

In the above equation, Z is the resultant and X, Y, 100 and 1000 are operands. There are three operators, +, * and /. The equation would have to be impimented using dfile by the following operations:

1st operation:	res=X+Y	X+Y
2nd operation:	res=res*100	(X+Y)*100
3rd operation:	res=res/100	((X+Y)*100)/1000

As you can see, you must follow the standard rule for parenthesis, where you must start at the innermost set and work your way out.

The variables X and Y will be used to represent the first and second operand positions in the following TABLE OF LEGAL OPERATORS. It will always be assumed that res= will preceed each function example.

TABLE OF LEGAL OPERATORS

Op	Func	Description of Operation
++	cX++cY	SUM the columns X thru Y.
+	X+Y	ADD the values X and Y.
-	X-Y	SUBTRACT the value Y from X.
*	X*Y	MULTIPLY the values X and Y.
/	X/Y	DIVIDE the value X by Y.
^	X^Y	Raise the value X to the EXPONENT Y. Y may be +, - or a fraction.
sgn	.sgnX	Changes the SIGN of X from + to - or - to +.
sqr	.sqrX	Returns the SQUARE ROOT of X.
inv	.invX	Returns the RECIPROCAL of X (1/X).
abs	.absX	Returns the ABSOLUTE VALUE of X.
int	.intX	Makes a fractional X into an INTEGER.
log	.logX	Returns the NATURAL LOG of X to the base e. To convert to log base 10, divide by .log10
exp	.expX	Returns the value of the mathematical constant e (2.71827183) raised to the power of X.
sin	.sinX	Returns the SIN of X, where X is an angle in RADIANS.
cos	.cosX	Returns the COSINE of X.
tan	.tanX	Returns the TANGENT of X.
sec	.secX	Returns the SECANT of X (or 1/sin).
csc	.cscX	Returns the COSECANT of X (or 1/cos).
cot	.cotX	Returns the COTANGENT of X (or 1/tan).
atn	.atnX	Returns the ARCTANGENT, or angle in radians, whose tangent is X (inverse tangent).
aco	.acoX	Returns the ARCCOSINE, or angle in radians, whose cosine is X (inverse cosine).
asn	.asnX	Returns the ARCSIN, or angle in radians, whose sin is X (inverse sin).
asc	.ascX	Returns the ARCSECANT of X (inverse secant).
acs	.acsX	Returns the ARCCOSECANT of X (inverse cosecant).
act	.actX	Returns the ARCCOTANGENT of X (inverse cotangent).
rad	.radX	Converts the angle X, in degrees, to RADIANS.
deg	.degX	Converts the angle X, in radians, to DEGREES.
mms	.mmsX	Converts the distance X, in inches, to MILLIMETERS.
ins	.insX	Converts the distance X, in millimeters, to INCHES.
kms	.kmsX	Converts the distance X, in miles, to KILOMETERS.
mls	.mlsX	Converts the distance X, in kilometers, to MILES.
dfr	.dfrX	Converts the temperature X, in degrees centigrade (celsius), to degrees FAHRENHIET.
dcg	.dcgX	Converts the temperature X, in degrees fahrenhiet, to degrees CENTIGRADE (celsius).
gms	.gmsX	Converts the weight X, in ounces, to GRAMS.
ozs	.ozsX	Converts the weight X, in grams, to OUNCES.
kgm	.kgmX	Converts the mass X, in pounds, to KILOGRAMS.
lbm	.lbmX	Converts the mass X, in kilograms, to pounds.
nwf	.nwfX	Converts the force X, in pounds-force, to NEWTONS (1 Newton = .102 kg).
lbf	.lbfX	Converts the force X, in newtons, to POUNDS-FORCE.
mps	.mpsX	Converts the velocity X, in feet/sec, to METERS/SECOND.
fps	.fpsX	Converts the velocity X, in meters/sec, to FEET/SECOND.
nsm	.nsmX	Converts the pressure X, in pounds per square inch, to NEWTONS PER SQUARE METER.
psi	.psiX	Converts the pressure X, in newtons per square meter, to POUNDS PER SQUARE INCH.

TABLE OF LEGAL OPERATORS (continued)

Op	Func	Description of Operation
=	X=Y	If X equals Y then True (res=1), else False (res=0).
<	X<Y	If X is less than Y then True.
>	X>Y	If X is greater than Y then True.
<>	X<>Y	If X is not equal to Y then True.
or	XorY	If either X or Y are True (greater than 0) then True.
and	XandY	If both X and Y are True (greater than 0) then True.

CAUSES FOR ??ERROR??

When dfile is performing calculations, the program will try to catch as many mathematical errors as possible. Some errors caused by calculations may force the computer to crash while others may just produce inaccurate results. Syntactical errors (which cause the premature termination of the program) have, to the best of my knowledge, been accounted for).

The following produce an error condition, resulting in a value of 0 for the particular row/column cell, or an ??error?? flag.

1) Dividing by zero, X/0.
2) Taking the root or square root of a negative number, sqr(-X).
3) Taking the TAN, SEC, CSC and COT of zero, tan(0).
4) Taking the natural log of a number less than or equal to zero, log(0) or log(-X). Also: ASN, ACO, ASC and ACS.
5) Taking the mathematical constant e to a power (exponent) greater than 87, exp(87).
6) Using an invalid or non-legal operator or operand.
7) Using an operator where an operand should be in an equation (and vice-versa).
8) Using a numerical constant outside the range of the computer.
   a) Integers (whole numbers without decimal places) must be within the range of -32768 to +32767.
   b) Floating-point numbers, in scientific notation, must be within the range of +2.93873588e-39 to +1.70141183e+38.

Note: A calculation that tries to produce a result outside the legal range of the computer will result in an ?OVERFLOW ERROR and will definitely crash the program. There is not a simple way to test for this, so stay away from galactic calculations. Since dfile uses a buffer for each column to keep track of column totals and values for calculations within a row, it is possible that your field data may inadvertantly cause an unexpected ?OVERFLOW ERROR. For example, one user used a calculated report to print a list of items which included serial numbers. One of his serial numbers was 1e800015. This number, though it was not used in a calculation, was read by the buffers as a floating point number in scientific notation (1e800015 is the same a 1 followed by 800,015 zero's. This is a LARGE number!). To help prevent what is intended to be printed as TEXT to be interpreted as a number, you should preceed suspect record field items with an alpha character, such as # or P. This will always return a value of 0 since numbers after alpha characters are ignored (ie., 1234567abc=1234567, abc1234567=0, 3e6=3000000, #1e38=0, etc.).

GENERAL NOTES

Converting DATAFILE record files to dfile128 format.

Records are held in arrays which must be properly dimensioned whenever a file is read or loaded into memory. The C128 computer allows twice as many records as the C64 and consequently structures the arrays slightly different than its predesessor. Those of you who have used DATAFILE, by Mike Konshak (c)1983, to store records may want to use the more powerful dfile128 to manipulate an existing file developed with DATAFILE. Although the files are compatible, the size of the record file will still be limited to the number of records determined by DATAFILE. This may be corrected by performing the following steps:

1) Using a C64 or a C128, load and run your DATAFILE program and, at the MAIN MENU, read the DATAFILE record file into memory. The number of records possible will be shown at the bottom of the screen.

2) Press the RUN/STOP key. The screen should display BREAK IN 30.

3) Type, followed by RETURN:

```
R=R*2     <RETURN>
CONT      <RETURN>
```

4) Press W and enter the file name to save (write) the file back on the disk. You might consider a new file name or a different disk to keep your C128 and C64 files separated.

You have just doubled the capacity by making the computer think it can hold more. Press W to write or save the file back to the disk. Once saved, with the variable R changed, the file will dimension itself correctly when loaded with dfile128. DO NOT try to load this file back into your C64 with DATAFILE!

WHAT ELSE?

Other databases using relative files, even though converted to sequential files, are not compatable with dfile. A utility program may be developed, along with several others, that may provide the necessary conversions and enhancements. All registered owners of dfile128 will be notified of any new programs that will further enhance the capabilities of your files.

A UTILITY disk will be made available that will supplement the capabilities of dfile128 in manipulating your files. The programs on the utility disk will allow more flexibity to using your files toward creating new files, mergeing files, exporting files, and printing files.

A TEMPLATE disk will also be available which will contain numerous databases (record files) in popular configurations useful to most everyone. Besides the record files being created for you, print format files will also reside on the disk which will assist the novice and expert alike in providing applications easy to impliment.

NEW REVISIONS

As any program matures and is used for many and varied applications, invariably the public will request more capabilities and enhancements, not already provided by the program. dfile128 is no exception. As mentioned in the forward, page 1, this program was developed by LISTENING to the users who have exercised the routines within the program in ways unexpected by the author.

You may discover that your neighbor has a revision different than yours which may have an enhancement to your liking and is not present in your version. I would apologize but for the fact that dfile would not be the program it is today without it having to undergo many evolutionary cycles. At the time you received your copy, it was the best one available. Tomorrow, who knows?

You may obtain the most current version of dfile128 by returning your original disk (or serialized label) along with $8.00 U.S to the following address:

michaelsoft
4821 Harvest Court
Colorado Springs, CO 80917

Suggestions and comments are always welcome. If you desire a reply, please send along a self-addressed stamped envelope. I will try to answer as soon as possible, especially if there is a problem not covered by this manual. Emergencies may be directed to the author from 6:30 PM to 10:00 PM mountain standard time by calling this number:

303-596-4243

REFUNDS and WARRANTIES

Because dfile128 is a low cost and high quality program that is intentionally made easy to copy (backup for personal uses), there are no refunds given to purchasers. Any complaints and problems will be handled as sincerely as possible on an individual basis.

There are no warranties, expressed or implied to this software and the author and agents cannot be held liable for data lost or damage to users equipment.

COPYRIGHTS

Yes, this program is copyrighted and is not given unto the public domain. However, because it is easily copied, it is very tempting to share your program with others. Aside from the fact that you would be cheating yourself, it would also be illegal. However, the FBI will not bust down the door if your friend gets a copy. I would ask, though, if your friends are interested in dfile128, that you go ahead and let them try it, on the condition that if they decide to use dfile128 that they promise, to you, to send for their own copy. I will leave the reliance upon the sincerity and honesty of your friends up to you. I, my wife Becky, and our children thank you for this consideration. Mike Konshak.

NEW REVISIONS AND INSTRUCTIONS TO DFILE128

Hi, folks! I was able to to find some free time that allowed me to go in and make some changes to the 128 programs. Many of these changes were suggested by some of you, and others were simply improvements that made the programs more reliable and 'bullet proof'. All-in-all, as I describe some of the features I am sure that you will realize that the enhancements will be well worth the $8.00 that Becky and I charge for updated disks. This announcement is also the ERRATA sheet for DFILE128 which you should keep with your manuals because we will not be sending additional documentation with the disks. WORDFILE (2.0) ERRATAs are found in document files on the WORDFILE disk, so additional errata sheets are never provided since you can just print them out. WORDFILE has been changed to be compatible with more printers, and has also been corrected for some hidden flaws.

The DFILE UTILITY PROGRAMS, which were previously sold as a separate product, have been combined on the same disk as the DFILE DBMS program. This was part of our overall price reduction policies. Therefore, when you send us your $8.00 you will automatically receive the most current UTILITY programs on that disk. Those of you who did not purchase the UTILITY programs before may obtain the UTILITY manual for an additional $5.00.

DFILE128 VERSION 2.1C

- I have increased the storage capacity in memory by allowing the number of records to increase if you have not filled up the fields in previous records. The number of records possible will fluctuate from time to time as additional records are added. In fact you may also experience a negative number being displayed as you reach the end of your file. The total number of records will vary from previous revisions, but this new method allows loading MERGED files on the disk that are small enough to be put into memory.

- Added a new INPUT routine to all prompts and data entry that prevents you from accidentally cursoring away from where you are supposed to be and allows the use of the ESCAPE key to exit routines and operations (not implemented in the DFILE UTILITY PROGRAMS). Just press [ESC] if you get in trouble.

- The programs save the current configuration in a special file which remembers what printer option you are using, disk drives, and programmable function key data settings. The current settings are saved every time you save (write) your record file to the disk and loaded every time you load a file. Only one CONFIG file can exist on a disk.

- Removed the MODIFY and DELETE from the main menu and added DISPLAY FILE, GLOBAL REPLACE and OTHER FUNCTIONS.

- DISPLAY FILE is similar to the VIEW FILE mode but allows you to view 20 records at a time instead of individually. In 80 column mode the first 10 characters of the first 6 fields will be displayed. In 40 column mode only 3 fields are displayed. The prompts are similar to the VIEW prompts. Very handy for fast scanning of the record file.

- At the OTHER FUNCTIONS menu you will find the ability to program your function keys F1-F8 to remember a sequence of key presses. Carriage returns are entered as a slash (/). When you move through the program write down the key presses that you usually enter to get to where you are going, then program a function key to do the same job.

Examples: When you first turn on your computer and load DFILE128 you would normally load up your record file "FAMILY" by pressing [R] to read file, then enter the name [F][A][M][I][L][Y] followed by [RETURN]. Your file will be loaded and you will be back at the main menu. You can simplify this process by programming [F1] to be:

F1= rfamily/ (Note: non-shifted characters, / = RETURN)

Now by pressing [F1] as soon as you load your program your file will be loaded with one key press. Your secretary will love it! A total of 255 characters may be stored amongst all 8 keys and they will work anywhere in the program. You can actually load a file, load a print format, print the file, go back and quit the program all with one keypress. Looks like this (file="mail list"):

F2= rmail list/prl/pa//eeeq (Your sequence may be different)

Note: Several keypresses that you normally perform (such as the print options menu) have been eliminated because of the configuration file. Others are added because of additional features (subtotals). Always use [E] to denote exiting menus with the function keys. [ESC] will exit the prompt. Settings are saved in CONFIG file.

o Another function chosen at the OTHER FUNCTIONS menu is the DUPLICATE CHECKING feature. You simply toggle this ON and OFF. What it does is check your FIRST field when you are adding records for duplicate data in the other records in your file. This is handy for entering part numbers, ham radio call signs, etc. where the data is unique and you do not want double entries. Since the entire file must be checked each time, there will be a slight delay after you press [RETURN] on the first field. If no match is found you can continue with the record. If the first field data already exists somewhere you will be notified and not allowed to continue. Last setting is saved in CONFIG file.

o GLOBAL REPLACE is a new feature at the main menu which allows you to search for specific data in one field and replace the data in that field (or a different field). This is handy for changing membership dates and special codes that are common in many records. The program will display which records it has found and show you the change. Press [ESC] if, for some reason, you see records changed that were not supposed to because of an accidental entry!

o The CLONE FIELD feature, which will duplicate the previous record's data in the current record (when ADDING or MODIFYING records), will now immediately display the data on the screen when you press [=]. Previously the equal sign was only displayed until you went back to view the record again. Deleting the data in the field has changed also (by entering a minus sign [-]) and you are shown special keys at the top of the screen.

o Data that are numbers and not strings will now be SORTED correctly without having to pad the numbers with zeros (0). Before, you had to insure that all numbers had the same number of characters in order to do a proper sort (Ex: 0001, 0123, 2345, etc). When asked for the field to be sorted, just enter the field number preceded by the number symbol [#] as in #2. The program will then treat that field as a value instead of a string. Logical searches (when in the print routines) may be more accurate by preceeding the DATA with a # as in searching for number #23456.12 .

o At the PRINTER CONFIGURATION menu you will notice option [5] which allows you to 'print' your reports, calculated report or mailing labels, to the disk as a sequential file which is a compatible WORDFILE document. You can then add additional comments using the wordprocessor or even transmit your report into the home office with your modem. WARNING: WORDFILE will become extremely confused if your report widths are greater than 79, so do not use wide reports if you are planning on importing them to WORDFILE. Other wordprocessors may accept the reports as-is.

o Other features at the PRINTER CONFIGURATION menu are:

Secondary address selection for PRINT CODES and TEXT CODES.
Page number justification, LEFT, RIGHT, CENTER (reports).
Number of lines per page (66 is default for reports).
Sending PRINTER CODES simplified (Example: 27/66/1).

o At the PRINT OPTIONS menu you will discover a new report type called BAR CHARTS. This will print in graphic representation the value of one field in your records. One other field may be used to label the data. This routine only works on PURE Commodore compatible printers that utilize the repeating function of CHR$(8)CHR$(26)CHR$(XXX). Siekosha and Star printers (no external interfaces) seem to work OK. Sorry, no gaurantees that this will work on yours.

o There are new operators for making equations in CALCULATED REPORTS. They are >= or => (greater than or equal to) and <= or =< (less than or equal to). In addition the ++ operator (sum all) will work with fields (F2++F9) as well as with columns as previous (C1++C12). Additional error checking has been incorporated to make sure you have entered your equations correctly.

o The size of all reports have been increased to allow 16 columns for a total width of 160 characters across the page.

o CALCULATED REPORTS will now calculate and print subtotals. Subtotals, if selected when choosing which records to print, will be triggered (calculated and printed) whenever a particular field changes value. If, for example, you have chosen the DATE field to trigger subtotals (the record file is also assumed to have been sorted by DATE), every time the data in the DATE field changes value, the sum of all the records having the same date will be printed. This can also be used for NAMES, ACCOUNT NUMBERS, etc.

o You can START printing records at a selected record number and STOP printing at a selected number also.

GENERAL NOTE: Some 1571 disk drives experience errors and speed losses when writing sequential files to the back side of the disk (you are on the back side when the number of BLOCKS FREE on the directory is less than 664 blocks). To be safe, try to stay on the front side of the disks (Program files are OK on the back).

Since your function keys are reprogrammable, you must reset the computer to restore the keys to their original values.

Michaelsoft, Mike and Becky Konshak
4821 Harvest Court
Colorado Springs, CO 80917

michaelsoft™

A COTTAGE INDUSTRY OF HOME SPUN SOFTWARE

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