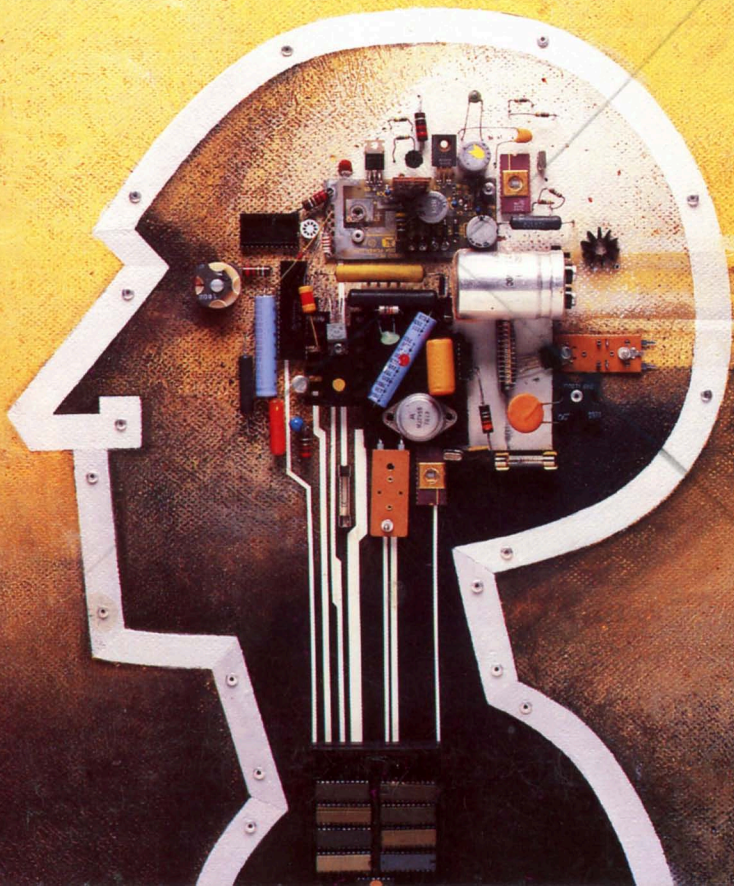


# PERSONAL DATA BASE



\$29.95

# CompuColor Programming Software Series

## Personal Data Base

File and Data management system for addresses, inventories, etc.



**CompuColor<sup>®</sup> Corporation**

### IMPORTANT INFORMATION

A. For extended media life of your Sof-Disk™ — take the following precautions.

1. Do not put fingers on the precision surface.
2. Insert the Sof-Disk carefully into the disk drive.
3. Keep the Sof-Disk far from magnetic field which will erase it.
4. Store the Sof-Disk in the jacket when not in use.
5. Handle the Sof-Disk with care. Bending and folding will damage it.
6. Sof-Disks are best stored at temperatures ranging from 10° to 52°C or 50° to 125°F.
7. Do not leave Sof-Disk in disk drive while turning your CompuColor II ON or OFF.

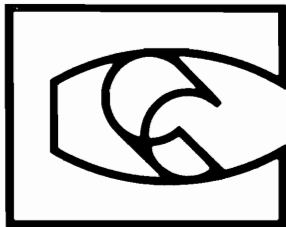
B. ALL COMPUTER PROGRAMS ARE DISTRIBUTED ON AN "AS IS" BASIS WITHOUT WARRANTY.

CompuColor Corporation shall have no liability or responsibility to customer or any other person resulting from the use of computer programs. Copyright © 1978 by CompuColor Corporation.

DATA BASE SYSTEM

# **Compucolor II**

SOF-DISK LIBRARY



**A complete system for storing, selecting,  
retrieving and printing information of  
all types.**

Manual No. 999266 Rev.  
Library Album No. 990016

Requires 16K user RAM



## DATA BASE SYSTEM

TABLE OF CONTENTS	PAGE
I. SUMMARY	2
II. INTRODUCTION TO THE DATA BASE	3
III. HOW TO USE THE DATA BASE	7
Configuration Selection	9
Data Base Menu	11
Creating a Descriptor File	12
Creating a Random File	16
Editing Random File Records	17
Listing the Contents of an Entire File	20
Creating a Printer Format File	21
Print Using The Format File	26
File Recovery Program	29
IV. SPECIAL APPLICATIONS	30
Numeric Key Field	30
Files on Separate Disks	31
Optimum Record Length	31
Disk Storage Requirements	32
V. DATA BASE STRUCTURE	33
Data Base Programs	33
Data Base Program Structure	34
Data Base Files	35

## I. SUMMARY

The Personal Data Base consists of a series of BASIC programs which allow you to create a data base system tailored to your requirements. The available features include the ability to:

1. Create a file which determines the contents of each data base record.
2. Update, add and delete data base records.
3. Print all records in alphabetical order.
4. Create a format file which determines how output will be printed.
5. Search all records using selection criterion.
6. Create a file using selection criterion.
7. Index a random file created by another program for use with the data base.

Although the Data Base will run in 16K, long records may require more memory. The Data Base will work properly on a system with a single disk drive, but it operates more efficiently on a system with a dual disk drive. For sophisticated applications, the addition of this external drive is recommended. It is available from your CompuColor II dealer.

## II. INTRODUCTION TO THE DATA BASE

This section will give you an understanding of how the Data Base works and what its capabilities are. Detailed instructions about using the programs are discussed in the next sections.

In order to familiarize you with the Data Base, an example has been selected which illustrates a standard use of the program's features. This example will be used throughout the documentation, and it has already been created on your Personal Data Base Sof-Disk. Having the example at your disposal while reviewing this manual will help you become completely familiar with each data base program and its function within the Personal Data Base.

This example is the sort of Data Base application that might be useful to a store owner who wishes to create a mailing list so that information can be sent to people who fit into a specific category. A method is needed for keeping records of the following information for 20 customers. (We will limit this to 20 customers for the sake of brevity.)

1. Customer's Name
2. Address
3. City
4. State
5. Zip Code
6. Age
7. Number of Dependents
8. Name of Dependents (room for up to 4)
9. Whether or not they are a computer owner

The conventional method of record keeping would require that some sort of form be designed that would have lines on which each piece of information could be entered. This form would be filled out for each customer and filed under the customer's name. As information changed, the old form would be updated or replaced.

The Data Base operates in a very similar manner. The major difference is that the papers and the filing cabinet are replaced by a Sof-Disk. The contents of each form is stored on the disk and is referred to as a "record". Each individual item in a record is known as a "field". The first field is known as the "key field". Our example is made up of 12 fields (item 8 is 4 fields long) with the key field being the Customer Name. The key field is used to identify individual records and therefore must be unique for each record. In this example, it means that no two customers can have the same name.

In the Data Base a "DESCRIPTOR FILE" is created which describes each field in the record. Creating this file is the same as determining the contents of a form. This file contains the name and length of each field in the record. Once the descriptor file is completed, a "random file" is created. The random file is where the actual data associated with each field is stored. Two other support files are created at the same time as the random file. They are called the "INF" and "INX" files. The INF file is similar to the descriptor file and is used to determine the fields within the random file records. The "INX" file is used to locate records within the random file. Once the Random, INF and INX files exist, data can easily be entered into the files. The Data Base contains programs which give the user the ability to add, delete and edit any record in the random file.



The real power of Data Base is its ability to search through the records. Suppose, for example, that you wish to make a list of every customer who owns a computer. The Data Base contains a feature which allows you to interrogate each record according to a selection criterion and print out a list of records which meet the criterion. In this example, all of the records which have the "Computer Owner" field equal to "Yes" could be printed. It may seem that such a list would be easy to create by hand, and if there were only twenty customers, it probably would be. But the Data Base allows for a search through a large number of records in a short period of time. The hand-sorting, reading and typing involved in doing such a search manually would literally take hours. More intricate searches and sorts can be done by the Data Base. For example, it could make a list of all computer owners between the age of 25 and 40 who live in California and have 2 dependents.

When records are printed it is not necessary to print the entire contents of each record. If only a mailing list is wanted, the items of interest would be the Name, Address, City, State and Zip Code. The other information would be important to our search but not necessary for our output. This differentiation is done by the creation of a "format file". This file determines which fields are to be printed and how they will be arranged. You could, for example, arrange the output so that it would print the mailing list onto a pin-feed form with a return address already on it. Several different format files can be created and saved so that data from the same records can be printed differently depending on the requirements. The Data Base contains provisions which allow you to output through the RS232 port to a printer, monitor, or another computer.

The Data Base uses advanced techniques for the handling, sorting and updating of records. Although this is transparent to the user, it is important if files created outside Data Base are to be used. In order to use files of this type, the Data Base includes a program which will create the information and index files necessary to support a random file.

### III. HOW TO USE THE DATA BASE

The Data Base System is supplied with two Sof-Disks. One Sof-Disk contains seven of the nine Data Base programs (Program Disk). The second disk contains sample data files and the remaining two Data Base programs (Data Disk). The last two programs are supplied on the Data Disk because the manufacturing method used limits the amount of information which can be placed on a disk. However, there is room on the Program Disk for the two remaining programs. These programs should be added to the program disk by the user. The following procedure should be followed:

- (ESC) W        Initialize Basic  
                  Insert Data Disk
  
- LOAD"PARLST"   Loads BASIC program "PARLST"  
                  Insert Program Disk
  
- SAVE"PARLST"   Saves BASIC program "PARLST"  
                  Turn Program Disk over
  
- SAVE"PARLST"   Saves BASIC program "PARLST" on second side  
                  Insert Data Disk
  
- LOAD"INKDBS"   Loads BASIC program "INKDBS"  
                  Insert Program Disk
  
- SAVE"INKDBS"   Save BASIC program "INKDBS"  
                  Turn Disk over
  
- SAVE"INKDBS"   Save BASIC program "INKDBS" on second side

Once you have accomplished the program transfers, your Data Base Program Disk is complete and can be used as indicated in the Documentation. If desired, the "INXDBS" and "PARLST" programs can be deleted from the Data Disk. This will provide more space on the disk for data storage. Use the following procedure:

(ESC) D	Enter PCS Insert Data Disk
DEL PARLST.BAS;01	Deletes BASIC program "PARLST"
DEL INXDBS.BAS;01	Deletes BASIC program "INXDBS"

The above procedure can be repeated for the second side of the Data Disk. These programs should not be deleted until you are certain that they were properly transferred to the Program Disk without any errors.

The Data Base is loaded by inserting the Sof-Disk into the CompuColor Disk Drive and depressing the "AUTO" key (BASIC must have been initialized). The following display will appear on the screen:

#### DATA BASE MANAGEMENT SYSTEM

Select your system configuration from the following list:

1. Single Disk System - Program and Files on Same Disk
2. Single Disk system - Program and Files on Separate Disk
3. Dual Disk System - Program on CD0, Files on CD1

Enter Number and Hit Return

## CONFIGURATION SELECTION

You must select one of the three configuration options by typing its corresponding number and hitting return. If item 1 is selected, you will be limited to the space remaining on your program disk for your Data Base files. If you have a single disk system, it is recommended that you select configuration 2. This will provide you with an entire disk for your files. When this configuration is selected, the program will direct you to insert alternately the "Program" or "Data" disk. The "Program Disk" is the disk containing the Data Base BASIC program and the "Data Disk" contains the Data Base files. The following commands are used to indicate when to change the disks:

INSERT DATA DISK - HIT RETURN  
INSERT PROGRAM DISK - HIT RETURN

Once the proper disk is inserted, close the disk drive door and hit return to continue execution of the program.

After the configuration has been selected, the program will print:

WILL YOU BE USING A PRINTER?

This question is answered by entering "Y" (yes) or "N" (no). If the answer is "YES" then the program will ask:

#### WHAT BAUD RATE IS REQUIRED?

The only acceptable baud rates are 110, 150, 300, 1200, 2400, 4800, or 9600. If a number other than these is entered, the question will be repeated. Answering "yes" to this question does not automatically send all output to the printer. Unless this option is selected, however, you will not be given the opportunity later in the program to select the printer as the output device. The output device does not have to be a printer. It could also be a modem or even another computer. There are two programs which allow output to be directed to a printer. They are the listing and printing programs.

## DATA BASE MENU

Once the configuration and printer options have been selected, the Data Base Menu will appear and display the following options:

1. Create a Descriptor File
2. Create a Random File
3. Edit Random File Records (Add, Delete, Update)
4. List Entire Contents of a Random File
5. Create a Format File
6. Print Using the Format File
  - a. Print all Records
  - b. Make a Selective Printing
  - c. Create a Selective Hold File
7. File Recovery Program

All of the above programs will ask for the names of various files. It is up to the user to select these names. The only restriction is that a name must contain only letters and/or numbers and cannot exceed six characters. It is a good practice to use the same name for files which will always be used together. If the same name is used for the same file type the later version will be used by the program. For example, if a second descriptor files is created called **SAMPLE**, the program will use the second version. The previous version of the file can be used if the file name is followed by a semi-colon and the version name. For example:

"SAMPLE;01" (must be entered with quotation marks  
when version number is added)

The word "EXIT" may not be used as a file name. EXIT is a reserved word which allows the user to abort a program by entering EXIT as a file name. This will result in the program returning to the Data Base Menu.

### 1. Creating a Descriptor File (Option 1)

This option is selected to create the descriptor file which describes the contents of each record, It is always the first option you will use when setting up any new file, and it is used only once for each file. As mentioned earlier, the individual descriptions in the descriptor file are referred to as **fields**. In our example "CUST-NAME" would be the name of first field, "ADDRESS" the second, etc. A field name may consist of up to 16 characters with no internal spaces, numbers, or special characters. The field name is followed by a specifier which indicates the field type and length. Two types of fields may be specified; Alpha and Numeric. These are entered as "A" or "N".

**Alpha** -- An alpha field consists of a string of characters which can contain both numbers and letters. If an alpha field is specified, the length of the field must also be specified. (1-80 characters) For example:

**Address A18**

specifies an Alpha field named Address which is 18 bytes long.



**Numeric** -- A numeric field consists of a numeric quantity in floating point format. It is not followed by a number indicating length since a numeric value always occupies 4 bytes of space. For example:

**Age N**

specifies a numeric field named Age (always 4 bytes long).

The first field (key field) of the descriptor file must be an Alpha field. It may be followed by any number of alpha and numeric fields. A field name consists of up to 16 characters with no internal spaces or commas. A field name can be used only once. Names such as AND, NOT & OR, which are BASIC logical operators, are not permitted.

When option is selected, the program will ask for the name of the descriptor file and will check to see if a descriptor by this name already exists. A file name is selected by the user and can contain up to 6 letters or numbers.

### **Pre-Existing Descriptor File**

If the file does exist, the program will display the contents of descriptor file and ask the user to select one of the following options:

1. **Create new file:** This option restarts the descriptor program so that a new file name can be used.
2. **Edit the current descriptor to create a new file:** (See editing descriptor file)
3. **Print descriptor file:** return to menu
4. **Return to Menu**

## **No Existing Descriptor File**

If no descriptor file exists, the program will respond with:

KEYFIELD COL (1):

and wait for the field name and specifier to be entered. Remember that the key field must be an Alpha field. "COL (1)" indicates the starting position (column) of the field within the record. The field name and specifier are entered at the same time and separated by a blank. Once the key field has been entered the other fields are entered in the same manner. If the program detects an invalid entry the word "ERROR" is displayed and the field must be reentered. Once all fields have been entered, hitting RETURN will exit this mode. The program will reprint the fields and ask if you wish to make any changes. If changes are necessary, the file can be edited as described below.

## **Editing The Descriptor File**

Two editing functions, "Add" and "Delete", are available. Any field may be deleted, but a field may only be added to the end of the file. If "D" is used the program will ask for the name of the field to be deleted. Adding a field is done in the same manner described previously. The descriptor file is not actually created until editing is complete. Therefore, the BREAK, LINEFEED, and RUN sequence can be used to start over if necessary.

## Repeat Factor

It is possible to create a single field with space for more than one entry by using a repeat factor. The repeat factor follows the field specifier and is separated from it by a blank. In the following example, field 7 has a repeat factor of 4. By doing this we have provided room for the names of up to 4 dependents. We could do the same thing by creating 4 individual fields. That, however, would require four different field names. We will see later that this can also be beneficial when making selective searches.

## The Descriptor File

The following example shows how a descriptor file might be created to accept the information needed in our standard example:

```
KEY FIELD COL (1): CUST-NAME A25
FIELD 1 COL (26): ADDRESS A18
FIELD 2 COL (44): CITY A12
FIELD 3 COL (56): STATE A2
FIELD 4 COL (58): ZIP N
FIELD 5 COL (62): AGE N
FIELD 6 COL (66): NODEP N
FIELD 7 COL (70): DEP-NAME A12 4
FIELD 8 COL (118): COMPOWN A3
FIELD 9 COL (121): (RETURN)
```

Notice that no spaces are used in the field names and that the field specifier and repeat factor are separated by blanks. Notice also that field 7 is 48 bytes long (12x4) due to the repeat factor. Field entry was terminated by hitting "RETURN" at field 9. Since field 9 would start at the 121<sup>st</sup> position (or byte) in the record, the length of the record is 120 characters (or bytes).

Notice that no spaces are used in the field names and that the field specifier and repeat factor are separated by blanks. Notice also that field 7 is 48 bytes long (12x4) due to the repeat factor. Field entry was terminated by hitting "F5/END" at field 9. Since field 9 would start at the 121<sup>st</sup> position (or byte) in the record, the length of the record is 120 characters (or bytes).

## Printing the Descriptor File

An existing descriptor file can be displayed or printed out via the ES232 port by selecting option 3.

## 2. Creating a Random File

This option is the second step used to create files. It is always used after the Descriptor file is completed, and is used only once for each file. The following items will be displayed when this option is selected. Items in bold type represent user inputs.

Enter Name of Random File: **SAMPLE**  
Enter Name of Descriptor File: **SAMPLE**  
Enter Today's Date (M,D,Y): **3,22,79**  
Number of Expected Records: **20**

This Will Take Awhile . . .

### A Random File Named SAMPLE Has Been Created With 20 Records

The random file is the file containing the user data. In order to create a new random file the user must also enter the name of the Descriptor file. This is necessary in order to allocate the proper amount of space for each record. This is followed by the date and number of expected records. The number of records cannot be increased once the random file is created. Therefore, you should be certain that the maximum number of records that will be required are created. The program will create the random file and its two supporting files. The time required to create these files will range from a few seconds to several minutes depending on the file size and number of records.

### 3. Editing Random File Records (ADD, DELETE, UPDATE)

When this option is selected, the program will ask for the name of the random file and then display the following:

```
FILE NAME-SAMPLE      DATA BASE    LAST ACCESSED
                        5 27 79
RECORDS LEFT-8
ENTER FUNCTION:      ADD,DELETE,UPDATE,LIST,RANGE,END
```

~~This option is used to do the actual data entry that enters information into each record. It performs all editing of the random file.~~ While other areas of data base can read the random file, none can change its contents. Three editing functions (ADD,DELETE and UPDATE) and three supporting functions (LIST,RANGE and END) are available. Only the first letter of each function needs to be entered. The upper left corner of the display

indicates the name of the random file being edited. Below it is the number of unused records available. The upper right corner indicates the date the program was last accessed.

**ADD** - When this function is selected the program will print the name of the key field and wait for the user to input the field data. For example:

**CUST-NAME MICKEY MOUSE**

"**CUST-NAME**" is the name of the key field and **MICKEY MOUSE** is entered as the customer for this record. After the key-field is entered, the program checks to see if a Key by this name already exists (the key field for each record must be unique). If the Key does exist, "KEY ALREADY IN FILE" is printed and the cursor returns to the function entry line of the display. If the key does not exist, the name of each field is listed down the left side of the screen. Each field is followed by its specifier (S-string, N-numeric) and a blue shaded area which indicates the length of the field. The cursor indicates the field for which data is to be entered. After each field is entered, the cursor will automatically move to the next field. The entries must be kept within the blue shaded area so that the data will fit within the allocated space. To leave any particular field blank, hit TAB followed by a RETURN. When the last field has been entered, the cursor will return to the function entry line.

**DELETE** - As might be expected, the DELETE function is used to delete records from the random file. When this function is selected, the program will ask for the name of the key field of the record to be deleted. The Data Base will locate the record and delete it from the file. "KEY DELETED" is printed and the program returns to the function entry line. If an attempt is made to delete a non-existent record the program will print "KEY NOT IN FILE" and return to the function entry line.

**UPDATE** - Update operates in the same manner as the ADD function, although the record must already exist and the entire contents of the record will be displayed. New data can be entered for each field, except the key field, in the same manner as with the ADD function. If a field does not require changing, hitting TAB followed by RETURN will bypass it. If the record does not exist, "NO PREVIOUS RECORD" is printed. Key fields cannot be updated by this command. In order to change a key field, you must delete it and re-enter it entirely.

**LIST** - This function list the contents of the specified record. If the record does not exist, "KEY NOT IN FILE" will be displayed.

**RANGE** - The range function allows the user to print a list of the key fields between two limits. For instance, entering "A" as the lower limit and "Z" as the upper limit would print a list of all key fields that fall between "A" and "Z" in the alphabet. When the RANGE function is entered, the program will ask for the following:

UPPER LIMIT =  
LOWER LIMIT =

The program will list all keys fields between specified limits in groups of 20. After each group is listed, it will ask if you want to continue the listing. This allows long listings to be terminated.

**END** - END must be the last function executed after editing the random file. This function moves the buffers to the disk and closes all open files. Failure to use this function after entering data could result in loss of that data, as it would not be added to the disk files.

#### 4. Listing the Contents of an Entire File

When this option is selected the following is displayed:

##### **LIST DATA BASE FILE**

Enter Name of Random File: **SAMPLE**

Output to a printer or the screen? **5**

File Last Accessed **3-25-79**

(list of records in alphabetical order)

This program is used by simply entering the name of the random file. If the output is directed to the printer, the program will return to the screen after the records have been printed. Unless the baud rate was set on the first Data Base display, the option to select the printer as the output device will not be available.

Once the records have been listed, the user has the option of listing another file or returning to the Data Base Menu.



## 5. Creating a Printer Format File

This program creates a file which determines the arrangement of fields when the general printing program is used. Four specifiers, A,N,X and #, are used to control the printing location and length of fields.

**A** - This command indicates an alpha field and must be followed by a number indicating how many characters are to be printed from the field. It is possible to start printing with any character in the field by following this number with a decimal and starting character. For example:

A5 prints first five characters of field  
A5.2 prints five characters starting with the second character.

If there are fewer characters in the field than specified, the available characters will be justified to the left and followed by trailing blanks.

**N** - This command follows numeric fields and is followed by a number which indicates the length of the print field and location of the decimal point. The decimal point counts as one digit. All numeric fields are right justified with leading zero suppression.

Examples: N5 prints a number as a five digit integer  
N5.2 prints a number with two digits before the decimal and two after.

X - This is used to space between fields and is followed by a number indicating the number of spaces required.

Example: X5 will leave 5 spaces

# - This generates a carriage return and starts the next field on a new line. When the format program is entered, the program will print:

ENTER NAME OF FORMAT FILE: **SAMPLE**

The program checks to see if a format file by that name exists. If the file already exists, the user is given three options:

1. Edit the format file to create a new file.
2. Create a new file.
3. List file - return to menu.

The first option allows you to edit the current format to create a new one. Option two restarts the Format program so that a new name can be entered. Option 3 lists the file and returns to the Data Base Menu. If the printer option was selected it can also be listed to a printer via the RS232 port.

### Creating a New File

The format file is created by entering the name of the desired field followed by its field type and length. If a field has a repeat factor, the subscript number of the field in parenthesis must follow the field name. The program will indicate the current line and column at which the next field will start. The commands "X" and "#" can be used to start a field at any desired point. It is not necessary to print the fields in the same order in which they were entered, and it is not necessary to print all fields. But, those fields which are printed must correspond exactly in name and length to those created by the Descriptor File. Entries are terminated by hitting return after the last entry. For example:

    Create a format file named ADR which will print the records according to the following format.

```
CUST-NAME
ADDRESS
CITY                STATE        ZIP
```

The following entries are required:

		REMARKS
LINE 1		
1: COLUMN 1	CUST-NAME A25	
2: COLUMN 26	#	START NEW LINE
LINE 2		
3: COLUMN 1	Address A18	
4: COLUMN 19	#	START NEW LINE
LINE 3		
5: COLUMN 1	City A12	
6: COLUMN 13	X2	MOVE 3 SPACES
7: COLUMN 15	State A2	
8: COLUMN 17	X2	
9: COLUMN 19	ZIP N5	
10: COLUMN 24	#	
LINE 4		
11: COLUMN 1	<RETURN>	END OF FORMAT

When the format entries are completed, the program will show how the file will print and asks if there are changes. If no changes are necessary, the program will ask for the name of the descriptor file. This is necessary to ensure that all fields in the format file exist. If a field cannot be located the error:

CANNOT LOCATE FIELD - name of field

is displayed. The error can be corrected by editing the Format. If all fields exist in the descriptor file, a format file is created and the program returns to the Menu.

## **Editing the Format File**

Changes can be easily made to the format file. By answering 'yes' to the question "Are there any changes?" the program enters an edit mode. Any line can be changed by entering its line number. A correcting entry can then be made. Depending upon the gravity and number of errors made while creating the format file, it may be easier to start over than to make all of the changes through the editing commands.

## 6. Print Using the Format File

This program is a general print program. It allows the user to print or output records according to the format specified by the Format file. The following is displayed on entry:

PRINT FILE WITH OPTIONS

ENTER NAME OF RANDOM FILE: **SAMPLE**  
SAMPLE WAS LAST ACCESSED ON: **3,25,79**  
ENTER NAME OR FORMAT FILE : **ADR**

SELECT ONE OF THE FOLLOWING OPTIONS:

1. Complete Listing
2. Selected Listing
3. Create a Selective Hold File

The following is a description of the three available options:

1. **Complete Listing** - Selecting this option will print all records according to the format specified.
2. **Selected Listing** - This option represents the real power of Data Base. With it, files can be searched according to a selection criterion printing only those files which meet that criterion.
3. **Create a Selective Hold File** - This is essentially the same as a selective listing. Instead of printing the results of a selection, a file is created which contains pointers to the selected records. The last record in this file will have a '0' in it. If options 1 or 2 are selected and the baud rate was previously set, the program will print:

OUTPUT TO THE PRINTER OR SCREEN?

The question is answered with a "P" or "S". If "P" is selected, the output will be directed out the RS232 port. Once outputting is complete, the system will return to the screen as the output device.

### **Entering Selection Criteria**

Options 2 or 3 require that selection criteria be entered. Any field or combinations of fields may be used to determine the criteria for listing. The fields need not be a part of the format but must be in the record. If the field contains a repeat factor the number of the desired field must follow the field within parentheses. Care must be exercised when entering the selection criteria as any mistake will result in a program error. The following rules must be adhered to:

1. The logical expression must begin and end with brackets ( [ ] )
2. Strings must be within **single** quotation marks.
3. The logical operators AND,OR and NOT may be used as well as parentheses, <,> and =, <=,>=

### **Examples:**

[CUST-NAME>'JONES']

all records with a name following JONES in the alphabet will be printed.

[AGE<30]

all records with an Age less than 30 will be printed.

[AGE>22 and AGE<35]

all records with an age between 22 and 35 (exclusive) will be printed.

In the Introduction to Data Base two examples were discussed for which we will now create a logical expression.

[COMPOW='YES']

a list of all computer owners will be printed

[COMPOW='YES'AND AGE>=25 AND AGE <=40 AND STATE='CA AND NODEP=2]

a list will be compiled of all computer owners between the age of 25 and 40 who live in California and have 2 dependents.

More than one line can be used when entering the selection criterion. Hitting return will move the cursor to the next line and entry will continue. Entry is not terminated until the closing bracket (]) is used. The maximum length for entry is 255 characters.



## 7. File Recovery Program

This program should not be used as a routine matter since it is quite time consuming. It is used to recover files that are lost because the END function is not used after data entry. An occurrence such as a power failure or an accidental striking of the CPU RESET key while editing records would require this option to restore files that were entered but not saved. Its other use is for indexing files created outside the Data Base.

In order to use this program, a descriptor file must exist. In the event it does not, a new one can be created. The program will create the information (INF) and index (INX) files.

#### IV. SPECIAL APPLICATIONS

##### NUMERIC KEYFIELD

The Data Base is structured such that the keyfield must always be an alpha field. In some instances, you may wish to use a numeric key field. Unique identification numbers might describe the record contents more effectively in areas such as inventory, library systems, etc. The fact that we are limited to an alpha field does not preclude the use of numbers. In fact, longer and more precise numbers can be stored in this manner. There are three short-comings to using a number as a key field.

1). . More space is required to store numbers. A numeric field occupies 4 bytes, but an alpha field will take one byte for each digit plus the decimal point.

2). Numbers must be entered in such a manner that the decimal points fall at the same point in the field. If integer numbers are used this means that the entries must be entered in a right justified format. This is necessary for the numeric values to be compared correctly. Data Base supresses leading blanks so that leading zeros must be used when entering a numeric values.

3). Use of the plus and/or minus signs may result in keys not being organized in the proper numeric sequence.

If it is not necessary to order records by the key fields, the last two short-comings are rendered inconsequential.

## FILES ON SEPARATE DISKS

The Random, Index, and Information files must always be on the same Sof-Disk. The Format and Descriptor files can be on separate disks if desired. However, this cannot be done on a single disk system, since files from each disk must be open at the same in some instances. On a single disk drive system, the initial configuration option selected should be #2. (Single Disk System-Program and Data on separate disk). When a file name must be entered or specified it can be preceded by the device specifier in order to direct it to the desired disk drive.

Example: "0:TEST" "1:SNIPLE"

When a file specifier is used the entry must be within quotes.

Attempting to do this with configuration option #3 (Two Disk System) will result in an error since the device specifier is automatically added to the file name.

## OPTIMUM RECORD LENGTH

Considerable disk storage space can be saved if care is used in creating the descriptor file. The descriptor file determines the length of the records in the random file. Data is organized on the disk in 128 byte blocks. If a record length is a multiple of 128 or divides evenly into 128 the optimum use is made of the disk space. A record that is 32 bytes long would allow 4 records to fit exactly into 128 bytes.

Data Base will automatically adjust the blocking factor to insure that the loss of space does not exceed 10%. In certain cases this could result in a large enough blocking factor to cause an out of memory error. For example, a file with records of 129 bytes could require a blocking factor of 10 so as not to exceed 10% loss of space. This would require a memory buffer 1290 bytes long. A file with records 128 bytes long would only require a 128 byte memory buffer. The following record length will optimize disk storage and buffer requirements: 1, 2, 4, 8, 16, 32, 64, 128, 256, 384... (increments of 128).

### **DISK STORAGE REQUIREMENTS**

Determining the number of records which will fit on a single disk is quite complicated because of the numerous variables involved. Computing the length of the random file is just a matter of computing the number of records in a block and then figuring how many blocks are required. (See above.) Each disk contains 400 128-byte blocks. The first 4 blocks are usually used for the disk directory, leaving 396 blocks for the files.

The Random file also requires two supporting files, the Index and the Information file. These are created at the same time as the Random file. Therefore, sufficient space must be available for them on the disk.

The Information file will be 25 bytes longer than the Descriptor file. Determining the size of the index file is extremely complex. As a general rule of thumb it will be about one third the size of the Random file. This could vary considerably if the key field is extremely large. In the SAMPLE example included on the Data Disk, 235 records will fit on a disk if it contains only the RANDOM, INDEX, and INFORMATION files.

## V. DATA BASE STRUCTURE

### DATA BASE PROGRAMS

The Data Base consists of the following nine BASIC programs.

**MENU** - This program is called only once. It is used to select the user's system configuration. This information is stored by backing up the end of BASIC and storing it in high memory. Note that if BASIC is not re-initialized each time MENU is run that the available memory will become progressively smaller.

**MAIN** - This program contains the menu of main Data Base Programs. It chains in all other programs and all programs return to it.

**CREDDL** - Creates the descriptor file.

**CRSEG** - Creates the Random, Index, and Information files.

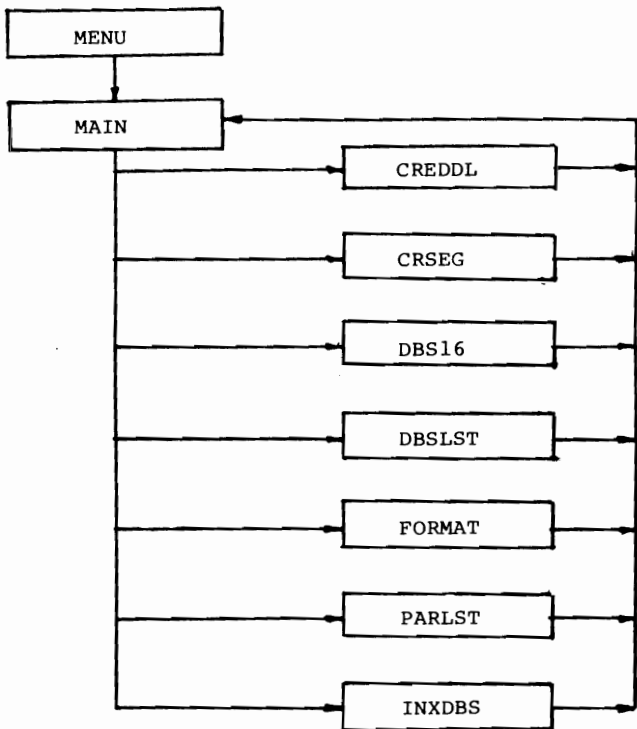
**DBS16** - Edits the random file.

**FORMAT** - Creates the format file.

**PARLST** - Prints or selective prints records in a random file according to a selected format file.

**INXDBS** - Indexes a random file by creating supporting INX and INF files.

# DATA BASE PROGRAM STRUCTURE



## DATA BASE FILES

The Data Base consists of three working files with suffixes: .INF, .INX, and .RND. The system also includes 2 other files with suffixes of .DDL and .FMT. These last two are fixed files and not affected by changes and updates.

**.DDL - Descriptor File:** This file is created by the program CREDDL and contains a list of each field's name, type, and length as entered by the user. Once this file has been created it is never updated.

**.INF - Information File:** This file is created by the subprogram CRSTG. It holds the information necessary to restart the file system. It contains all of the information in the .DDL file and other information such as the date of last access.

**.INX - Index File:** This file contains information on the location of data in the .RND file. Without this file and the .INF file, the system will not work. It is imperative that the END command be used at the end of every edit session or this file will not be valid. In the event this is not possible (power failure) a program called INDECS is provided to rebuild this and the .INF file.

**.RND - Random File:** This file stores all of the data in the system. Its space is allocated to the next available space regardless of the order of the data. The format of this file is stored in both the .DDL and .INF files. Regardless of the arrangement of data, the wasted space of the .RND file will not exceed 10%. Since all of the data is stored in the .RND file, loss of this file is not recoverable. The RND, INF, and INX files are all created at the same time by CRSTG.

**.FMT - Format File:** This program is created by the program FORMAT. It contains user information on how data in the random file is to be arranged when printed using the program PAPERST.

There is a whole library of Sof-Disk Albums available for your **COMPUCOLOR II**. Most Albums include at least four different programs, and that adds up to a lot of variety! There are financial management packages that guide investments, personal information systems, educational programs for learning at all levels, and much, much, more. Below is a list of just a few of the Albums you can buy. To see them all, visit your nearest **COMPUCOLOR II** dealer.

**OTHELLO**

**CUBIC TIC-TAC-TOE**

**MATH TUTOR**

**BONDS AND SECURITIES**

**EQUITY**

**STAR TREK**

**PERSONAL DATA BASE**

**BLACKJACK**

**PERSONAL FINANCE I&II**

**TEXT EDITOR**

**CHESS**

**ASSEMBLER**

**STATISTICS I-III**

**SHARKS**

**SCREEN EDITOR**

**BASIC LANGUAGE**

All software is sold on an "AS IS" basis without warranty.



CompuColor Corporation  
P.O. Box 569 Norcross, GA 30071  
(404) 449-5961