

General Systems
Division
P. O. Box 2150
Atlanta, Georgia 30301

IBM 5110
Computing System

79-1

April, 1979

This is the first issue of the IBM 5110 Newsletter for 1979. In keeping with our objective to pass on various tips and techniques to help you and your IBM 5110 be as productive as possible, this issue presents assorted topics of interest to both the new and the experienced programmer. We hope you will find them useful.

Contents

- Tips on Extending the Life of Your Diskettes
- Programming Notes and Tips (for the Experienced Programmer)
 - Accessing by Key is Faster When. . .
 - A Way to See the Keys in a Key File
 - How to Read Alpha Characters. . .Including Quotes and Commas
 - Efficient Use of Disk by Using the L Spec
 - Efficient Use of the Form Statement
 - On Using SEQ Parameter in Open File
 - Building a Key from Two Numeric Fields
 - Sorting an Index File: General Procedure
 - Special Note on Keys Which Have Numeric Meaning
 - Special Note on Using &REC
 - Entering Percent (%) and Quote (") Characters from the Keyboard

IBM 5110 NEWSLETTER

- Special Technique for Accessing Records
- Time Required to Open a File and Its Index
- Common Problem When OPENing Master and Index Files
- Technique to Read from Anywhere on the Screen
- Implications of RECL (Record Length) and Diskette Format (Sector Size) on Performance and Memory
- What Happens When a File and Its Index are Created?
- How to Verify (Edit) Keys Without Causing a Search to the Master File
- When the HOLD Key Does Not Stop Processing
- How Memory is Allocated During OPEN and CLOSE File
- How to Use the ADDRESS OUT Sort

TIPS ON EXTENDING THE LIFE OF YOUR DISKETTES

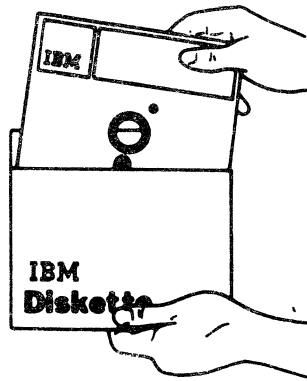
We recently completed an examination of some diskettes returned from our customers as unreadable. The examination showed that in a large majority of the cases the diskettes were unreadable because of physical damage or because of surface contaminants. The physical damage included folds, scratches, bumps and wrinkles in the diskette. Surface contaminants were identified as fingerprints, coffee, sugar, and saliva. All of the above are examples of improper care and handling of the diskette.

Some suggestions for proper handling of diskettes are printed on the back of the diskette envelope. For example:

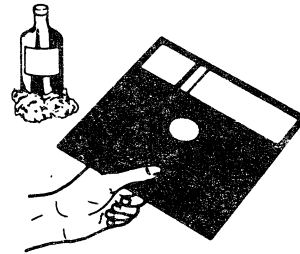
- Protect the diskette by keeping it in the protective envelope when not using.
- DO NOT BEND.
- Insert diskette carefully in drive.
- Never touch exposed portion of diskette proper.
- Do not expose diskette outside this temperature range: 50°F-125°F (10°C to 51°C).
- Never place diskette near magnetic fields:
 - Telephones.
 - Fluorescent lights.
 - Electronic calculators.
 - Dictation equipment.

The next page shows other handling considerations that can extend the productive use of your diskettes.

Return a diskette to its envelope whenever it is removed from the diskette drive.

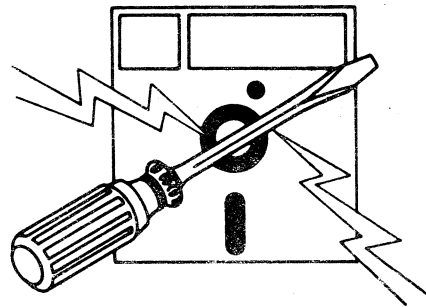
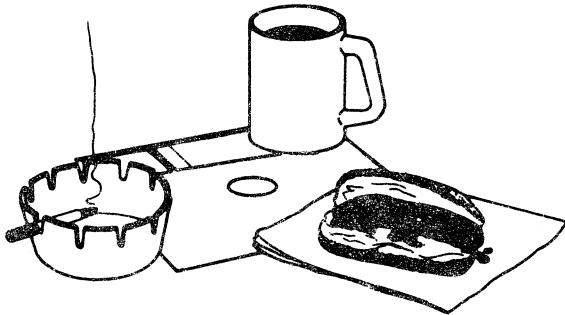


Do not touch or attempt to clean diskette surfaces. Contaminated diskettes must be discarded.



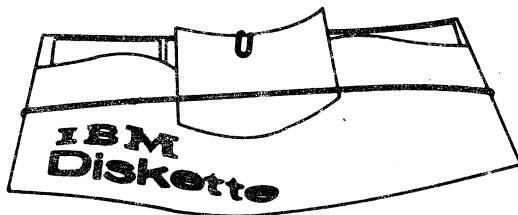
Do not place diskettes near materials that might be magnetized. Data can be lost from a diskette exposed to a magnetic field.

Do not lay diskettes near food, drink, or ashtrays.

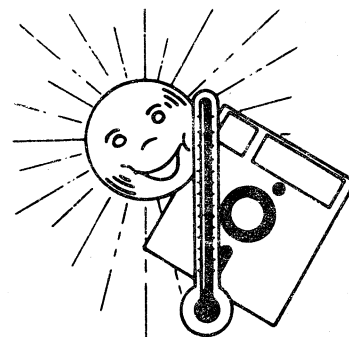
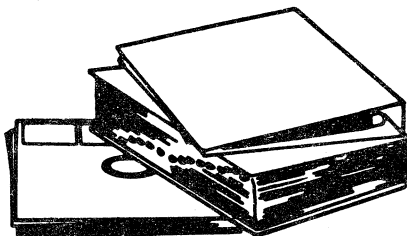


Do not expose diskettes to heat greater than 51.7° C (125° F) or direct sunlight.

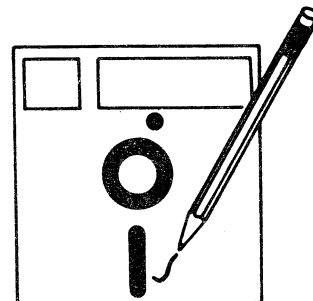
Do not use clips or rubber bands on a diskette.



Do not place heavy objects on diskettes. The weight can cause serious damage.



Do not write on diskettes outside label area.



PROGRAMMING NOTES AND TIPS

Experienced programmers know the value of discovering different ways of accomplishing the same tasks: time saved, storage saved, or flexibility added, to name a few. Following are 21 points that fall under the category "nice to know" when writing or altering an application.

While some of these points are described in the 5110 manuals, most were "discovered" through the use of the system...they resulted from trying different techniques just to see if they would work. All 21 points will not apply to all users; the experienced programmer will find a few points trivial. However, we hope that all of you will find some points useful.

Our next issue will pick up with point 22 and the list will be continued in future issues, so that you may maintain a file of notes and tips on programming your IBM 5110.

1 Accessing by Key is Faster When...

- A The KEY file is sorted (or mostly sorted). This sorting must be done by using the diskette sort feature or the SORT INDEX FILE program in the BASIC USER'S GUIDE.
- B The KW parameter is used in the OPEN for the index. (The KW parameter only helps speed up the key search for the sorted portion of the key file.)

NOTE: It may not be necessary for KW to be very large. In a test case of 8000 keys, KW = 8000 was no better than KW = 120.

2 A Way to See the Keys in a Key File

```
2 DIM K$
8 OPEN FILE FL1,'D80','INDEX',IN
10 READ FILE USING 20, FL1, K$, EOF 50
20 FORM C4 ( use C6 for key length 6 etc )
30 PRINT K$
40 GOTO 10
50 END
```

The two strange looking keys are not keys; they are called marker records. The keys found between these special records are assumed to be sorted and any search for a key will be done in this section using a very fast technique. The use of the KW parameter in the OPEN statement further speeds the search for this section.

3 How to Read Alpha Characters...Including Quotes and Commas

```
10 DIM A$64
20 OPEN FL1,'001',IN
30 WRITE FILE FLS,'ΔΔΔΔΔΔFL1'
   ( Δ MEANS A BLANK - USE EXACTLY 6 )
40 INPUT A$
   ( OR )
40 GET FL1,A$
```

See FILE FLS in BASIC Reference Manual, SA21-9308-1, for more detail. Also, see the 5100 to 5110 Conversion Aid coding shown in the Technical Newsletter SN21-0281. This shows the technique used for input from tape as well as keyboard.

4 Efficient Use of Disk by Using the L Spec

Since numbers are stored in the 5110 in floating point format, any attempt to store numbers on disk that requires a number conversion will take longer than one that doesn't. Floating point also provides efficient data packing.

Numbers with up to 15 digits will store in 8 bytes (rather than 15) with the L specification. Thus, it is not only faster but may be more efficient to use L rather than NC.

TEST CASE:

```
1 DIM A(32)
2 MAT A = (1234)
3 MAT WRITE FILE USING 20,A

20 FORM 32*L
was four times faster than
20 FORM 32*NC4
```

The speed was three times faster for READING.

5 Efficient Use of the FORM Statement

Many programmers feel that a POS must be used before each FORM spec. If the fields are to follow each other, no POS is needed.

If the fields are not next to each other, using X instead of POS will save two characters and still give the 'spacing' capability.

6 On Using SEQ Parameter in Open File

The SEQ parameter in the OPEN statement should ordinarily not be used. Its use means that you may not use direct access or key access. In normal practice, the only reason to use SEQ is when you are also using NOBLOCK and RECL = 128 to make a type B9 file (for diskette interchange with other systems).

7 Building a Key from Two Numeric Fields

J1 = 123 (item number)
J2 = 38 (warehouse number)

PROBLEM: To create an alphabetic combination item/
warehouse 'number' (K\$).

```
1 WRITE FILE USING 2, FLS,J1,J2
2 FORM POS22,NC3,NC2
3 READ FILE USING 4,FLS,K$
4 FORM POS22,C5
```

8 Sorting an Index File: General Procedure (With Diskette Sort Feature)

```
Index file > 'INDEX'
Temporary holding file > 'TEMP'
(If required) work file > 'WORK'
```

A Copy INDEX to TEMP	(PROGRAM)
B UTIL DROP 'INDEX'	(COMMAND)
C Sort from TEMP to INDEX	(DISK SORT)
1) Use WORK if needed	
2) Sort on 4, 12, or 28 characters as appropriate	

9 Special Note on Keys Which Have Numeric Meaning

Remember that only alphanumeric keys are supported. If only numbers are being used there can sometimes be unanticipated results. See Chapter 12 of the User's Guide (Additional Uses of File FLS) for one method of converting 22 to '0022'.

10 Special Note on Using &REC

If you type in &REC from the keyboard the value seen is NOT the last record accessed. The same is true for PRINT &REC in a program. The proper value is determined when &REC is assigned to a variable (eg, A1=&REC).

11 Entering Percent (%) and Quote (") Characters from the Keyboard

Press and hold the Upper Shift key and hit the . key for % or - key for " on the numeric pad.

12 Special Technique for Accessing Records

PROBLEM: You would like maximum flexibility in accessing open invoices and would like to access a given invoice if the customer number and invoice number are known. You also would like to be able to find all open invoices for a given customer (without having to know which they are).

SOLUTION: Using technique shown in number 7 above create a key which is a combination customer/invoice number (assume four digit/characters for customer and invoice number making the key 8 characters). If you want to find a specific invoice 1122 for customer 2345:

```
DIM K$8,I$4
READ FILE USING nnnn,FL1,KEY = K$
( where K$ = '23451122' )
```

If you want to find all invoices for customer 2345:

```
FIRST : READ FILE USING nnnn, FL1, KEY = I$
( where I$ = '2345' )
```

```
SECOND : Issue successive
          READ FILE USING nnnn,FL1,.....
          ( no reference to KEY = )
```

The first READ accesses the lowest invoice for customer 2345. The second type of read accesses the next highest key in the file. If there are more invoices for that customer you will access them in invoice order. You must check that you do not process the next customer's invoices.

13 Time Required to Open a File and Its Index

When these files are opened, the system searches the index file BACKWARDS for the second marker record. Therefore, the more unsorted keys you have, the longer it takes.

14 Common Problem When OPENing Master and Index Files

```
OPEN FILE ..... 'MASTER',OUT
OPEN FILE ..... 'INDEX',OUT,KEY
```

This has been attempted when the objective is to add new records to both the master and index file. This is what happens. New records are added to MASTER but the old index is wiped out and a new one built with only the added records. ALL should have been used instead of OUT for both OPENs.

15. Technique to Read from Anywhere on the Screen

This user subroutine sets the cursor and then reads 18 characters from a given line column. The first two parameters are the line and column.

```
9000 DEF FNR$(L,C)
9010 @9=(64*(L-1))+C
9040 REWRITE FILE USING 9050,FL0,' '
9050 FORM POS 896,C1,POS @9
9060 READ FILE USING 9070, FL0, $$
9070 FORM POS @9,C
9080 RETURN $$
9090 FNEND
```

IBM 5110 Newsletter, Vol 78-2.

16 Implications of RECL (Record Length) and Diskette Format (Sector Size) on Performance and Memory

- A When an OPEN FILE is executed the system must reserve enough memory to read in any record.
- B The minimum requirement is one sector.
- C If the record length is less than the sector size and equal to 1/2, 1/4, 1/8, 1/16, etc, of the sector size, only one sector need be reserved. If it is not, two sectors worth of memory must be set aside to assure that a record that spans two sectors can read. In addition, there will be the time needed to read in two sectors.
- D If the record size is larger than the sector size, enough memory must be reserved to read in the record. You may need even more memory if record size is not evenly divisible by the sector size.

- E A good general strategy to follow is to use a 2D diskette initialized with Format 8 (512-byte sector). Then choose a record size of 8, 16, 32, 64, 128, 256 or 512 bytes.

17 What Happens When a File and Its Index are Created?

EXAMPLE:

```
10 OPEN FILE FL1,'D80',2,'MASTER',OUT,RECL = 64
20 OPEN FILE FL1,'D80',3,'INDEX',OUT,KEY,KP=1,KL=2
```

At OPEN time:

- A Since $KL \leq 4$, the index file will be 8 bytes long.
- B An 8-byte first marker record is placed as the first index file record. The first four bytes are hex 00s. The next two bytes contain the value of KL in B2 format. The last two bytes contain KP.
- C An 8-byte second marker record is placed in the file. The first three bytes contain hex FFs; the remaining bytes are not used.

At WRITE FILE time:

- A Each WRITE FILE causes a record to be put in both MASTER and INDEX. The INDEX records have the key in the first four bytes. The fifth byte is unused. The last three bytes contain the relative record number as in a 24-bit binary number.
- B The system must check for duplicate keys. To do this it keeps track of the highest key already in the index. If the new key is greater than this key, it can be immediately placed in the index. If not, the entire index must be searched to make sure this is not a DUPKEY.
- C As a consequence of the above, if records are loaded in NON key sequence, file building can take a long time--hours are not unheard of!
- D To make file building a relatively fast procedure we suggest that the temporary master be built first, sorted by key, and then use the sorted file as input to a MASTER and INDEX.

IMPORTANT POINT: Even though you follow the procedure in Step D the system will NOT 'know' that the index is sorted and you will NOT get optimal access speed until you use one of the index sort techniques discussed in Item 1 above: "Accessing by Key is Faster When..."

- 18 How to Verify (Edit) Keys Without Causing a Search to the Master File

```
RESET FILE ... KEY = ..... NOKEY nnnn
```

- 19 When the HOLD Key Does Not Stop Processing

The HOLD key does not stop processing if the screen is completely blank (ie the IN PROCESS light is on). If HOLD is needed, change the program so that something is PRINTed to the screen first when the program is executed.

- 20 How Memory is Allocated During OPEN and CLOSE File

CLOSEing a file does not free up memory. Memory is released from a previous CLOSE when another file is OPENed. An easy way to free up memory after CLOSE FILES is to then execute: OPEN FLN '000', OUT.

- 21 How to Use the ADDRESS OUT Sort

EXAMPLE:

File TEMP was the output of a sort where an A was the last parameter in the SORT statement. The input file is MAST. You want to print a report on MAST in the order specified by the SORT.

```
10 OPEN FILE FL1,'D80','TEMP',IN
20 OPEN FILE FL2,'D80','MAST',ALL
30 READ FILE USING 40,FL1,N,EOF 200
40 FORM B4
50 READ FILE USING nnnn , FL2,REC=N
60 PRINT .....
70 GOTO 30
200 END
```

The important aspects of the above are the ALL parameter on the OPEN FILE FL2...'MAST' which allows the REC= in line 50. Also notice that FORM B4 must be used in line 40.

For other information, see IBM 5110 BASIC User's Guide, SA21-9307, pages 81 and 82.

