MEDIA/250
Programming Manual

Part No. 45251-90025

Hewlett-Packard
8010 Foothills Boulevard, Roseville, California 95678

Printing History

New editions of this manual will incorporate all material updated since the previous edition. Update packages may be issued between editions and contain replacement and additional pages to be merged into the manual by the user. Each updated page will be indicated by a revised date at the bottom of the page. A vertical bar in the margin indicates the changes on each page. Note that pages which are rearranged due to changes on a previous page are not considered revised.

The manual printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) The manual part number changes when extensive technical changes are incorporated.

September 1979 ... **PRELIMINARY EDITION** (ALIEN DROM)
February 1980 ... **FIRST EDITION** (MEDIA DROM)
January 1983 ... **SECOND EDITION**

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

CAUTION! Since tolerances on IBM 3740 compatible media are not fully specified by vendors and do vary, we cannot guarantee that IBM 3740 media can be read by the HP250 in every case. We have had excellent success in reading the IBM 3740 format at all available test sites.

Table of Contents

Chapter 1: Introduction

Chapter 2: Accessing Media Formats

Chapter 3: Utility Programs

CHAPTER 1

Introduction

The MEDIA DROM (available beginning with O.S. rev. 2.E) provides access to data stored on flexible discs and cartridge tapes which have a format different from the HP250 media format. Two disc formats can be accessed: IBM 3740 format 1/4 Mbyte flexible discs and HP interchange format flexible discs and tape cartridges. These two formats are discussed next.

The MEDIA DROM replaces the ALIEN DROM which was introduced with the preliminary version of this manual.

IBM 3740 Format

The IBM 3740 1/4 Mbyte flexible disc format has several features which differ from the HP250 format. The differences in the logical structure affect the typé of operations that can be performed on the IBM media.

Overall Structure

The IBM media consists of a single-sided flexible disc with 77 accessible tracks. Of the 77, there are 73 data tracks, 1 index track, 2 spare tracks, and 1 track is unused. Each track contains 26 128-byte sectors. Thus, the total data area is 26*73=1898 sectors or 242,944 bytes. The index track has 7 sectors reserved for volume-specific information and the remaining 19 sectors are reserved for directory information. Because there is one directory entry per sector, the IBM media supports a maximum of 19 files. A maximum of 2 tracks (neither one being the index track) can be spared. The figure below shows the overall structure of the IBM media.

Overall Structure of IBM Media

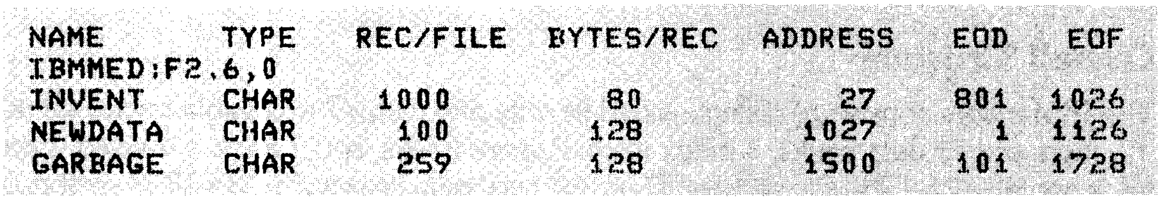
TRACK(S)	SECTOR(S)	CONTENTS
0	1-7	Volume Identification Info
0	8-26	Directory Records
1-73	27-1924	Data Area
74-75	1925-1976	Spare Tracks
76	1977-2002	Unused

Directory Record Structure

Proper terminology for IBM media is to refer to a file as a data set and to a directory record as a data set header. One record (128 bytes) is reserved for each data set header. The information in the data set header is:

- DATA SET NAME - An 8-byte field containing the name of the data set.
- LOGICAL RECORD LENGTH - A 3-byte field containing the record length of the data set. Normally, this is "080", however, it may be anywhere from "001" to "128" for some IBM machines (such as 3742 with the "128" feature).
- BEGINNING OF EXTENT (BOE) - A 5-byte field, coded into a head/track/sector format, that identifies the first data record of the data set.
- END OF EXTENT (EOE) - A 5-byte field, coded like BOE, that identifies the last sector of the area reserved for the data set.
- END OF DATA (EOD) - A 5-byte field, coded like BOE, that identifies the next unused sector of the data set.

Using the above information, MEDIA will display a catalog like the one shown below:



```
NAME       TYPE   REC/FILE  BYTES/REC  ADDRESS   EOD   EOF
IBMMED:F2,6,0
INVENT     CHAR    1000        80          27   801  1026
NEWDATA    CHAR     100       128        1027     1  1126
GARBAGE    CHAR     259       128        1500   101  1728
```

Sample Catalog of IBM Media

Notice the differences in this catalog from the normal HP250 catalog. There is no provision for a protect key because MEDIA will not support protect keys on IBM media. All data sets are considered unprotected. The type column will always contain "CHAR". Records/file, bytes/record, and data set starting address are displayed like the HP250 catalog. EOD indicates the next available data record in the file. Note that EOD has nothing to do with the physical end-of-file. For a file that has no data in it, the EOD value will be 1. The EOD value corresponds to the EOF marker that is written in a HP250 DATA file. EOF is the last record of the file and can be computed by adding the number of records in the file to the starting address and subtracting 1. EOF is displayed for convenience.

In the example catalog, there are three data sets on the disc. The volume label is IBMMED (maximum size of a volume label on IBM media is 6 characters). The data set INVENT starts at record 27 (the first record available for data sets). INVENT has 1000 records of 80 bytes reserved for it, but has only used (at most) 800[1] of those records. The data set NEWDATA has not yet been written to because the EOD marker is at the start of the data set. NEWDATA has records of 128 bytes. A data set called GARBAGE occupies 259 records on the media.

[1] There are at most 800 records because some of the records may be deleted in the data set. There is no way to know in advance how many deleted records a data set contains.

Deleted Records

A unique attribute of the IBM media is the manner in which records of a data set are deleted from the data set. A 3740 operator, for example, may discover that two records are identical for some reason and that one of the records should be deleted from the data set. If punching cards, the operator could simply remove the duplicate card. The corresponding 3740 operation is to delete the invalid record. When the operator deletes the record, the record is marked in a way that tells the machine to ignore the record and move on to the next one when reading the data. MEDIA will skip any records with this special mark in it.

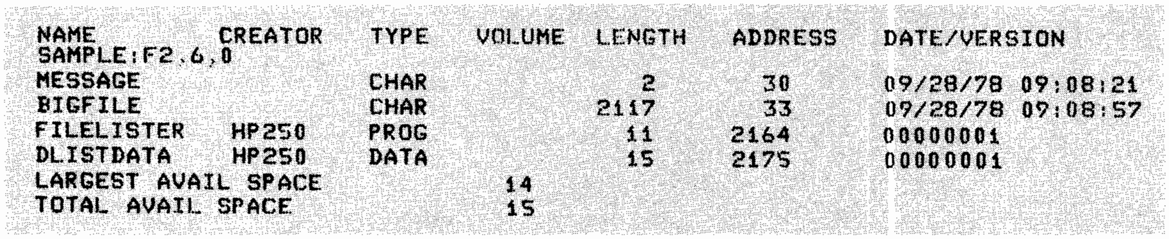
Features Not Used

As noted earlier, the IBM media have a very large area reserved for data set headers. The data set header contains some information that MEDIA does not use. Refer to Appendix C for a listing of the information not used.

HP Interchange Format

The HP interchange format can be used by HP250s to exchange data with any other HP computer which supports the format. Files of any type can be stored on this media by one computer and then shared with another.

There are some differences between files stored in the HP250 format and those stored with the interchange format. All files stored with the interchange format must have a record length of 256 bytes; protect codes are not allowed. Run-only programs and data sets cannot be stored. The catalog of the media also has some differences.



```
NAME          CREATOR   TYPE   VOLUME  LENGTH   ADDRESS   DATE/VERSION
SAMPLE:F2.6,0
MESSAGE                 CHAR              2        30     09/28/78 09:08:21
BIGFILE                 CHAR           2117        33     09/28/78 09:08:57
FILELISTER     HP250    PROG             11      2164     00000001
DLISTDATA      HP250    DATA             15      2175     00000001
LARGEST AVAIL SPACE             14
TOTAL AVAIL SPACE               15
```

Interchange Format Catalog

File names are limited to 10 characters consisting of upper case letters and digits. The first character must be a letter. Volume names are limited to 6 characters beginning with an upper case letter and consisting of upper case letters and digits.

The file creator is shown for file types other than CHAR. The file creation date and time is also listed for creators other than the HP250.

Care must be taken when using the CATBIN binary, or when doing CATs to the screen and then reads the lines via LENTER. Such a program must take into account the varying format of the catalog from media to media.

The HP250 does not provide any special facilities to deal with multi-volume CHAR files. The user application program may ASSIGN any volume on the file. The system treats each volume of that file as though it were an independent file.

CHAPTER 2

Accessing Media Formats

CHAR Files

Since the two media currently supported by the MEDIA DROM do not use HP250 format data files as their standard method for storing data, a special file type called a CHAR file has been defined to support data transfer from these media. A CHAR file is a file consisting of character data. It resembles most closely the display format files of the HP3000 and HP300. Each record consists only of characters. The length of a record may either be defined in the record itself or may be fixed for a particular file. Whether records have fixed or variable length is a function of the media on which they are created. CHAR files on IBM format flexible discs are composed of fixed length records. On the interchange media and the standard HP250 media they are variable length. For variable length records there is an overhead of 2 bytes per record.

CHAR files can be used on HP250 format media when the MEDIA DROM is loaded. The internal format of CHAR files on HP250 formatted media and HP Interchange formatted media is different from the internal format on the IBM formatted media. Because of this internal difference, COPY cannot be used to transfer files between IBM and HP formatted media.

> NOTE
>
> In order to access a cartridge tape which has been initialized in HP interchange format, the system must be in DIRECT mode. This means the tape will be accessed directly instead of being buffered through the disc. Simply type in DIRECT to select this mode of operation. Be certain to type in INDIRECT before inserting a different tape.

Creating a CHAR File

Use the CREATE command to create files of type CHAR. If the create command is suffixed with the sequence ;CHAR, the file is created as a character file. On an IBM media, there is also a file start address which must be specified.

To create a CHAR file on an IBM format disc, use the following syntax:

```
CREATE file spec , record count , record size , start address ;CHAR
```

The record size is a record length between 1 and 128 bytes. Other values are not allowed because of restrictions in the definition of this disc format. The start address parameter is the sector address of the start of the file, and record count is the number of records allocated. On an IBM floppy, each record consumes one sector, unused bytes in the sector are ignored. Also, sectors on an IBM media are 128 bytes long rather than 256.

To create a CHAR file on an interchange format media, or a standard HP250 media, use the following form:

CREATE file spec , record count ; CHAR

Here, record count refers to the number of 256 byte sectors to be allocated to the CHAR file. If a record length parameter is specified, an error will result. If a file start address is specified, an error will be given. The actual number of records that can be put into this kind of CHAR file is dependent on the length of the records. Each record can be from 0 thru 32767 bytes long. A record of length L bytes will take up (L+3)/2 words in the file. An EOF mark is written as the first record of the file.

The FCREATE command functions exactly like CREATE on the HP interchange format and standard HP250 format. On the IBM format, FCREATE will force the end-of-data marker to point at the end of the file. A CHAR file is never initialized with data, regardless of which create command is used.

Assigning a CHAR File

The normal ASSIGN command is used to ASSIGN a CHAR file. In the case of IBM format discs, the class list parameter can additionally contain the keywords EBCDIC or EBCDIK. Suffixing the ASSIGN command with EBCDIC will tell the system that data read from or written to that file should be translated from ASCII to EBCDIC. For Katakana translation, EBCDIK should be used.

The following statement will assign #1 to file "FILE" in update mode and do EBCDIC/ASCII conversion:

```
ASSIGN #1 TO "FILE";UPDATE,EBCDIC
```

The following statement will assign #6 to file "BIG". Since EBCDIC was not specified, the file is assumed to be ASCII.

```
ASSIGN "BIG" TO #6
```

The following statement will Assign #1 to file "LONGNAME". This example shows the use of a file name longer than six characters. HP Interchange Format media can have file names up to 10 characters long.

```
ASSIGN "LONGNAME" TO #1
```

The end-of-file marker for an IBM data set is always kept in the directory record for the data set. Since it is unreasonable to write to the directory of the IBM disc every time a PRINT # occurs past the current end of the file, the end-of-file marker is only updated when the file is de-assigned. If the disc is removed from the drive before the file is de-assigned, then the end-of-file marker may be incorrect.

Printing to a CHAR File

Printing to a CHAR file is similar to printing to a printer. The spacing functions (TAB, SPA, LIN and PAGE) can be induded in the PRINT # command.

$$\texttt{PRINT \#}\ \text{file number [, record number [, byte pointer]] ;} \begin{Bmatrix} \texttt{END} \\ \text{print list [; } \texttt{END} \text{]} \end{Bmatrix}$$

The record pointer moves the file pointer to any record in the file. The byte pointer parameter is treated as a column position parameter.

When a line is printed to a CHAR file, a CRLF (carriage return, linefeed) advances the print pointer to the next record in the file. Since CHAR files resemble printers, the file has a pseudo-printer width associated with it as well which may generate CRLFs. For an IBM CHAR file this width is the fixed record length specified in the CREATE. For interchange and standard media, this width is 32767.

If only a LF is received, the print pointer advances to the next record of the CHAR file, but the pointer is not set to the beginning of a new record.It is left at the same column position and preceding bytes in the record are blank filled. If only a CR is received, it is treated as a CRLF.

Since CR and LF are interpreted as control characters by PRINT #, it is impossible to write a CR (CHR$(13)) or an LF (CHR$(10)) to an HP250 or HP Interchange CHAR file. A CR or LF can be written to an IBM file only by using a special sector dump statement, IBMWREC.

Since it cannot be determined at syntax time if a particular print list is valid for a particular file, an error is issued if an attempt is made to print a TAB, LIN, SPA or PAGE to a non-CHAR file. An error is also issued if the " ; " or multiple commas are used as a separator when printing to a non-CHAR file.

There are some differences between the two types of CHAR files. Both may be accessed randomly via the record pointer. Because the interchange format CHAR file has variable length records, the only way to find a record is via a serial read. The IBM type CHAR file can calculate the disc address for any record. Secondly, an implicit EOF is written after a PRINT # to an interchange type CHAR file. Thus, it is not possible to modify data in the middle of such a file without losing all subsequent data. On the IBM type CHAR file, it is possible to modify any record without disturbing subsequent data.

Reading Data From a CHAR File

Since data is written on a CHAR file as a printer would print rather than in internal format, the READ # statement is not adequate for extracting such data from a CHAR file. The LINPUT # statement was introduced to allow such data to be read. It has the following syntax:

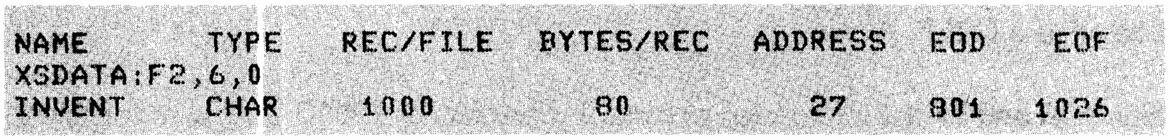
LINPUT # file number [, record number [, column position]] ; string variable

The entire record starting at the specified column in the record is read into the string variable. Since an IBM format CHAR file has fixed length records, data read from such a file will always result in a string with a record length matching the record length of the file. The length of the string read from an interchange format CHAR will match exactly the record length of the current record.

Sample Application

A programmer has just received a flexible disc from a local keypunch service. It contains inventory records of his company. The data on the disc is in an IBM 3740 format which can be read with the MEDIA DROM.

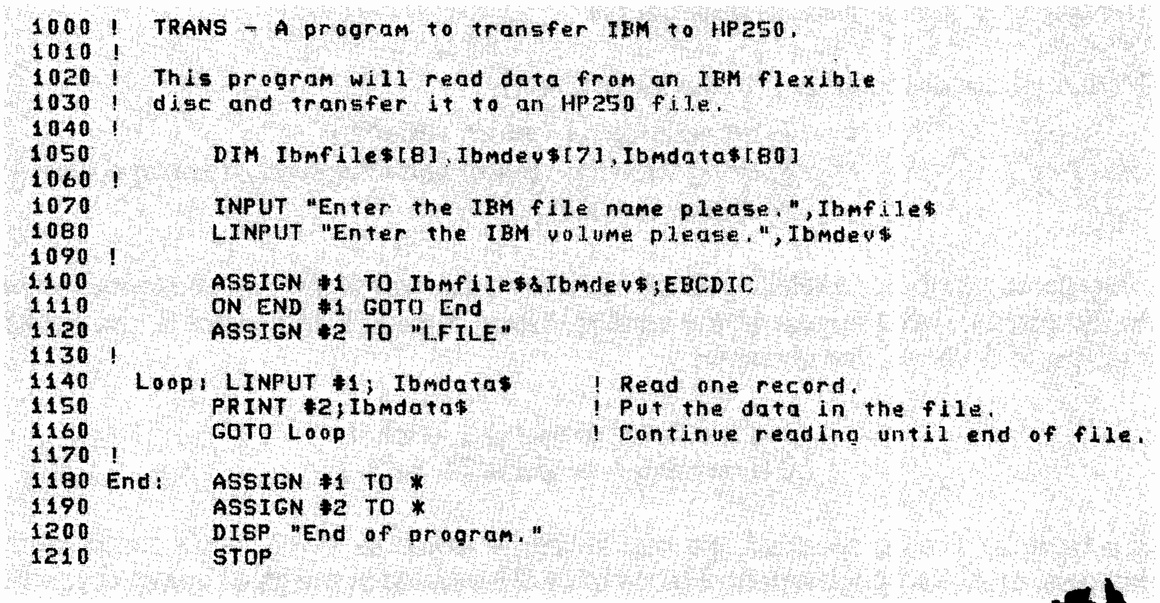
The following catalog is displayed:



```
NAME        TYPE    REC/FILE   BYTES/REC   ADDRESS   EOD    EOF
XSDATA:F2,6,0
INVENT      CHAR      1000         80          27    801   1026
```

The only data set on the disc is one called INVENT. The data set consists of 80-byte records and 800 records have been keypunched (EOD is 801), although space was reserved for 1000 records at the time the data set was created.

The program shown next transfers the data to an HP250 disc. Note that the ASSIGN statement in line 1100 has the EBCDIC option included.



```
1000 !   TRANS - A program to transfer IBM to HP250.
1010 !
1020 !   This program will read data from an IBM flexible
1030 !   disc and transfer it to an HP250 file.
1040 !
1050          DIM Ibmfile$[8],Ibmdev$[7],Ibmdata$[80]
1060 !
1070          INPUT "Enter the IBM file name please.",Ibmfile$
1080          LINPUT "Enter the IBM volume please.",Ibmdev$
1090 !
1100          ASSIGN #1 TO Ibmfile$&Ibmdev$;EBCDIC
1110          ON END #1 GOTO End
1120          ASSIGN #2 TO "LFILE"
1130 !
1140    Loop: LINPUT #1; Ibmdata$         ! Read one record.
1150          PRINT #2;Ibmdata$           ! Put the data in the file.
1160          GOTO Loop                   ! Continue reading until end of file.
1170 !
1180 End:     ASSIGN #1 TO *
1190          ASSIGN #2 TO *
1200          DISP "End of program."
1210          STOP
```

Delete Statement

There is no way to delete a record from a data set using PRINT #. For this reason, the DELETE # statement has been added to the MEDIA DROM. The syntax is:

DELETE #file number ; record number

The file number is the number used in the ASSIGN statement. The record number is the number of the record in the data set to be deleted. The record specified will be marked deleted with no change in the data except that a D will be added as the first character. There is no easy way to un-delete a record that has been previously deleted.

The DELETE # statement will give an error if you try to delete a record that is outside of the file space or on a media other than the IBM format media.

IBMDUMP Statement

The IBMDUMP statement can be used to bypass normal file system formatting, overhead, and deficiencies. With the IBMDUMP statement, you can read any sector on the IBM disc in one of the following three ways:

- Read a sector into a string variable,
- Read a sector into a numeric array,
- Display a sector on the CRT in different formats.

The syntax of the IBMDUMP statement is:

```
IBMDUMP record number [ON device specifier]
                     ⎧ ,numeric array[ ,return variable]                                   ⎫
                     ⎨ ,string variable[ ,return variable[ ;conversion specifier] ]        ⎬
                     ⎩ [;display specifier]                                                ⎭
```

The record number is a relative sector offset from the start of the disc and corresponds to the numbers that appear in the catalog. Given a track T and sector S, the record number is T*26+S. For example:

1 is translated to sector 1, track 0
50 is translated to sector 24, track 1

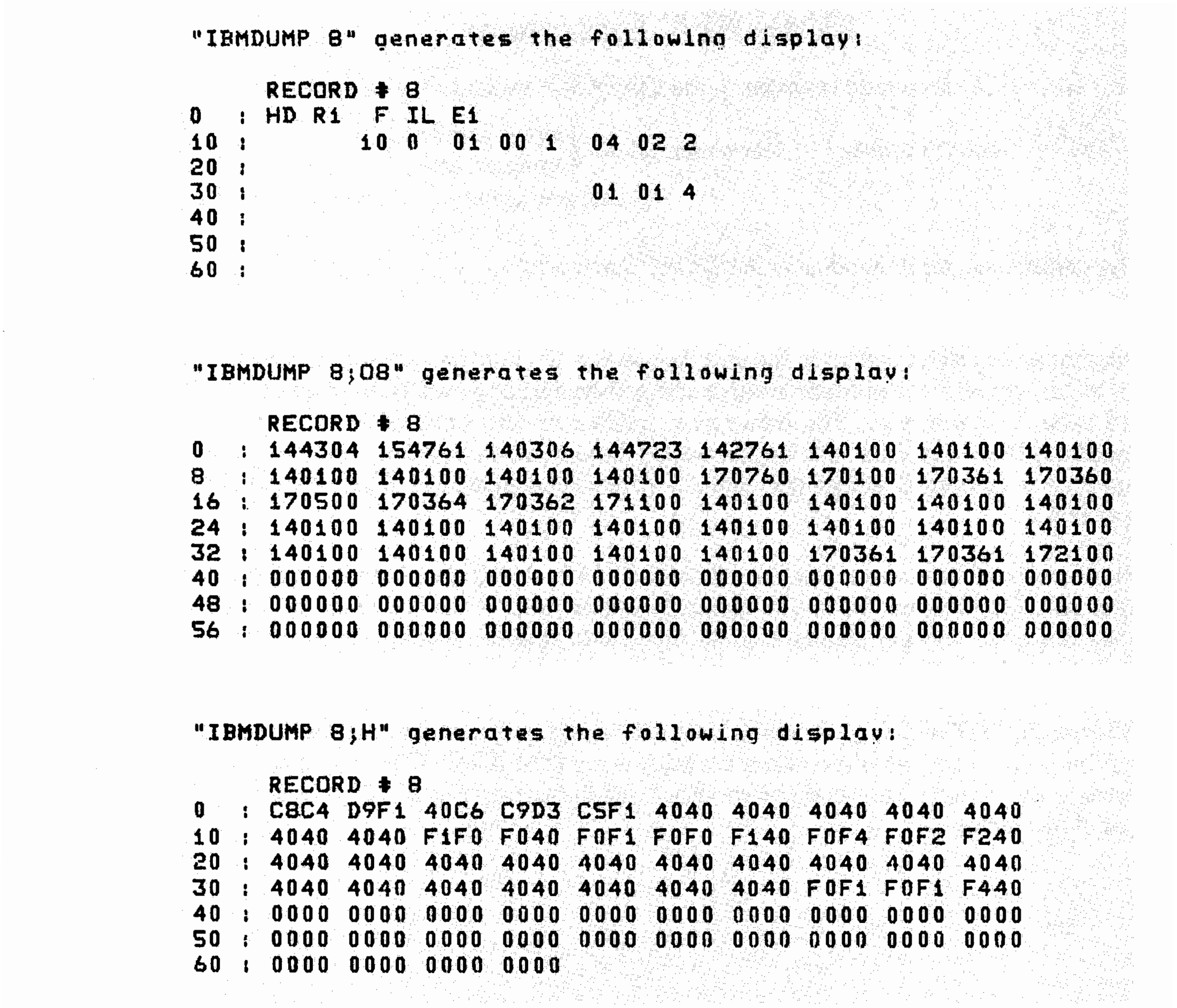
If a numeric array is specified, the record will be stored as 16-bit, two's complement integers, one word per element The record will be stored in the given string, character for character, if a string is specified. If neither a string or a numeric array is specified, then the data will be displayed on the CRT. The user has a choice of display formats. The display specifier is a letter followed by a number. The letter indicates what conversion should be performed on the data and the digit following it indicates how many words per row should be displayed. The letters stand for the following conversions:

E - EBCDIC to ASCII conversion, display in ASCII
A - No conversion, display in ASCII
O - No conversion, display in octal
H - No conversion, display in hexadecimal
D - No conversion, display in decimal
K - EBCDIK (Katakana) to ASCII conversion, display in ASCII

The default display format is E10. If a letter is not followed by a number, 10 is used. The next figure shows some sample sector dumps to the screen.

If the data is going to a string, an E (EBCDIC to ASCII conversion), A (no conversion), or K (EBCDIK to ASCII) conversion specifier is optional. The default is E. If a conversion specifier is given for a numeric array, it will be ignored.

The return variable can be any simple numeric variable. It is used only for deleted records. If the specified record is deleted, the return variable is made non-zero, otherwise it is forced to zero. If no return variable is given, then an error 373 will be issued whenever a deleted record is read. The error 373 is generated AFTER all other processing is complete, thus the data from the deleted record is still valid.



```
"IBMDUMP 8" generates the following display:

     RECORD # 8
0  : HD R1  F IL E1
10 :       10 0  01 00 1  04 02 2
20 :
30 :                      01 01 4
40 :
50 :
60 :
```

```
"IBMDUMP 8;O8" generates the following display:

     RECORD # 8
0  : 144304 154761 140306 144723 142761 140100 140100 140100
8  : 140100 140100 140100 140100 170760 170100 170361 170360
16 : 170500 170364 170362 171100 140100 140100 140100 140100
24 : 140100 140100 140100 140100 140100 140100 140100 140100
32 : 140100 140100 140100 140100 140100 170361 170361 172100
40 : 000000 000000 000000 000000 000000 000000 000000 000000
48 : 000000 000000 000000 000000 000000 000000 000000 000000
56 : 000000 000000 000000 000000 000000 000000 000000 000000
```

```
"IBMDUMP 8;H" generates the following display:

     RECORD # 8
0  : C8C4 D9F1 40C6 C9D3 C5F1 4040 4040 4040 4040 4040
10 : 4040 4040 F1F0 F040 F0F1 F0F0 F140 F0F4 F0F2 F240
20 : 4040 4040 4040 4040 4040 4040 4040 4040 4040 4040
30 : 4040 4040 4040 4040 4040 4040 4040 F0F1 F0F1 F440
40 : 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
50 : 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
60 : 0000 0000 0000 0000
```

Sample Display Formats of IBMDUMP Statement.

IBMWREC Statement

The IBMWREC statement is similar to the IBMDUMP statement. The syntax is:

```
                                             { string variable }
IBMWREC record number [ON device specifier], { numeric array   }
                                      [, numeric variable[;conversion specifier] ]
```

The parameters are defined as in IBMDUMP statement.

A numeric array or string variable must be included with this statement. The data in the string or array will be written to the specified sector. If a numeric variable is included in the statement after the numeric array or string, then the sector will be marked deleted if the variable is non-zero. The conversion specifier is also optional. If conversion is specified, then the data will be converted before being written. The conversion specifier is ignored if a numeric array is given. The default is ASCII to EBCDIC conversion.

If the length of the numeric array is greater than 64 or the length of the string is greater than 128, only the first 64 or 128 elements will be used. If the source array or string does not have enough elements to fill one sector, the remainder of the sector will contain random garbage.

When using IBMWREC to modify a data set header, for example, the sector should first be read into a string (or numeric array) using IBMDUMP. This way the byte or bytes to be changed can easily be modified without generating all of the data for the rest of the sector.

CHAPTER 3

Utility Programs

IMEDIA Program

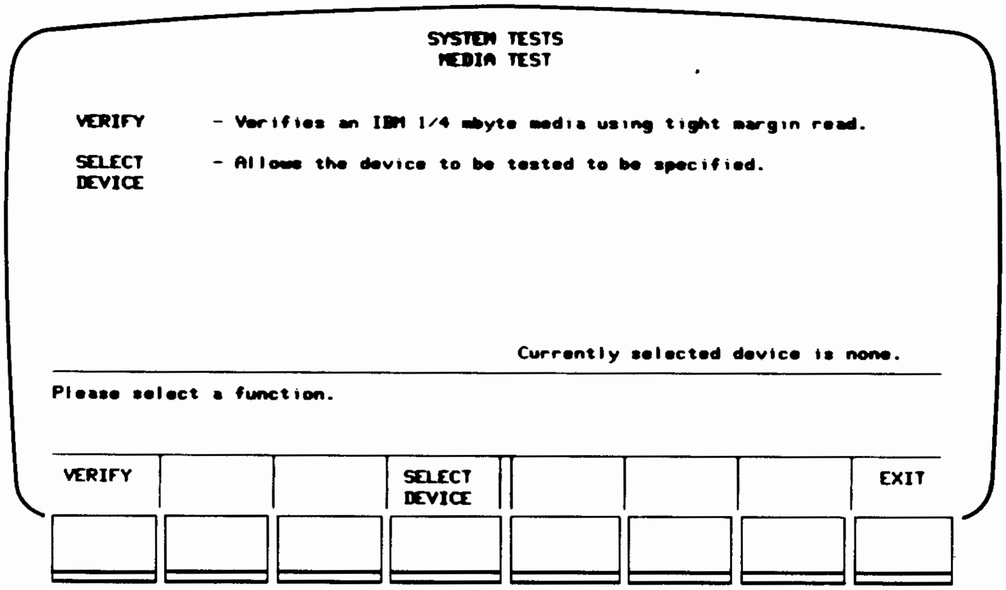
The IMEDIA program is a part of the system TEST utility. IMEDIA tests an IBM formatted flexible disc to determine if it is readable by the HP250. Before transferring large amounts of data from another machine to the HP250 on an IBM formatted flexible disc, a test file should be made and then tested with IMEDIA to ensure its readability.

IMEDIA tests the flexible disc using severe tolerances. If a flexible disc can be read with these tolerances, there should be no problems reading from the flexible disc in normal operations. If sectors fail the test, they still may be readable in normal operation. The IMEDIA program serves only as a warning method for possible problems.

The IMEDIA program is a part of the system TEST utility on the SYSTEM disc. To run IMEDIA, insert the SYSTEM disc into the default drive and execute:

```
RUN "TEST"
```

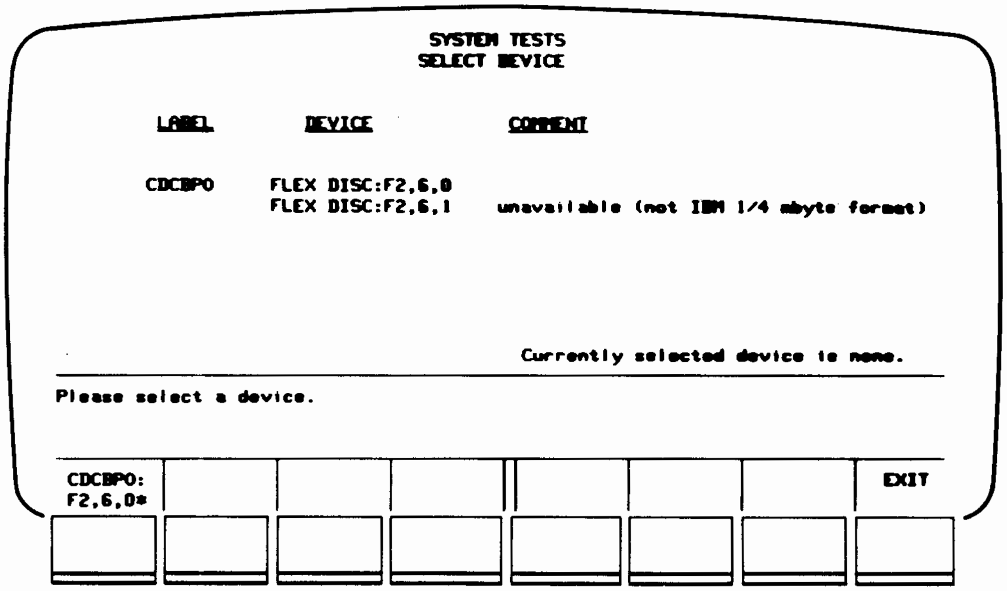
Next, press softkey IBM MEDIA TEST.



```
                         SYSTEM TESTS
                          MEDIA TEST

VERIFY     - Verifies an IBM 1/4 mbyte media using tight margin read.

SELECT     - Allows the device to be tested to be specified.
DEVICE




                              Currently selected device is none.
Please select a function.
```

If the MEDIA DROM is not loaded, the message `MEDIA DROM not loaded. Please press EXIT` is displayed. Use the CONFIG program to load MEDIA.

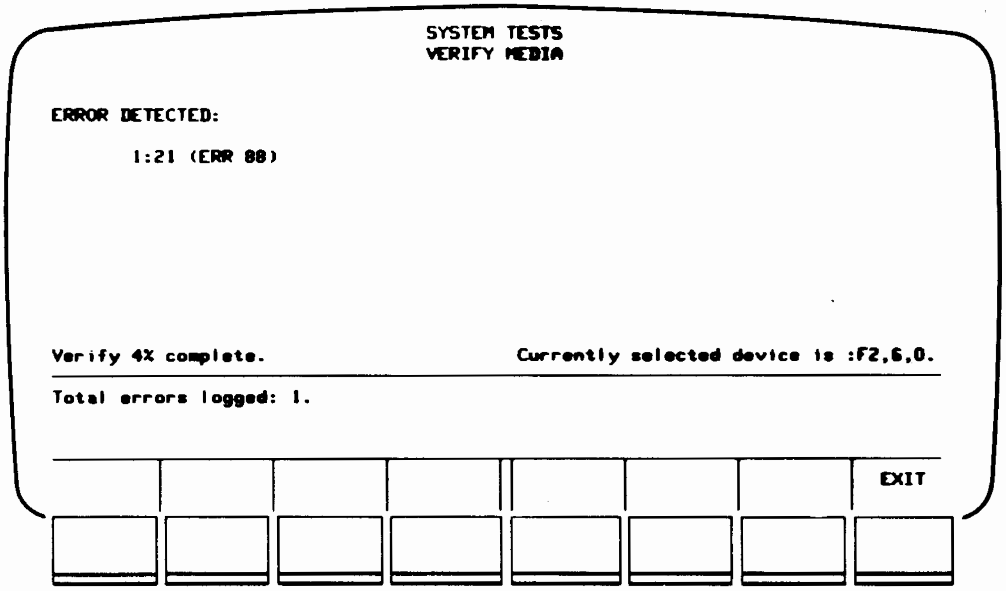
If an IBM media is currently in a disc drive, that device will be selected by the program. If no IBM media is loaded, press SELECT DEVICE. Insert the media to be tested and press the softkey specifying that drive.



```
                              SYSTEM TESTS
                              SELECT DEVICE

      LABEL        DEVICE              COMMENT

      CDCBPO       FLEX DISC:F2,6,0
                   FLEX DISC:F2,6,1    unavailable (not IBM 1/4 mbyte format)



                                   Currently selected device is none.
Please select a device.
```

After you have selected the device, the first screen is displayed. Press VERIFY to begin the testing.

If an error is detected, the track sector and error number is displayed.



```
                              SYSTEM TESTS
                              VERIFY MEDIA

ERROR DETECTED:

      1:21 (ERR 88)




Verify 4% complete.                 Currently selected device is :F2,6,0.
Total errors logged: 1.
```

If an error other than ERR 88 is found, the flexible disc is probably unusable. An ERR 88 means the sector is unreadable with the severe tolerance but may be readable under normal conditions. ERR 88 is a warning that the flexible disc itself and not the data is possibly bad.

If no errors are reported, the flexible disc can be used in the HP250 with a low chance of errors occurring in normal operations.

When the flexible disc has been tested, press EXIT to return to the previous screen. You can test another flexible disc or press EXIT to leave the IMEDIA portion of the TEST utility.

INIT Program

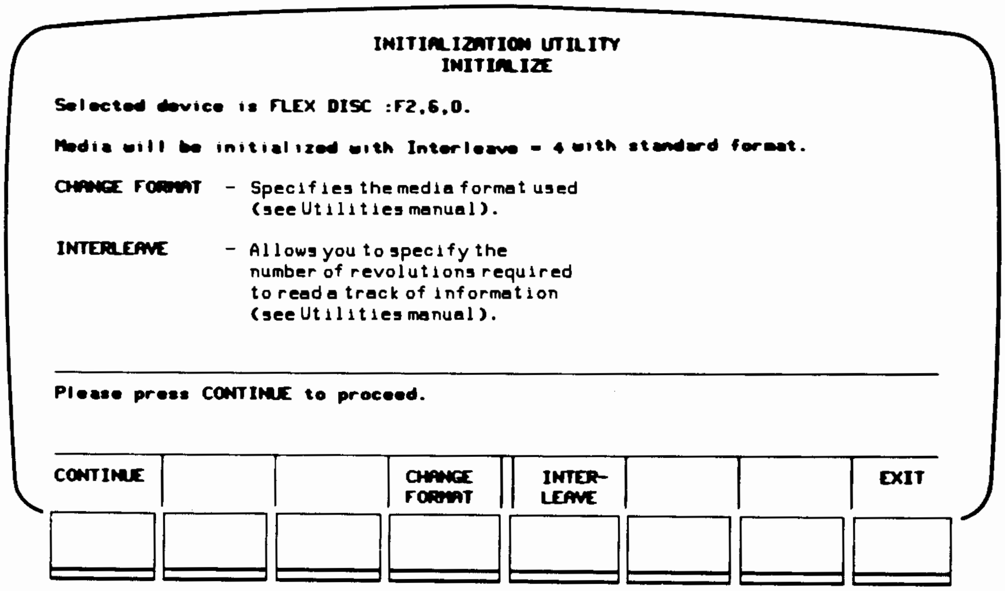
The Initialize program is located on the SYSTEM disc or tape cartridge and is fully documented in the HP250 Utilities Manual. The INIT program includes an option to initialize a flexible disc in the interchange format or IBM format, or to initialize a cartridge tape in the HP interchange format.

Physically, the HP Interchange format is the same as the standard HP250 format. Only the directory structure is different. The IBM format is physically different.

NOTE

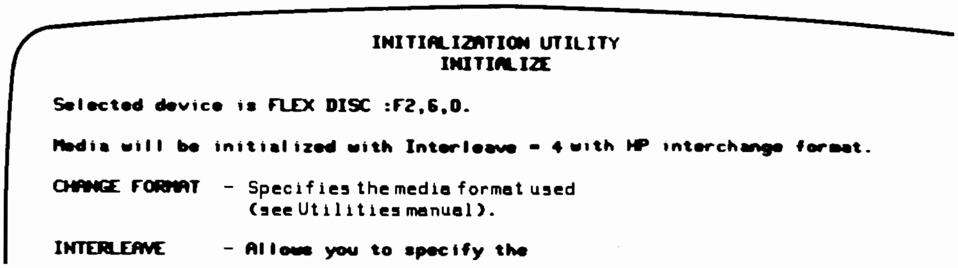
> Before a cartridge tape can be initialized in the HP interchange format, the system must be in DIRECT mode. Before running INIT, enter DIRECT. Enter INDIRECT before you insert another tape into the drive.

To select an alternate format for initialization, press the CHANGE FORMAT softkey.



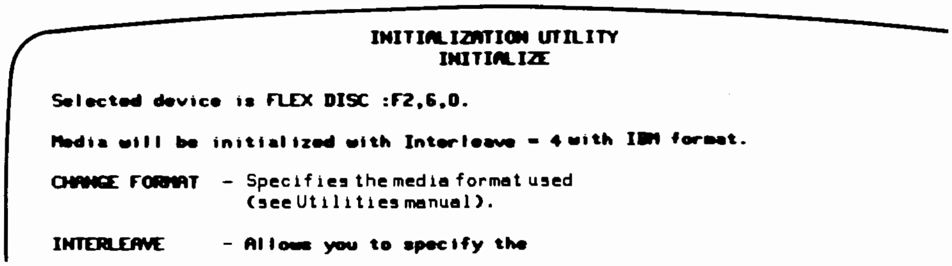
```
                         INITIALIZATION UTILITY
                               INITIALIZE

Selected device is FLEX DISC :F2.6.0.

Media will be initialized with Interleave = 4 with standard format.

CHANGE FORMAT   - Specifies the media format used
                  (see Utilities manual).

INTERLEAVE      - Allows you to specify the
                  number of revolutions required
                  to read a track of information
                  (see Utilities manual).

Please press CONTINUE to proceed.
```

The interchange format is selected first



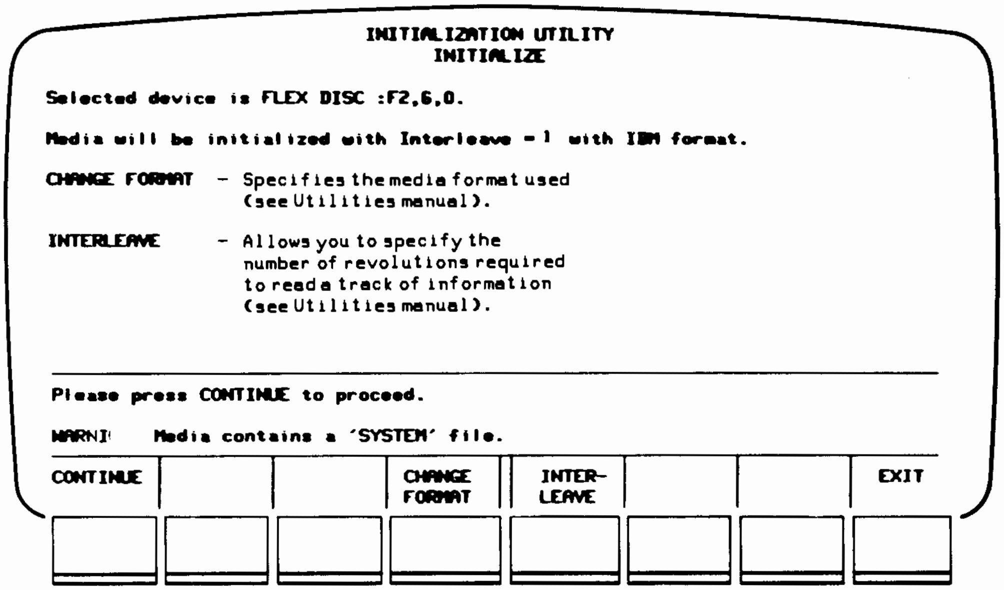
```
                          INITIALIZATION UTILITY
                                INITIALIZE

Selected device is FLEX DISC :F2,6,0.

Media will be initialized with Interleave = 4 with HP interchange format.

CHANGE FORMAT   - Specifies the media format used
                  (see Utilities manual).

INTERLEAVE      - Allows you to specify the
```

The IBM format is selected next



```
                          INITIALIZATION UTILITY
                                INITIALIZE

Selected device is FLEX DISC :F2,6,0.

Media will be initialized with Interleave = 4 with IBM format.

CHANGE FORMAT   - Specifies the media format used
                  (see Utilities manual).

INTERLEAVE      - Allows you to specify the
```

Use interleave of 1 when formatting IBM single-sided diskettes.
Press INTERLEAVE to change the interleave number.



```
                          INITIALIZATION UTILITY
                                INITIALIZE

Selected device is FLEX DISC :F2,6,0.

Media will be initialized with Interleave = 1  with IBM format.

CHANGE FORMAT   - Specifies the media format used
                  (see Utilities manual).

INTERLEAVE      - Allows you to specify the
                  number of revolutions required
                  to read a track of information
                  (see Utilities manual).

Please press CONTINUE to proceed.

WARNI    Media contains a 'SYSTEM' file.
```

Press CONTINUE to begin initializing the media.

The IBM format can accomodate only 2 bad tracks. Also, when initialization is complete, a file, DATA08, is created on the IBM flexible disc. This is the way purchased pre-initialized flexible discs are formatted.

APPENDIX A

Syntax Reference

```
ASSIGN { file spec TO # file number }[, return variable[, protect code][; class list]]
       { # file number TO file spec }
```

Opens a file. The class list parameter can contain the keywords EBCDIC or EBCDIK when data needs to be translated from ASCII to EBCDIC or for Katakana translations, EBCDIK.

```
CREATE file spec , record count , record size , start address ; CHAR
```

Creates a CHAR file on an IBM media.

```
CREATE file spec , record count ; CHAR
```

Creates a CHAR file on an HP250 or interchange format media.

```
DELETE # file number , record number
```

Deletes a record from a CHAR file.

```
IBMDUMP record number [ON device specifier]
                   { , numeric array [ , return variable]                                }
                   { , string variable [ , return variable [ ; conversion specifier]]   }
                   { ; display specifier                                                 }
```

Dumps a sector of an IBM disc into a string variable, into a numeric array or onto the CRT.

```
IBMWREC record number [ON device specifier] , { string variable }
                                              { numeric array   }
                        [ , numeric variable [ ; conversion specifier]]
```

Writes a sector of data to an IBM disc.

```
LINPUT # file number [ , record number [ , column position]] ; string variable
```

Reads data from a CHAR file into the string variable.

```
                                                  { END               }
PRINT # file number [ , record number [ , byte pointer]] ; { data list [ , END]  }
                                                  { print list [ , END] }
```

Prints data to a CHAR file. Spacing functions (TAB, SPA, LIN and PAGE) can be included.

APPENDIX B

Error Messages

General MEDIA Errors

340 Operation only allowed on IBM media.

341 Improper operation on CHAR file.

342 Operation not allowed on this media.

343 Invalid IBM data set record length.

344 File on IBM media must be type CHAR.

345 Invalid IBM file start address in CREATE command.

346 Cartridge tape in HP interchange format cannot be accessed while in INDIRECT mode.

IBMDUMP and IBMWREC Errors

370 Record number out of range for IBM media.

371 Device does not contain IBM format media.

372 Invalid display or conversion parameter.

373 Deleted record read.

APPENDIX C

IBM Sector Formats

The following table shows the format of sector 7. This sector is the volume identification sector and is always in EBCDIC format.

Start Byte	Length	Description
0	3	Volume label identifier VOL
3	1	Volume label number 1
4	6	Volume label
10	1	Accessibility indicator (blank indicates accessible)
11	26	Reserved
37	14	Owner identification field
51	24	Reserved
75	1	Physical record length indicator blank=128 byte sector
76	2	Physical sector sequence code (also known as interleave) blank or 01 thru 13
78	1	Reserved
79	1	Label standard version (must be a W)
80	48	Padding (binary zeros)

Sector 5 is an error map sector and is always in EBCDIC. The following format is usually used:

Start Byte	Length	Description
0	5	ERMAP
5	1	blank
6	2	If any bad tracks on disc, then this has the first bad track number. Otherwise, blank
8	1	Binary zero
9	1	blank
10	2	If there are two bad tracks, then this has the second bad track number. Otherwise, blank
12	1	Binary zero
13	67	Blanks
80	48	Binary zeros

The following table shows the format of sectors 8 thru 26. These sectors contain the data set headers, one per sector. An empty entry is deleted and has a D as the first character.

Used by MEDIA	Start Byte	Length	Description
★	0	3	Label identifier HDR (DDR for deleted data set)
★	3	1	Label number, always 1
	4	1	Reserved
★	5	8	Data set name
	13	4	Reserved
	17	5	Unused
★	22	5	Record length
	27	1	Record attribute

★	28	5	Beginning of extent, coded as CCHSS where:
			CC=cylinder (track)
			H=head (always 0)
			SS=sector

	33	1	Physical sector length indicator 0=128 byte sector
★	34	5	End of extent (CCHSS)
	39	1	Reserved
	40	1	Bypass indicator (blank means process data set)
	41	1	Data set security (non-blank indicates secured)
	42	1	Write protect file indicator: P=Read only

	43	1	Exchange type indicator:
			blank=basic exchange file
			E=system file

	44	1	Multi-volume indicator: blank =file complete on this volume C=continued on another volume L=last volume of multivolume file
	45	2	Multivolume sequence number
	47	6	Creation date (usually YYMMDD)
	53	7	Reserved
	60	5	Offset to next record space
	65	1	Reserved
	66	6	Expiration data (usually YYMMDD)

	72	1	Verify mark:
			blank =not verified
			V=verified

	73	1	Data set organization
★	74	5	End of data (CCHSS)
	79	1	Reserved
	80	48	Padding (binary zeros)