Gemini
Computer Systems Limited

M-F-B 2

SOFTWARE MANUAL

ISSUE 5

01-12-85

TABLE OF CONTENTS

Appendices

1. Introduction

The Multi-Format-BIOS (MFB) software is an extension of the normal Gemini CP/M BIOS, and as such supports all the usual enhanced features such as disk error trapping, screen edit, screen dump etc. However the MFB will support a variety of CP/M disk formats and disk sizes simultaneously, and provides the ideal way of copying CP/M disk files between dissimilar formats.

The BIOS is table-driven, there being a table entry for each logical drive on the system. These tables define the physical characteristics of each drive (size, tracks, step rate etc), together with the format to be used. To reconfigure the BIOS for alternative formats only these tables need to be changed.

The BIOS will support most current soft-sectored formats, but cannot handle those formats that depend on unique hardware features. (See Appendix A).

To support the MFB five programs (SETUP, SET, FORMAT, WHIG, and ANALYSE) and a data file (FORMATS.DAT) are supplied:

FORMATS.DAT holds the details of disk formats - method of access, reserved tracks, number of directory entries, disk block allocation size etc. The three main utility programs all pick up their data from this file.

SETUP is the main utility program and is used to add new formats to FORMATS.DAT, and to set up a different configuration of formats within the BIOS. This is the program that directly patches the drive and format tables within the BIOS, making system reconfiguration only a few seconds work.

SET is a stripped down version of SETUP that will configure a system to match data stored in a file. This provides an easy way of switching between several often used environments.

FORMAT is the utility program that will format disks to most of the formats in the data file. It also has a verification mode in which it will read an entire disk, reporting any errors and retries that occur during the process.

ANALYSE is provided to help determine the physical characteristics of an unknown format. When presented with a disk of unknown format, it will determine the recording density used, the number of tracks-per-inch, the number of sectors per track, the sector size, and a few other parameters relating to the physical format.

WHIG is a simple utility that displays the current format configuration of the running MFB software.

2. System Hardware

The MFB system is based on the Gemini Multiboard range of 80-BUS compatible computer boards, and in particular requires the GM849 disk controller card.

Currently the BIOS will support up to eight logical drives, plus a 'memory' drive. These logical drives utilise a Winchester drive and various floppy disk drives. The 'memory' drive may be either bank-switched memory, or a GM833 RAM DISK. Logical drive A is fixed in type as the Winchester drive and cannot be redefined, but the remaining seven logical drives may be set up to be a combination of extra partitions on the Winchester drive, and/or a variety of disk formats on the physical floppy drives.

A system could for example consist of the following physical drives:-

Miniscribe M3425 20Mbyte Winchester.
5.25" 96tpi double-sided floppy disk drive.
5.25" 48tpi double-sided floppy disk drive.
3.5" 135tpi double-sided floppy disk drive
8" 48tpi double-sided floppy disk drive.

The basic system could be setup to be:

Logical drive:	Format
A) B) C)	3 partitions of approx 7Mbytes on the Winchester
D E F G H	Any combination of formats from the format library that are supportable on the above list of hardware.
M	Memory drive (RAM disk) if fitted.

The format/drive combinations of drives D-H may be redefined at anytime by the utilities SET or SETUP.

NB If an MFB system is used on the Gemini MultiNet Network as a 'Superstation', then drives G and N will always be Network drives whilst the system is 'logged on'. ie The Network definition of drive G overides that of the MFB system.

3. System generation and BIOS customisation

The standard Gemini program GENSYS can be used to generate a new system from the supplied system files, and likewise CONFIG can be used to set up the usual features of printer support, baud rates etc. The enhanced features of the MFB are handled by the program SETUP and are described in the next section.

(See the standard system documentation for details on GENSYS and CONFIG).

NOTE: When an MFB system is generated by GENSYS it operates on the data in the .CFG file in a restricted manner. Starting with the first logical drive it accepts all Winchester partitions until it encounters an entry for a floppy format. This is forced to GEMQDDS on drive 0, and all subsequent definitions (if any) are ignored.

3.1. Normal Systems & MFB systems

Gemini Winchester-based systems support two system tracks. This allows users to switch between two alternate systems, the utility BOOT being provided to perform this switch. (BOOT 1 - to run the system on track 1, BOOT - to return to the system on track 0). In this instance it allows users to run a normal Winchester-based system as an alternative to the MFB system, and may be of interest to those who use the machine for other purposes than just copying disks.

While high capacity logical drives (e.g. 8Mbytes) are essential if large data-base files are to be handled, they can be an irritation if a large number of small files are the order of the day. The 'User Area' feature of CP/M provides one way of grouping files together according to usage and reducing the size of 'DIR' listings, but there are still penalties in the time it takes CP/M to re-login a drive following a ^C. This time is directly proportional to the maximum size of the directory of a drive. Therefore it may be more convenient to split the Winchester drive into a larger number of smaller logical drives, (e.g. a 20Mbyte drive into 5 logical drives of 4Mbytes each, rather than 2 logical drives of 8Mbytes, and one of 4Mbytes). However this will reduce the number of logical drives left for the floppy-based formats.

The way round this is to define a standard system with the partitioning required, but in the MFB system definition do not use the whole Winchester. (NB the Winchester partitions as defined in the MFB '.CFG' file **MUST** match those in the standard system as far as they go). This will give a working environment in which the whole Winchester is available, and disk-copying environment in which some of the Winchester is invisible.

e.g

Normal system	MFB system
4Mbytes of Winchester	4Mbytes of Winchester
4Mbytes of Winchester	4Mbytes of Winchester
4Mbytes of Winchester	4Mbytes of Winchester
4Mbytes of Winchester	)
4Mbytes of Winchester	) Any format from
GEMQDDS floppy	) FORMATS.DAT file
IBMPC floppy	)
IBM3740 8" single dens.	)

4. Using an MFB system

The system appears as a normal Gemini CP/M system, with the usual enhanced features such as on-screen editing and screen dump available. All CP/M programs can be used as usual, and the only difference is that the amount of memory available for user programs is slightly reduced due to the extra features incorporated into the BIOS. (The size reduction will be somewhere between 512 and 1024 bytes). If this small reduction in program memory area is found to be a problem with a particular application program, then a normal Gemini system can be run from the alternate system track - as described in the previous section. (Obviously in this case the wide range of alternative disk formats would not be accessible by the applications program).

The system configuration,(i.e. what logical drive is what format), can be changed quickly at any time by running the program SETUP and selecting option 3 (see section 5), or by running SET (see section 6). As the system is so flexible and can be changed so easily, the utility WHIG (What Have I Got?) is provided, so that you can see the current assignment of the logical drives.

IMPORTANT

One point to remember when using SETUP is that it performs two different functions. One is to maintain the format information in the data file FORMATS.DAT. The other is to use this information to patch the system to support those formats. So if you have created a format BLOGGS, and set up the system so that Drive C is BLOGGS format, and subsequently discover an error in your data entry for the format, you have to do two things. The first to use option 1 in SETUP to correct the BLOGGS entry in the data file, and the second is to use option 3 to set up the system again, this time using the correct data for the BLOGGS format. CHANGING THE DATA FILE DOES NOT ALTER ANY INFORMATION IN THE RUNNING SYSTEM. It is only when you use option 3 in SETUP that any changes occur.

5. SETUP

The program SETUP is used to enter the details of formats and drives to be handled by MFB. This data is stored in a disk file (FORMATS.DAT) which should be present on drive A whenever SETUP is run. The data file is restricted to drive A, because if SETUP is used to construct a new system configuration, all the pointers associated with the other drives may change, resulting in CP/M temporarily flagging them as Read/Only and thus preventing the save of an updated version of the file.

5.1. General

While SETUP is running, an index to the data file is held in memory, while the data itself resides on disk. The 'quick' list of formats displays just the directory of format names, whilst any other operation that requires the full data on a format will result in SETUP accessing the disk. When ever the database is edited, the changed information is immediately written to disk, thus protecting the database against corruption if SETUP is exited in an unorthodox manner.

5.1.1. Data input and defaults

Most command input to SETUP is obtained via a buffered line input routine. i.e. SETUP does not act on your reply until the RETURN key is pressed. In some instances (shown below) SETUP will display on the current line the default input value that it will assume if nothing is typed in except RETURN. Note that even if you only wish to change one character of the value displayed, the correct value must be TYPED IN FULL as this method of input does not support on-screen editing. For confirmation SETUP redisplays the value it has accepted when RETURN is pressed.

For some fields it is possible to enter more characters than are expected. The extra characters are ignored by SETUP, and the only side-effect is that the overlong data entry may have overwritten other parts of the screen display. These sections of the display may remain in their 'corrupted' form until that particular option of SETUP is exited.

5.1.2. Input checking

In general SETUP checks data on entry, and will provide error messages and/or warning messages if incorrect or inconsistent values are entered. The checking is comprehensive but not exhaustive, and is intended to catch the small mistake rather than the gross error. For example SETUP will not let you define a sector translation table with the wrong number of sectors, or where a sector number is entered more than once. But it will let you define a format with 50 1024-byte sectors to a track on a single density 5" disk.

5.1.3. Error messages

In the event of an error occuring, either in the response you have entered (e.g. illegal character), or as a result of that entry or an earlier entry, an error message will flash briefly at the bottom of the screen before the program continues. It may request you to re-enter the erroneous value(s), or may return to the main menu depending on the circumstances surrounding the error.

5.1.4. Aborting a Command

You may abort the current operation at any time by typing ESCape (the ESCAPE key) while replying to the prompt for input by SETUP. e.g. If you have selected option 2, (List current Formats), you may terminate the listing at any time by pressing ESCape in response to the prompt "Type <RETURN> to continue".

5.1.5. Print-out of display

You may print out the contents of the current screen at any time by typing ^P (pressing the 'P' key whilst holding down the CONTROL key) while replying to the prompt for input by SETUP. e.g. If you have selected option 2, (List current Formats), you may obtain a printer copy of the display by pressing ^P in response to the prompt "Type <RETURN> to continue".

5.2. Main Menu

On running SETUP it will display the initial menu shown in Fig 1 (overleaf).

The top three lines of the display will remain in place throughout the execution of the various options within SETUP, and show the current configuration of the MFB.

The top line of the display gives the version number of SETUP. The second line shows the formats supported by each drive, and the third line shows the drive physical address (in brackets) followed by the name of the supported drive. If a drive is undefined its entry remains blank. If the format is not a normal CP/M format, then the format name is highlighted.

The drive/format display starts with the first floppy format. All logical drives that are part of the Winchester drive are omitted from the display. (eg If your system is based on a 10Mbyte Winchester that you have partitioned into three logical drives - A,B, & C - then the first drive appearing in the heading will be

Drive D.

```
================================================================================
        Gemini Multi-Format-BIOS Setup program      Version 3.3
 D = GEMQDDS    E = GEMDDDS    F = IBMPC8     G = IBM3740    H =
(0)  MID5"     (1)  RIGHT5"   (1)  RIGHT5"   (7)  EIGHT"


                    Return to CP/M.............0
                    Enter/Edit a format.........1
                    List current Formats........2
                    Assign formats to drives.....3

                    Enter required option :
```

```
================================================================================
```

Enter a number between 0 and 3 followed by RETURN to perform the operation you require.

5.3. Enter a new Format

Typing "1 <RETURN>" in the main menu will select this option and produce the screen display:

```
================================================================================
        Gemini Multi-Format-BIOS Setup program      Version 3.3
 D = GEMQDDS    E = GEMDDDS    F = IBMPC8     G = IBM3740    H =
(0)  MID5"     (1)  RIGHT5"   (1)  RIGHT5"   (7)  EIGHT"
             Format Definition

       Format name.........: --------          OS type (CP/M=0)..: CP/M
       Format description...: ------------------------------

Drive type (8/5/W): 5    Side mode(C/T/S/U): S    Index mark (Y/N)...: N
Tracks-per-inch...: 96                            Format byte........: E5
Tracks-per-side...: 80   Logical secs/trk..: *    First sector no....: 0
Sides-per-disk....: 2    Reserved tracks...: 2    Phys. sector skew..: 0
Sectors-per-track.: 10   Block size (k)....: 4    Gap 1 length.......: 34
Bytes-per-sector..: 512  Extent mask.......: *    Gap 2 length(34/43): 35
Density (S/D).....: D    Directory entries.: 128  Gap 3 length(>34)..: 37
Data rate (H/L)...: L                             Side fields........: *
Invted data (Y/N).: N                             Sector size field..: *
Reverse sides(Y/N): N                             Special init.......: N

Sector translate table :
```

```
================================================================================
```

The cursor will be positioned ready to accept a Format Id (see below). If you have just selected option "1", then the string typed in here will be compared against those already stored in the data file. If a match is found then that entry will be copied to the screen, and you will pass immediately to the "Save, Delete, Edit, Ignore? " prompt (see below). If the string did not match an existing entry in the data file, then the remaining fields have to be filled in.

The answers required for each entry are shown below:-

Format name: Enter an eight-character string. This is the string that will be used to identify this particular format entry. The input string is forced to upper case irrespective of whether upper or lower case is used in typing it in. Blanks may be used within the name if desired. (eg GEM QDDS).

OS type (CP/M=0): This field shows the operating system that goes with the format. A single digit is used to identify the type of operating system, with 0 representing CP/M. (SETUP displays the operating system name for clarity, while for entry you should use a single digit). Currently one other format is recognised, and that is '1' for MSDOS.
NB. MFB II runs the CP/M operating system, and can only directly handle disks with an identical directory structure. (eg MP/M, OS/M, CDOS). Disks in a foreign format (such as MSDOS) require a special utility to read and write files on them. These utilities make use of the FORMATS.DAT file to determine the organisation of the files on the foreign format disk.

Format description: You can enter a string of up to thirty-one characters to expand on the "name" field. Initially the display shows a series of dashes to show the length of the field.

Drive type (8/5/3): This is a single digit that identifies the type of drive required for this particular format. SETUP cross checks this entry against that of the specified drive entry when constructing a system (see below).
If the format is to use the high density feature of the special 5.25" drive, (achieved by using a higher rotational speed than normal), then an 'H' should be appended to the 5. (ie Enter '5H' in this case).

Tracks-per-inch: Enter the appropriate figure for the drive (e.g. 48 or 96). SETUP cross-checks this field with a Drive description when setting up a BIOS. (It can also assign a 48tpi format to a 96tpi drive, in which case it arranges for the 96tpi drive to be

double-stepped so that it appears to the system as a 48tpi drive. To double-step a 135tpi drive, enter a figure of 67 rather than 67.5).

Tracks-per-side: Enter the number of tracks per side expected by the Format. SETUP cross-checks this field with that in a Drive description when setting up a BIOS. It also uses this field in calculating the capacity of a disk.

Sides-per-disk: Enter the number of sides to the disk required by this Format. (This must be 1 or 2). SETUP cross-checks this field with that in a Drive description when setting up a BIOS. It also uses this field in calculating the capacity of a disk.

Sectors-per-track: Enter the number of physical sectors per track per side of the disk. SETUP uses this figure to determine the physical sector numbers (shown below) and also in accessing the disk. The number must lie between 1 and 52.

Bytes-per-sector: Enter the number of bytes per sector. This must be 128, 256, 512 or 1024. SETUP uses this figure in determining whether Blocking/Deblocking is to be performed, and also in calculating the capacity of a disk. The Format command also uses this field.

Density (S/D): Enter S for single density, or D for double density. This field is used by the system for configuring the disk controller when accessing a disk of this particular format. It is also used by the format program when formatting a disk.

Data rate (H/L): The reply to this question is normally 'H' for 8" formats and 'L' for 5" formats. This sets the data rate between the controller and the disk drive. However for formats such as that of the IBMPC/AT, that utilise the newer high capacity 5.25" drives, this field should be set to 'H'. In this case the 'Drive type' (see above) should be set to **5H** to request a drive with a high speed option. (This is the MID5" drive on standard MFB systems).

Inverted data (Y/N): The reply to this question is normally N for No. However some formats (e.g. Superbrain) write the complement of the data onto the disk. If this field is set to "Y" the BIOS will complement the data during all disk I/O.

Reverse side (Y/N): The reply to this question is normally N for No. However some formats (eg Sharp MZ80B) format side 0 of a disk as side 1, and vice versa. In this case enter 'Y', and the MFB system will treat the

disk in a similar manner.

Side change (C/T/S/U): For formats that use both sides of a double-sided disk this field indicates how the system should access both sides of the disk.
S indicates that it should use one side completely before using the other side. (i.e. The tracks are ordered side 0 tracks 0->N, side 1 tracks 0->N).
U is a variation on S in that the logical track ordering on the second side is in the reverse direction. (i.e. The tracks are accessed in the order side 0 0->N, side 1 N->0).
C indicates that each logical track is a Cylinder. (i.e The tracks are accessed in the order track 0/side0, track 0/side 1, track 1/side 0, track1/side 1,..etc).
T results in the disk being accessed in an identical manner to **C**, but each logical track matches a physical track.
The C & T formats are distinguished by the number of logical sectors/track shown in the 'stat dsk:' display. Taking the example of an 80 track 5" disk that is formatted with 10 512 byte sectors per side: This gives (10 * 512 / 128 = 40) logical sectors per track per side. A **C** format disk would appear to have 80 tracks of 80 sectors/track, while a **T** version with the same physical layout would have 160 tracks of 40 sectors/track.

There is also another very unusual variant that can be specified. This variation is unique in that it departs radically from convention in how a disk is *physically* formatted. (As it is a very unusual format, no attempt has been made to include a reminder for it as part of the prompt for the 'Side change' field).

This format is designated BUTLER, (after the system that uses it!), and is selected by entering a **B** in the 'Side change' field. With this format the disk is accessed in **T** mode described above, but the tracks on the disk surface are *physically* numbered in the order they are accessed. ie track 0/side 1 is formatted with a '1' in the 'track' field of the sector headers, track 1/side 0 with a '2', track 1/side 1 with a '3', and so on.

The prompts in the remaining section of the centre column of the display will depend upon the OS type indicated earlier.

CP/M prompts

Logical secs/track: Normally a logical track is equivalent to a physical track and SETUP will use a figure calculated automatically based on the Bytes-per-sector/Sectors-per-track/Side change fields. As with the 'Extent Mask' field this automatic option is selected by entering an '*' (or 0). However in the case of the supplied RAIRQD and M2215C formats you fill find that the field contains a different value to that which you would expect. This is because, taking the M2215C format as an example, you will find the system is configured so that a physical sector is a logical track. (Hence the logical secs/track is 4). For formats like this the BIOS has to do additional work to translate a logical track and sector combination to the correct physical track and sector numbers on every disk Read/Write. In the unlikely event of your encountering a similar format, the appropriate logical track size should be entered here.

Reserved tracks: Enter the number of logical tracks on the disk reserved for the system tracks.

Block size (kbytes): This is the block allocation size used by CP/M for this particular format. SETUP uses this field when it constructs the disk parameter block for a particular drive while setting up a BIOS.

Extent mask: Normally you should enter an '*' for this field. SETUP takes this as an instruction to use the correct value for this parameter when constructing the disk parameter block. However some formats have been encountered, (e.g. Televideo), where this value has been set incorrectly. In this case the value that SETUP is to use should be entered here. (See appendix B).

Directory entries: This is the number of directory entries supported by this particular format.

NOTE: Most of the CP/M specific items above can easily be determined by doing a "STAT DSK:" command (or equivalent) on a system running the desired format.

MSDOS parameters. Further details on these can be found in the MSDOS section of the manual. These fields are used by the various MSxxxx utility programs.

Boot size(sectors): Enter the Boot size in sectors.

Sectors/Cluster: Enter the number of sectors per cluster.

Sectors/FAT: Enter the number of sectors per FAT. (File allocation Table).

Media byte: Enter the value of media byte for this format. (A two-digit hexadecimal value).

Directory entries: Enter the number of directory entries.

Number of FATS: Enter the number of FATS (File Allocation Tables) on the disk.

FAT ID byte: Enter the FAT ID byte.

The final column of entries relate to the physical characteristics of the disk and are used in the main by the format program. Therfore, if you only wish to read/write disks and will not be formatting them, arbitary values can be entered in the various fields with the exception of 'First sector no', and 'Side fields'.

Index mark (Y/N): This field is used by the Format program. If a 'Y' is entered the format program will write an Index mark at the start of every track on the disk before writing the data sectors. With the Western Digital controller used on the Gemini disk interface cards the Index mark is superfluous and is never used. It must be left off for the Gemini formats (to ensure that there is sufficient space for the data blocks), but should be included for those formats that require it.

Format byte: This is the byte that is used to fill all the sectors of a disk when it is formatted. CP/M expects the hexadecimal value of E5. Press <RETURN> to take the default of E5, or enter a two-digit hexadecimal number.
NB The format program checks the 'Invted Data' field, and will automatically invert the format byte for those formats that require it. (eg SUPERBRAIN).

First sector no: Enter the hexadecimal number of the first sector of a track. This is usually either 0 or 1.
NOTE At this stage the display on the screen will change slightly. The 'sector translate table' field at the bottom of the screen will be replaced by a display of the 'physical sector numbers'. This is done so that the exact effect of the 'sector skew' setting (see below) can be seen. The 'sector translate table' is redisplayed once the 'gap3' entry is passed.

Physical sector skew: This field is used for those formats, such as the Gemini formats, that achieve sector skewing by physically skewing the sectors on a disk during

the formatting operation. If no skew is to be used then a 0 should be entered. This field is used only by the Format program.

Gap 1: This field is used by the Format program, and is the number of bytes that are inserted following the Index mark, (if present), and before the first sector header.

Gap 2: This field is used by the Format program, and is the number of bytes that are inserted between a sector header and its associated data block.

Gap 3: This field is used by the Format program, and is the number of bytes that are inserted between the end of a sector and the start of the next sector header.

Side fields: This field is used by the format program. The sector header of a soft-sectored disk includes a byte holding the side number (0 or 1). Some formats use non-standard values in these fields, in which case they should be entered here.
An asterix ('*') indicates that the correct values of 0 and 1 are to be used. To subsitute alternative values, a single 4-digit hexadecimal value should be entered, where the first two digits represent the setting for side 0, and the next two digits the setting for side 1. This will be redisplayed as two 2-digit hex numbers with an intervening space for clarity.
NOTE: The disk controller on the MFB's disk interface board (a Western Digital 2793) can be set to ignore the side field in the header record, and so can successfully read disks irrespective of the setting of this field. Not all disk controllers have this feature.

Sector size field: This field is used by the format program. The sector header of a soft-sectored disk includes a byte which indicates the sector size. An asterix ('*') indicates that the correct value is to used, and should be used in 99.99% of all formats. As above - a two digit hexadecimal number may be entered to overwrite this field with another value for certain special cases.
CAUTION The effect of non-standard values in this field on the controller is uncertain, and should be carefully verified before being used.

Special init: This field requires a 'Y'es or 'N'o entry, Yes indicating that the format program has to take further steps once the disk is formatted. Some disk formats require special initial information to be written to certain sectors on the disk before they can be recognised by the host machine. (eg

partitioning information or a format code). If a 'Y' is entered here, the Format program will look for a file with a name <FORMAT$NAME.INI> (eg GEMQDDS.INI) which holds the additional information. (See section on FORMAT).

Physical sector numbers: (See note above under 'First sector no.'). This field shows the order in which sectors will be written onto the disk by the Format program. It is automatically filled in by SETUP using the values from the First sector no/Physical sector skew/Sectors-per-track fields.

Sector translate table: If sector skewing is performed by a table, (such as in the standard 8" single density format), then the table must be entered here. There must be as many entries as there are physical sectors to a track. (Note: If the side change mode is C, then the translate table is assumed to span both sides of the disk, and twice as many entries are required compared to the S, T and U options). If the physical sector size is 128 then the translation is done via SECTRAN in the BIOS in the usual way, but if the sector size is > 128 then the translation is performed within the disk read/write routines of the BIOS. SETUP will not let you duplicate sector numbers in the table, and can help you with the entry if you require.

You may either enter all the sector numbers in one go on a single line, or press <RETURN> after entering each number. In the latter case SETUP will assist you in constructing the table. It will determine the sector skew you are using by examining the first two sector numbers entered. Thereafter every time <RETURN> is pressed it will redisplay the values entered so far, and will add onto the end what it thinks should be the next sector number. If <RETURN> on its own is pressed this will be taken as the default value for the next sector, and it will update the display.

NOTE SETUP does not automatically "slip" the next sector number if the skew does not exactly fit in with the sectors per track. However if you fail to notice that a sector number is about to be repeated SETUP will provide an error message when <RETURN> is pressed, and will not accept the entry. You can then manually alter the number and continue.

If you have previously entered a sector translation table and wish to delete it, this can be done by entering a single '-' at the start of the table.

Once all the fields have been completed the prompt:

Save, Delete, Edit, Ignore? (S/D/E/I):

will appear at the bottom of the screen. As indicated four options are open:

S Will save the displayed entry to the data file, overwriting any existing entry with the same "Format name"

D Will delete any entry with a matching "Format name" from the data file.

E Will move the cursor back to the start of the entry, allowing any erroneous values to be changed. Note that this time altering the "Format name" field to match an existing entry will not result in that entry overwriting the currently displayed values.

It is also possible to quickly enter a new format that is very similar to one that already exists. By typing the ID of the existing format in the first instance, the appropriate values are copied into the display. Then the "E" option can be used to alter the ID field, and any of the other values that need to be changed.

I Will return you to the main menu without altering the data file.

5.4. List Current Formats

Typing "2 <RETURN>" in the main menu will result in the data on all the current Formats being displayed on the screen. In the "full" display mode the entries are shown one at a time in the form illustrated below.

```
================================================================================
        Gemini Multi-Format-BIOS Setup program     Version 3.3
 D = GEMQDDS    E = GEMDDDS    F = IBMPC8     G = IBM3740    H =
(0)  MID5"     (1)  RIGHT5"   (1)  RIGHT5"   (7)  EIGHT"
             Format Definition

       Format name.....: GEMQDDS           OS type (CP/M=0)..: CP/M
       Format description...: Gemini QD DS

Drive type (8/5/W): 5    Side mode(C/T/S/U): S    Index mark (Y/N)...: N
Tracks-per-inch...: 96                            Format byte.......: E5
Tracks-per-side...: 80   Logical secs/trk..: *    First sector no....: 0
Sides-per-disk....: 2    Reserved tracks...: 2    Phys. sector skew..: 2
Sectors-per-track.: 10   Block size (k)....: 4    Gap 1 length.......: 44
Bytes-per-sector..: 512  Extent mask.......: *    Gap 2 length(34/43): 34
Density (S/D).....: D    Directory entries.: 128  Gap 3 length(>34)..: 42
Data rate (H/L)...: L                             Side fields........: *
Invted data (Y/N).: N                             Sector size field..: *
Reverse sides(Y/N): N                             Special init.......: N

Sector translate table : None


                    Use Cursor Keys to Select next entry
                  or ESC to Exit, E to Edit, D to Delete
================================================================================
```

The RETURN key or CURSOR keys can be used to step through from one entry to the next. (The -> key will step forwards through the entries, the <- key will step backwards). Entries may also be Edited or Deleted by pressing the appropriate key. The ESCape key can be used to return to the main menu.

Pressing 'E' enables the entry to be edited in an identical fashion to option 1 of the Menu.

Pressing 'D' for delete results in a prompt for a confirming 'Y' before the action is carried out.

As mentioned in section 5, typing Control/P will result in the screen display being copied to the printer.

If only a brief summary is required a 'Q' (for Quick) should be appended to the "2". (i.e. type "2q"). In this case only a sorted list of the format names is printed, with eight entries per line, until the list is complete or the screen is full. If there is insufficient space available in the display area to show all the formats, then the cursor-up and cursor-down keys may be used to scroll the list of formats up and down through the display area. If the format library is particularly large, Shift/Cursor up and Shift/Cursor down will scroll the display in larger increments

```
================================================================================
       Gemini Multi-Format-BIOS Setup program     Version 3.3
 D = GEMQDDS    E = GEMDDDS    F = IBMPC8     G = IBM3740    H =
(0)  MID5"    (1)  RIGHT5"   (1)  RIGHT5"   (7)  EIGHT"

                        Formats

AARDVARK  ALM5000   ALMDSDD   ALPHAP2   ALTOS     ALTOS3R   ALTOS8    ALTOS8NS
ALTOSM55  APRICOTC  APRICOTM  ARACOM    AT        AT1       B25       BBC
BBC/96    BBC96     BBCDD     BBCDD96   CROMDSDD  CROMSYS2  DEC100?   DEC100DS
DEC100SS  DECVT180  DIGICO    DTIDELTA  EPSONQX   EQUIDSDD  EQUINSX   EQUISSDD
FTSDS     FTSSS     FUJI16S   FUJI16SM  FUJITSU7  FX20INT   GEMDDDS   GEMQDDS
GEMQDSS   GEMSDDS   GENIII    HEADST40  HHTIGER   HP150M    IBM3740   IBM3740D
IBMPC8DM  IBMPC8SM  IBMPC8SS  IBMS34    IBMS34DS  IBMXT9DM  IBMXT9SM  IBMXT9SS
ICL8801C  ICLDRS20  ICLPC1    ICLPC1SD  ICLPC2    IDS2009   INTELPDS  ITT3030
KAYPRO    LOG2200C  LOGICVTS  LSI4DSD8  LSIM3     LSIM4     LSIOCT40  LSIOCT80
M2200 QD  M4000     MCCOMBO   MCS       MEMORY4   MIMI803D  MINSTRL2  NASLDSDD
NASLSSDD  NCR       NCRDM5    NEC58800  NEC8001   NEC8800   NECAPC    NEWBRAIN
NOHALT    OLBOSSB   OPEOPLE   OPEOPLEM  OSBEXEC   OSBSSSD   P2000CQD  P2000CSS
P3500     PERCOMDS  PERCOMS8  PICCOLO   PIPER     PORTICO   PRODIGY   RAIRSDD
RAIRQD    RAIRSSDD  RAIRSSSD  RMLDDSS5  RMLDDSS8  RMLDS     RMLSDSS5  SAMDSDD8
SAN1000   SAN1110   SAN2000   SAN3000   SBRAIN40  SBRAINDD  SBRAINQD  SDSYSSD
SHAR3201  SHELIII   SLICER    SONY35WP  TORCH     TOSHT200  TRANSDS   TRANSTSS
TVID802   TVID96    TYCOM     TYCOM/M   WANGPC    WILCOX    XER820DS  XER820SD
              Use Cursor keys to Scroll - RETURN to exit

================================================================================
```

5.5. Assign Formats to drives

Typing "3<RETURN>" in the main menu will select this option.

This option directly patches the running system, thus the current configuration will be disturbed. The patching starts with the first non-winchester logical drive, and proceeds through succeeding logical drives until one of the following occurs:

a) The available table space in the BIOS has been exhausted.
b) Five floppy logical drives have been defined.
c) The ESCape key is pressed.

On completion you are given the option of copying the changed BIOS tables to the system track of drive A in order to record the changes permanently. NOTE This is NOT equivalent to a "SYSGEN". Only the changed parts of the BIOS are written to the disk, not the entire CP/M system.

The first prompt for a Format Code is shown below. Note that a series of ^^^^^^^^ on the fourth row of the display indicate which logical drive is being defined.

```
================================================================================
       Gemini Multi-Format-BIOS Setup program     Version 3.3
 D = GEMQDDS    E =           F =           G =           H =
(0)  MID5"
configuring->   ^^^^^^^^^^

                                  Formats

AARDVARK  ALM5000   ALMDSDD   ALPHAP2   ALTOS     ALTOS3R   ALTOS8    ALTOS8NS
ALTOSM55  APRICOTC  APRICOTM  ARACOM    AT        AT1       B25       BBC
BBC/96    BBC96     BBCDD     BBCDD96   CROMDSDD  CROMSYS2  DEC100?   DEC100DS
DEC100SS  DECVT180  DIGICO    DTIDELTA  EPSONQX   EQUIDSDD  EQUINSX   EQUISSDD
FTSDS     FTSSS     FUJI16S   FUJI16SM  FUJITSU7  FX20INT   GEMDDDS   GEMQDDS
GEMQDSS   GEMSDDS   GENIII    HEADST40  HHTIGER   HP150M    IBM3740   IBM3740D
IBMPC8DM  IBMPC8SM  IBMPC8SS  IBMS34    IBMS34DS  IBMXT9DM  IBMXT9SM  IBMXT9SS
ICL8801C  ICLDRS20  ICLPC1    ICLPC1SD  ICLPC2    IDS2009   INTELPDS  ITT3030
KAYPRO    LOG2200C  LOGICVTS  LSI4DSD8  LSIM3     LSIM4     LSIOCT40  LSIOCT80
M2200 QD  M4000     MCCOMBO   MCS       MEMORY4   MIMI803D  MINSTRL2  NASLDSDD
NASLSSDD  NCR       NCRDM5    NEC58800  NEC8001   NEC8800   NECAPC    NEWBRAIN
NOHALT    OLBOSSB   OPEOPLE   OPEOPLEM  OSBEXEC   OSBSSSD   P2000CQD  P2000CSS
P3500     PERCOMDS  PERCOMS8  PICCOLO   PIPER     PORTICO   PRODIGY   RAIRSDD
RAIRQD    RAIRSSDD  RAIRSSSD  RMLDDSS5  RMLDDSS8  RMLDS     RMLSDSS5  SAMDSDD8
SAN1000   SAN1110   SAN2000   SAN3000   SBRAIN40  SBRAINDD  SBRAINQD  SDSYSSD
SHAR3201  SHELIII   SLICER    SONY35WP  TORCH     TOSHT200  TRANSDS   TRANSTSS
TVID802   TVID96    TYCOM     TYCOM/M   WANGPC    WILCOX    XER820DS  XER820SD
                  Enter Format code: GEMDDDS

================================================================================
```

A default Format code will be shown if the logical drive has been defined in the previous configurations. In response to the prompt a format code should be selected from those displayed on the screen and entered. (Or RETURN pressed to take the default). If the number of available formats is too large to be displayed in the space available, then cursor-up and cursor-down keys can be used to scroll the display up and down within the available display window. (The cursor keys will only be recognised while the cursor is positioned at the start of the data field). One other option is open, and that is to press the ESCape key instead. Doing this will terminate the "drive definition" phase.

Once RETURN is pressed SETUP checks that the selected format exists. If it does not the error message ("Format Code is undefined") will flash briefly at the bottom of the screen, and you will be prompted for a new format.

Once the format has been located in the data file, SETUP checks the characteristics of all the drives in your system in order to locate the drive that can support it. (Right type of drive, correct tpi, sufficient number of tracks/sides, etc). If no suitable drive can be found, the error message "No drive supports the format" will be displayed and you will be requested to re-enter the format.

NOTE 1: It is permissable to enter several formats that will use the same physical drive. For example on a system with a Teac FD55B type drive:

Logical drive D could be GEMDDDS format on the FD55B
Logical drive E could be GEMSDDS format on the FD55B,
Logical drive F could be SBRAIN format on the FD55B.

This would enable transfers to be carried out between the various formats with only one 48tpi drive in the system, but obviously all transfers would have to be via the Winchester drive or drive **M** (if fitted). To attempt to "PIP" files directly between D,E and F will only result in disaster! If the transfer of the files is to be done via the Winchester drive it will probably be more convenient to do them via an unused CP/M user area. This would allow the shorthand file specification of '*.*' to be used for the transfer onto and off the Winchester, and also for the subsequent removal of the temporary copies of the files from the Winchester.

NOTE 2: If required, SETUP can select a 96tpi drive to support a format that normally uses a 48tpi drive. In this case SETUP arranges for the BIOS to double-step the drive so that it appears to the system as a 48tpi drive. It is also possible to influence SETUPs choice of drive when requesting a format. This is done by following the format ID by a physical drive address in square brackets []. e.g. If the drive with a physical address of 0 is a 96tpi drive, then entering GEMDDDS[0] will result in it being selected to support the format (and double-stepped as a result), in place of the 48tpi drive which would have been the natural choice.

Once an acceptable format code has been entered, SETUP modifies the BIOS so that it is supported, and moves on to configure the next drive. Once all drives have been defined, (or the ESCape key is pressed), SETUP prompts with:

Modify the system track to match?

Responding with a 'Y' results in the system track being updated to match the current BIOS. 'N' leaves it alone, in which

case the current configuration will only remain valid until the system is 'cold' started. (eg Pressing RESET or switching the power off and then on again).

During the construction of a new BIOS the message:

Workspace Overflow - requires a smaller system

or Sector translation table overflow

may occur. The former occurs because as new drives are added to the system, SETUP allocates workspace at the end of the BIOS for the appropriate space allocation bit-map and directory check area. The larger the capacity of the drive added, the more space required. Each time SETUP allocates workspace it checks its pointer to the top of the BIOS to ensure that it has not wrapped round to 0. In the event of it doing so the above message appears and the BIOS configuration ceases. All drives defined upto (but not including) the current drive remain defined.

The latter message occurs when the area of memory reserved in the BIOS for the sector translation table overflows. Currently SETUP creates an individual table for each drive (when required), and so, for example, a system with two 8" drives would use up 52 bytes of the table area. A total of 168 bytes is reserved for the table which should be more than adequate.

In the event of the message appearing, a different (or reduced) mix of formats will have to be selected. (Remember SET provides a way of switching rapidly between various predefined format environments, and can be used to minimise the effect of not having every popular format online simultaneously).

6. SET

SET is a stripped-down version of SETUP that only performs option 3 of SETUP's menu (Assign formats to Drives), and takes its input from a file rather than the keyboard. It provides a shorthand way of switching between various standard environments.

It is invoked by typing:

```
SET filename
```

where <filename> is the name of an ASCII file on drive A: containing the configuration information. The file name must have the extension .CFG. (e.g. NORMAL.CFG).

6.1. Configuration File

The .CFG file holds the desired formats as a list, one per line, in exactly the same form as they would have been typed in response the prompt from SETUP. Additionally a semicolon may be used to add a comment to the file.
(NB A blank line in the file will result in SET taking the default of current setting for that drive).

e.g. The file NORMAL.CFG could contain the entries:

```
gemqdds
gemddds[0]
ibm3740
apricot
```

and the system configuration could be reset at any time by typing:

```
SET NORMAL
```

NOTE: SET only changes the configuration of the system in memory. It does not offer the option of saving the new settings to the system track. SET provides a shorthand mechanism for making often used temporary changes to the format/drive combinations. Any permanent change to the system configuration has to be done via option 3 of SETUP.

7. FORMAT

FORMAT is a separate program that runs in an MFB system and formats or verifies disks using the details stored in the FORMATS.DAT file. FORMAT is invoked by typing FORMAT<RETURN> in response to the CP/M 'A>' prompt.

Once FORMAT has loaded it puts up the usual MFB heading and then prompts for a drive letter:

Which Drive? ([D-L] or ESC)

You may respond with either a Logical drive number (A-L excluding any Winchester drives), or press the ESCape key. If a valid logical drive letter is entered then FORMAT will use the drive and format pair (currently displayed at the top of the screen) which are associated with that logical drive. If ESCape is pressed FORMAT will display the fuller prompt of:

```
================================================================================
        Gemini Multi-Format-BIOS Format program     Version 3.3
 D = GEMQDDS    E = GEMDDDS    F = IBMPC8     G = IBM3740    H =
(0)  MID5"     (1)  RIGHT5"   (1)  RIGHT5"   (7)  EIGHT"

                          Formats
AARDVARK  ALM5000   ALMDSDD   ALPHAP2   ALTOS     ALTOS3R   ALTOS8    ALTOS8NS
ALTOSM55  APRICOTC  APRICOTM  ARACOM    AT        AT1       B25       BBC
BBC/96    BBC96     BBCDD     BBCDD96   CROMDSDD  CROMSYS2  DEC100?   DEC100DS
DEC100SS  DECVT180  DIGICO    DTIDELTA  EPSONQX   EQUIDSDD  EQUINSX   EQUISSDD
FTSDS     FTSSS     FUJI16S   FUJI16SM  FUJITSU7  FX20INT   GEMDDDS   GEMQDDS
GEMQDSS   GEMSDDS   GENIII    HEADST40  HHTIGER   HP150M    IBM3740   IBM3740D
IBMPC8DM  IBMPC8SM  IBMPC8SS  IBMS34    IBMS34DS  IBMXT9DM  IBMXT9SM  IBMXT9SS
ICL8801C  ICLDRS20  ICLPC1    ICLPC1SD  ICLPC2    IDS2009   INTELPDS  ITT3030
KAYPRO    LOG2200C  LOGICVTS  LSI4DSD8  LSIM3     LSIM4     LSIOCT40  LSIOCT80
M2200 QD  M4000     MCCOMBO   MCS       MEMORY4   MIMI803D  MINSTRL2  NASLDSDD
NASLSSDD  NCR       NCRDM5    NEC58800  NEC8001   NEC8800   NECAPC    NEWBRAIN
NOHALT    OLBOSSB   OPEOPLE   OPEOPLEM  OSBEXEC   OSBSSSD   P2000CQD  P2000CSS
P3500     PERCOMDS  PERCOMS8  PICCOLO   PIPER     PORTICO   PRODIGY   RAIRSDD
RAIRQD    RAIRSSDD  RAIRSSSD  RMLDDSS5  RMLDDSS8  RMLDS     RMLSDSS5  SAMDSDD8
SAN1000   SAN1110   SAN2000   SAN3000   SBRAIN40  SBRAINDD  SBRAINQD  SDSYSSD
SHAR3201  SHELIII   SLICER    SONY35WP  TORCH     TOSHT200  TRANSDS   TRANSTSS
TVID802   TVID96    TYCOM     TYCOM/M   WANGPC    WILCOX    XER820DS  XER820SD
                    Enter Format code:

================================================================================
```

In reply to this enter the Format that you require in a similar way to that of the 'Assign formats to drives' option of SETUP. FORMAT will locate and use the drive appropriate to that format. If the number of available formats is too large to be displayed in the space available, then cursor-up and cursor-down keys can be used to scroll the display up and down within the available display window.

NOTE: The reply is forced to upper case.

Once a valid format has been selected the display will change to:

```
================================================================================
       Gemini Multi-Format-BIOS Format program     Version 3.3
 D = GEMQDDS    E = GEMDDDS    F = IBMPC8     G = IBM3740    H =
(0) MID5"     (1) RIGHT5"   (1) RIGHT5"   (7) EIGHT"

                        Drive is MID5"
                        Format is GEMQDDS




              'F' to Format, ESC to Exit, any other key to Verify

================================================================================
```

There will be a short delay before the messages appear while FORMAT constructs the appropriate track image in memory that matches the selected format.

In the event of there being no disk in the selected drive, the message 'Insert disk in drive' will flash on and off in the centre of the screen until one is inserted in the drive. At this point the 'F' to format message will appear.

Upon pressing 'F' FORMAT will format the disk, displaying the number of the track currently being formatted as it proceeds. The formatting process may be interrupted at any time by pressing the ESCape key. Once the format operation is complete FORMAT proceeds to verify the disk. It restores the head to track 0 and then reads the entire disk to verify that all the sectors can be read. Once the verification phase is complete FORMAT will return to the 'F' to format... prompt unless one or more disk errors occured. In that case it pauses with the verfication error messages on the screen until the RETURN key is pressed.

Pressing a key other than 'F' or ESCape will result in FORMAT going straight to the verification phase for a disk. This provides an easy way of checking suspect disks for readability. (In normal use the BIOS only reports a read or write error after many retries, and so a poor quality disk formatted on another system may not be noticed until it is too late).

During the verification phase it will report all retries that occur, along with the reason why the read failed. It is only after eight retries that a sector is abandoned, the message

<<< Bad <<<

being appended to the display to indicate this.

Any disk that produces more than one or two retries should be regarded with suspicion, and any one that results in the <<< Bad <<< messages should be discarded if they re-occur if the disk is re-formatted.

The more usual error messages during the verification read are:

RNF - Record not found. The controller was unable to locate either the sector header, or the following data block.

CRC - CRC error. A CRC error was detected, either in the header field or the following data block.

Other possibilities are:-

DNR - Drive Not Ready.

WP - Disk Write protected. (Should not appear).

LD - Lost Data. (Should not appear).

??? - Undefined error status. (Should not appear).

The current version of FORMAT does not support the more unusual formats, although it does cater for minor variations. (See the descriptions of the various data fields in the section on SETUP).

Specifically:

FORMAT will not format disks in the BUTLER format. (This uses unconventional physical track numbering).

FORMAT will not format disks to the Sharp MZ80B format. (This uses unconventional physical sector numbering on the second side. The MFB system copes with this when reading and writing by using a slightly unusual sector translation table.)

8. WHIG

WHIG, (What Have I Got?), is a simple program that just displays the current system configuration at the top of the screen in a format similar to that of SETUP. If you are working with a variety of different formats you may forget exactly what configuration you are currently running. Typing "WHIG" answers this question with a display

```
                    **** Current BIOS configuration ****
 D = GEMQDDS    E = GEMDDDS    F = IBMPC8     G = IBM3740    H =
(0) MID5"     (1) RIGHT5"   (1) RIGHT5"   (7) EIGHT"
```

at the top of the screen.

As with the display within SETUP, any non CP/M format will be highlighted.

Unlike the other MFB utilities, WHIG will also run under a normal Gemini CP/M system. In this instance it will also produce a listing of the current system configuration, but the display format is slightly different, and is illustrated below:

```
**** BIOS Configuration ****

BIOS version = 3.4
Space for 8 logical drives with 7 defined
Logical : Physical : Format
   A          W       Winchester
   B          W       Winchester
   C          W       Winchester
   D          0       QDDS
   E          1       QDDS
   F          1       IBMPCDS
   G          7       8SD
```

9. ANALYSE

ANALYSE is a program that has been written to assist in determining the values required by SETUP for the physical characteristics of a disk format. ANALYSE will attempt to determine:

* Rotational speed (if dual speed drive)
* Density
* No. of sides in use
* Disk tracks-per-inch (tpi)
* Number of tracks
* Sector size
* First sector number (0 or 1)
* Sectors/track
* Index mark presence
* Gap1/Gap2/Gap3 sizes
* The order of the physical sector nos.

ANALYSE will do its best to determine the above items, but by the very nature of what it is attempting to do it may make some mistakes in interpreting the data. It is advisable to run the program several times to check that the results it gives are consistent. With 5.25" disks, it has sometimes been found helpful to try the disk in both the 96tpi and the 48tpi drives.

To get the best out of ANALYSE the disk under analysis should be a freshly formatted one, or one with only a limited amount of data on it that has not been heavily used. With a disk on which the sectors of a track have been re-written several times, the Gap1/Gap2/Gap3 figures may vary between runs of the program.

In the following sections the operation of ANALYSE will be described, with the possible pitfalls at each stage being highlighted.

9.1. Running ANALYSE

When ANALYSE is run it first prompts for a logical drive address in a similar way to FORMAT. If the drive you wish to use is not currently defined as a logical drive on the system, then pressing ESCape will result in the alternative prompt for a Drive name. Once the drive to be used has been determined, ANALYSE asks for the disk to be analysed to be placed in the selected drive, and the drive door to be closed. Pressing RETURN will then start the analysis procedure.

ANALYSE may be used in an alternative 'Manual' mode by pressing 'M' instead of RETURN. This mode is still under development, but is described in outline at the end of this section.

9.2. Rotational Speed and Density

ANALYSE starts by homing the head on the drive, and then steps the head in to track 4. It sets the density to single and attempts to perform a 'Read Address' command on side 0. In response to this command the controller will read (and pass back to the program) the six bytes of the next sector header that passes under the drive head. If the controller fails to find a header, ANALYSE flips the density to double and tries again. If this read fails ANALYSE flips the density back to single and repeats. After four retries without success it will abandon all further attempts to analyse the disk.

However if the drive is a dual speed drive, (the MID5" drive on the GEMINI MFB system), ANALYSE will also flip the control line that changes the speed of the drive, and will change the data rate to match. Thus it will automatically detect and report on IBMPC/AT and related formats.

If the analysis is carried out in the dual-speed drive, ANALYSE will report the rotational speed used for the successful read.

A successful completion indicates that the correct density (and rotational speed if applicable) has been selected.

9.3. Side

If the drive is double-sided ANALYSE then attempts to do a "Read Address" on side 1 using the previously determined density. If this succeeds then ANALYSE deems the disk to be double sided. If the "side" field of the sector header is not set to "1", (as should be the case), then this fact is reported as it means that the format is non-standard in this respect.

BEWARE: The fact that ANALYSE has found data on the other side of the disk does not necessarily mean that the format is a double-sided one. The disk may be an old disk from another computer that has been reformatted on a new system. e.g. An early Gemini double-sided disk that has been re-used on a Galaxy 2 with single-sided drives.

9.4. Tracks-Per-Inch

Next ANALYSE checks the track number field of the sector header. If the disk tpi matches that of the drive it should find a "4" there. (It started by doing a seek to track 4). If the drive TPI is 96 and the disk was formatted on an 48 tpi drive, then it will find it is positioned over track 2. Similarly a 48 tpi drive and a 96 tpi disk will result in it finding track 8. If the track number does not match any of these combinations, then ANALYSE will print a series of " ????".

BEWARE: There do exist formats that physically number the

tracks in a totally non-standard way. (i.e. A seek to track 10 of the disk finds a track with a totally different number there.)

9.5. Tracks

Following this ANALYSE seeks in to the centre of the disk. It then proceeds to perform a 'Read Address' command. If this fails it steps the head out towards the perimiter and tries again. This is repeated until it succesfully performs 'Read Address' command, whereupon it displays the track number it found in the header record.
(BEWARE a previously used 96tpi disk reformatted on a 35track 48tpi system!).

9.6. First sector number

ANALYSE now attempts to read sector 0 of the track to find out if it exists. If it cannot locate sector 0 after several retries, it then attempts to read sector 1. One of the two reads should succeed, the first successful one being the first sector number of the track. In the unlikely event of ANALYSE failing to find either sector 0 or sector 1 it reports the fact and returns.

9.7. Sector size

When the read routine successfully reads the first sector, it returns a count of the number of bytes it read. This is taken to be the sector size.
CAUTION: The sector size is coded in one byte of the sector header. The controller chip can interpret this byte in one of two ways depending on the polarity of one bit in the Read command issued to it. In the BIOS this is fixed, (set), so the size field is interpreted as either 128/256/512/1024 bytes per sector. The alternative setting gives 256/512/1024/2048 bytes per sector. It is unlikely that any format should adopt this second approach, and there is no provision for its support in the MFB.

9.8. Sectors-per-track

ANALYSE now continues to increment the sector number and re-read the disk until it cannot locate a sector. From this it can determine the figure for sectors-per-track.

9.9. Index mark, Gaps, etc

Now comes the section of the analysis that is most prone to error. ANALYSE steps the head in to an inner track in the hope of finding a virgin track that has not been rewritten since the disk was formatted. It then executes a "Read Track" command.

This command reads in an entire track from the disk to the buffer. It reads in every byte that is actually written on the

track. This includes all the gaps, sector markers, sector headers, CRC check bytes, etc. In an ideal world this command would execute perfectly, but in practice this is not always the case. The data is actually written on the disk as a serial bit stream, and the controller has to synchronise to the byte boundaries within this bit stream* when reading the data back. Occasionally it does lose sync, (especially if a sector has been rewritten which introduces a discontinuity into the bit stream), but will re-synchronise to the next marker it finds.

In the following sections ANALYSE is looking for specific bytes (the markers) in the data stored in the buffer. Based on known facts and information it has already derived it looks in specific areas for them. If it fails to find the byte it is looking for, it will display the part of the buffer it is examining, tell you what it is looking for, and ask you to locate it. To ease the problem of counting the bytes between the start of the display and the wanted marker, ANALYSE numbers the rows and columns of the display in inverse video. An example is illustrated below:

```
Row numbers
|| .. .. .. .. C o l u m n   n u m b e r s .. .. .. .. .. .. ..
================================================================================

00 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
00 E4 E4 E4 E4 E4 E4 E0 00 00 00 00 00 00 00 00 00 00 00 14 A1 A1 BE 19 00
24 00 02 11 8E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 00
48 00 00 00 00 00 00 00 00 00 00 00 14 A1 A1 FB E5 E5 E5 E5 E5 E5 E5 E5 E5
72 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5
96 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5
<0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5
>0 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5

        I can't locate the sector header (FE). Enter byte offset:

================================================================================
```

Here the sector header FE has been misread as BE. You should respond with 22, the number of bytes it is away from the start of the display.

---

* When the disk is formatted, or data is written to it, the controller writes certain markers to the disk with code violations - it misses out certain clock transitions that should be present. When reading the data back it can get byte-synchronisation by finding these markers. The "missing clocks" code violation prevents it synchronising to any data elsewhere that might imitate the marker's bit-pattern.

9.9.1. Index mark

First ANALYSE looks for the Index mark (FC). This may well be absent, but ANALYSE will check with you first before deciding that there is no Index mark. During this check ANALYSE will home the head, seek back to the inner track, and re-read several times if it cannot locate the mark.

NOTE: If the data is garbled at the start of the buffer ANALYSE may find what it thinks is an Index mark. It is advisable to repeat this analysis several times to see if it is consistent.

9.9.2. Gap 1

Having found (or not found) the Index mark, ANALYSE then looks for the first sector header (FE). The distance between this and the start of the buffer (or the Index mark if it is present) is taken as Gap 1.

9.9.3. Gap 2

Next ANALYSE skips the sector header and its CRC bytes (6 bytes) and starts looking for the start of the associated data block (FB). This distance gives Gap 2.

9.9.4. Gap 3

ANALYSE then skips the data block and its CRC (Sector size + 2 bytes), and looks for the start of the next sector header (FE). This gives it Gap 3.

9.9.5. Physical sector numbers

Finally ANALYSE goes back to the start of the buffer and steps through looking for the sector headers (FE), and printing out the sector numbers it finds in the headers in the order that it finds them on the track.

This completes the operation of ANALYSE. You will now be offered the option of repeating the analysis (either on the same disk or a new disk in the same drive), or returning to the CP/M command level.

9.10. Manual Mode

In this mode ANALYSE performs track reads as described above, but just displays the contents of the track buffer on the screen for visual examination. This mode is still experimental, and is intended as a tool for the more knowledgeable user. An outline of its operation is given here.

There are several commands available:

The cursor up and cursor down keys may be used to move through the buffer. (Use the shift key in conjunction with them to increase the 'step' size).

The cursor left and cursor right keys will step the head in and out across the surface of the disk.

The 'R' key will re-load the buffer from disk.

The 'T' key will allow a specific Track number to be entered.

The 'S' key will toggle the side field between 0 and 1.

The 'D' key will toggle the Density between single and double.

The 'W' key will toggle the rotational speed of the MID5" drive between normal and high. (Think of W as omega - often used to designate angular velocity).

The 'A' key will toggle between a full display of the buffer contents and an abbreviated display. In the abbreviated display ANALYSE does not bother to display the contents of the data field of each sector, but it leaves the sector headers and inter-block gaps untouched.

9.10.1. The Manual Mode Display

The displayed buffer contains a complete byte-for-byte image of the track as it was read from the disk. This includes all the gaps, sector markers, sector headers, CRC check bytes, etc. As noted before in section 9.9, in an ideal world this command would execute perfectly, but in practice this is not always the case. The data is actually written on the disk as a serial bit stream, and the controller has to synchronise to the byte boundaries within this bit stream when reading the data back. It does this by synchronising to special markers at the start of each sector.

However, when executing a 'read track' command, the controller is in a 'transparent' mode of operation. In this mode it is effectively continually in a hunting state, looking for synchronisation patterns while reading (and passing back) the data from the disk. A side effect of this is that the controller may erroneously re-synchronise to the bit stream at any time.

It has been noted that various sequences of bytes in a sector can cause this re-synchronisation to happen, resulting in incorrect bytes being read in for the rest of that sector.

So, if in the Manual Mode display half a sector appears to contain garbage *do not worry*. The data in the sector on the disk is unlikely to be actually corrupt, and the controller will read the sector without error when executing a normal 'read sector' command.

The usual use the 'read track' display in the Manual Mode is to examine the contents of the sector headers and the lengths of the interblock gaps. The occasional *apparently* corrupt data block is of little consequence.

10. Entering New Formats to FORMATS.DAT

The following section describes, in a general way, the steps to be followed in adding a new format to the database. Further specific information will be found in the section describing the use of SETUP. Some understanding of CP/M disk structure and directory structure is necessary in order to follow some of the descriptions below.

To enter a new format on the system there are a variety of parameters that have to be determined. These fall into two categories: Physical and Logical. The physical parameters are those that the system needs to be able to read and write data to the disk. These are things such as the recording density, sector size, number of sectors per track etc. The Logical information is the information that the BIOS (and CP/M) need in order to read and write data to the correct areas of the disk. The latter are most easily determined from a running target system, but can be deduced with a bit of detective work given a disk from that target system.

10.1. Determining the Physical characteristics

Before the BIOS can read or write data to the disk it needs to know the following information:

Disk track density (48/96 tpi) - so the correct drive can be used.
Disk rotational speed (if appropriate).
Recording density (single/double).
Physical sector size (128/256/512/1024 bytes per sector).
First sector number. (0 or 1)
Number of sectors per track.
Number of tracks per side.
Number of sides in use (1 or 2).

and if you want to format disks as well as read and write them the following is also required:

Is an Index mark required?
What are the sizes of the inter-record gaps?
Are the sectors physically skewed on the disk?

If FORMAT is not going to be used, then arbitary figures may be entered for the Index/Gap/Skew parameters.

In the absence of any information on the disk format the program ANALYSE can be run. (See section 7). Unless you are extremely unlucky this should provide you with the essential information required.

10.2. Determining the Logical parameters

Having found out how data is recorded on the disk surface, we now have to determine exactly where it is placed. Some of this information is required by the BIOS, some the BIOS passes across to CP/M for use by the BDOS. What has to be determined now is:

If the disk is double-sided, does the system use side 0 of the disk in its entirity before starting on side 1? Or does it use track n/side 0, track n/side 1, track n+1/side 0, track n+1 side 1,....etc? The latter gives better system performance, but the former can provide compatibilty between single and double-sided formats if a hardware supplier offers systems with either single or double-sided drives.

How many tracks are reserved at the start of the disk for a CP/M system?

How many directory entries does the format allow for?

What is the "Block size" used by the system? (CP/M allocates space on the disk in fixed units or blocks. The minimum size supported is 1K, and the most likely values to be found are 1K, 2K or 4K, depending on the capacity of the disk. If the disk capacity exceeds 256k, then the minimum block size must be 2K.)

Is there a logical skew applied to the sectors, and if so what? If ANALYSE has shown that a Physical sector skew is in use, then it is probable that no logical skew is used. Conversely if no physical skew exists, (ANALYSE listed the sector numbers in correct numerical order), then a logical skew is likely to be in force.

If you have access to a system which supports the target format, then most of these parameters can be determined by doing a STAT DSK: (or equivalent if not CP/M80). This results in a print out like the following: (actually the GEMINI QDDS format).

```
   B: Drive Characteristics
6304: 128 Byte Record Capacity
 788: Kilobyte Drive  Capacity
 128: 32  Byte Directory Entries
 128: Checked  Directory Entries
 512: Records/ Extent
  32: Records/ Block
  40: Sectors/ Track
   2: Reserved Tracks
```

From this we can see that:
There are 128 Directory entries.
The Block size is 32 records = 32 * 128 bytes = 4k.
There are 40 sectors per track.
We already know that there are 10 512-byte **physical** sectors/track/side, which is equivalent to 40 **logical** 128-byte sectors. This implies that the format is likely to use one side

of the disk completely, before starting on the second. (N.B. Note that it only *implies* that this is so, there exist odd formats - such as one from RAIR - where the logical to physical translation goes:

```
Track 0 == track 0 side 0
track 1 == track 0 side 1
track 2 == track 1 side 0
track 3 == track 1 side 1
.........................
.........................
```

To cater for formats where a logical track does not correspond to a physical track, the "logical tracks" field of the "Format" should be set appropriately - see section 5).

Another advantage of the "STAT DSK:" listing is that it gives the total capacity of the disk. From this figure, together with the sectors/track and the reserved tracks figures, the total number of tracks on the disk can be calculated. From this you can determine, for example, whether a double-sided 48tpi disk comes from a drive with 35 tracks per side, or 40 tracks per side. (The latter is the more modern option).

If the format is unknown some arbitary values can be entered temporarily for the parameters. e.g.

128 directory entries
2k block size
'C' cylinder mode - use both sides
1 Reserved track

Also, importantly at this stage, no skew table should be entered.

Once the format has been entered option 3 of SETUP should be used to add it into the current BIOS (say as drive E). Next SETUP should be exited, and a disk patching utility such as DU* should be run. This program allows us to directly examine the sectors of the disk, and using it we can search for the directory, determine the number of reserved tracks, the number of directory entries, the logical skew factor in use, and finally, the system block size.

Ideally for this work the disk should hold several files, one of which should contain a reasonable amount of ASCII data. An ASCII file considerably simplifies the task of determining the sector ordering if any logical skew has been applied.

---

*DU.COM - available (with DU-V77.ASM) from the CP/M User group on VOL78. Also Sig/M (Amateur Computer Group of New Jersey) Volume 44

10.3. SUPERBRAIN format example

The following section will refer to the use of the disk patching utility DU.COM to examine a Superbrain QD format disk.

Initially you need a Superbrain QD disk containing some data, and failing such a disk to hand you can use your MFB system to produce it. Start off by using SETUP to configure your MFB system so that the Superbrain QD format is supported (using the existing data table entry) as one of your logical drives. Next insert a disk into the drive and use FORMAT to format it. Finally 'PIP' a series of files across onto the disk. We want about sixteen files on the disk, including a text file.

Now let us assume we have just been presented with the disk, but have been given absolutely no information about it - the worst possible situation.

1. Place the disk in one of the 5.25" drives and run ANALYSE. (See section 7). This should provide you with the following information on the disk's physical characteristics:

```
Drive rotational speed - normal
Double density.
48tpi
Double-sided.
First sector is 1.
10 sectors per track
35 tracks per side
No Index mark
Gaps of 44/38/35 (Gemini table)
No physical sector skew
```

2. Run SETUP and use option 1 to define a format - say TRIAL. Enter the data supplied by ANALYSE on the disk format. For the various other parameters enter some arbitary values, but DO NOT enter a sector skew table. (See example on the next page).

3. Save the entry and then use option 3 to set up a new BIOS and define drive E as TRIAL format.

4. Exit SETUP.

Before running DU it is advisable to draw up a logical-to-physical sector translation table as an aid in locating exactly which physical sector DU is currently displaying on the screen.

DU works in terms of CP/M logical sectors of 128 bytes each, and starts each track at sector 1. So for the Superbrain QD format the table would look like:

Logical:Physical	Logical:Physical	Logical:Physical
1 : 1	15 : 4	29 : 8
2 : 1	16 : 4	30 : 8
3 : 1	17 : 5	31 : 8
4 : 1	18 : 5	32 : 8
5 : 2	19 : 5	33 : 9
6 : 2	20 : 5	34 : 9
7 : 2	21 : 6	35 : 9
8 : 2	22 : 6	36 : 9
9 : 3	23 : 6	37 : 10
10 : 3	24 : 6	38 : 10
11 : 3	25 : 7	39 : 10
12 : 3	26 : 7	40 : 10
13 : 4	27 : 7	
14 : 4	28 : 7	

with sectors 41-80 being the corresponding physical sector numbers, but on the other side of the disk.

5. Run DU.

6. Log in Drive E. (LE)

7. Move to the start of the disk. (T0;S1)

6. Start displaying successive sectors until the first directory sector is found. (D;+;/)
(^S can be used to pause the listing, ^C to stop the search loop).
The display doesn't make any sense! No reasonable ASCII (like file names) ever turns up in the right hand side of the display, and the one important point to notice is that unwritten sectors display as a series of 1A bytes, rather than usual CP/M E5s. In fact 1A is the logical complement of E5, and so the data must be recorded in an inverted form on the disk. So..

7. Exit from DU (X) and run SETUP again.

8. Select option 1 and type in TRIAL to reload the trial format. Change the "Invted data (Y/N)" field from N to Y. Save the changed format. Use option 3 again to write the corrected information into the BIOS. Exit from SETUP.

9. Run DU again (as in step 6). This time the start of the directory sector should be found. Note down the track and sector number of it. (It should be Track 2, sector 1). This tells us that there are two reserved tracks.

```
================================================================================
        Gemini Multi-Format-BIOS Setup program     Version 3.3
 D = GEMQDDS     E = GEMDDDS     F = IBMPC8      G = IBM3740     H =
(0)  MID5"      (1)  RIGHT5"    (1)  RIGHT5"    (7)  EIGHT"
             Format Definition

       Format name.....: TRIAL              OS type (CP/M=0)..: CP/M
       Format description...: Attempt at Superbrain

Drive type (8/5/W): 5     Side mode(C/T/S/U): C     Index mark (Y/N)...: N
Tracks-per-inch...: 48                              Format byte........: E5
Tracks-per-side...: 40    Logical secs/trk..: *     First sector no....: 1
Sides-per-disk....: 2     Reserved tracks...: 1     Phys. sector skew..: 0
Sectors-per-track.: 10    Block size (k)....: 2     Gap 1 length.......: 44
Bytes-per-sector..: 512   Extent mask.......: *     Gap 2 length(34/43): 38
Density (S/D).....: D     Directory entries.: 128   Gap 3 length(>34)..: 35
Data rate (H/L)...: L                               Side fields........: *
Invted data (Y/N).: N                               Sector size field..: *
Reverse sides(Y/N): N                               Special init.......: N

Sector translate table : None


================================================================================
```

At this point it may be possible to determine the sector skew by examining the directory area. The first track (ignoring the system tracks), should contain the directory, and probably the start of the first file on the disk. There are four directory entries per logical sector, so with a physical sector size of 512 bytes we need 17 directory entries to ensure that at least two sectors of the directory are in use. Because there are four logical sectors to one physical sector we know that, for example, if logical sector 1 holds directory information, then so do logical sectors 2,3 and 4 as they are part of the same physical sector. So...

10. Starting at track 2 sector 1, display the contents of all sectors on the track, (T2;S1; then D;+;/80), and note down the contents of each physical sector (block of four logical sectors). Use the ^S (Control/S) key to pause and to restart the screen output while doing this. Your list should look like this:

Physical Sector	Contents
1	Directory
2	Part of a file
3	End of current directory
4	Part of a file
5	Unused (E5s)
6	Part of a file
7	Unused (E5s)
8	Part of a file
9	Part of a file
10	Part of a file
11	Unused (E5s)

```
                12        Unused (E5s)
                13        Unused (E5s)
                14        Unused (E5s)
                15        Unused (E5s)
                16        Unused (E5s)
                17        Unused (E5s)
                18        Unused (E5s)
                19        Unused (E5s)
                20        Unused (E5s)
Track 3          1        Part of a file
```

From this we can deduce the following:
By following the directory entries from the first to the third physical sector the sector skew must start 1,3,5,7,9,2..... The data file starts at sector 9, so with the skew implied above, the directory occupies 4 sectors, giving a total of 4 * 512 / 32 directory entries (=64). Also note that sectors 11-20, which lie on the second side of the disk, are unused. This implies that the format uses all of side 0, before starting on side 1. (If the disk holds a lot of files, then data will be found on the second side. In this case an ASCII file should be located, and it is a matter of determining whether the text appears amongst the data on the both sides of the disk, or only on several tracks of one side of the disk.)

11. Locate the ASCII file on the disk, and follow the text from one physical sector to the next to confirm that you have deduced the correct skew from the directory.

12. Go back and display the first directory sector (T2;S1;D). The display should look something like:

```
G=05:00, T=2, S=1, PS=0
00  00504950 20202020  20434F4D 0000003A  *.PIP     COM...:*
10  01020304 00000000  00000000 00000000  *................*
20  00535441 54202020  20434F4D 00000029  *.STAT    COM...)*
30  05060700 00000000  00000000 00000000  *................*
40  00442020 20202020  20434F4D 00000017  *.D       COM....*
50  08090000 00000000  00000000 00000000  *................*
60  00534554 55502020  20444F4D 01000080  *.SETUP   DOC....*
70  0A0B0C0D 0E0F1011  12131415 16171819  *................*
```

From this you can see that the first file starts at block number 1. (See Appendix C for a description of the CP/M directory format). In step 10 we determined that the directory consisted of four physical sectors, so as the directory only occupies one block, (block 0), the block size must be 512 * 4 = 2k.
NOTE: If the disk had been in use for a while, the file that used the first available block number might not be the first entry in the directory. The "M" (map) command of DU can be a help here, as it displays the block numbers in order and shows which files occupy which blocks. However this command

will only work properly if the data on the format that was used to set up the BIOS was correct, (or nearly correct), so at this stage this may or may not produce a useful listing.

13. Finally try to read a sector from a track >34. (e.g. T37;S1). This read should fail, indicating that only 35 tracks exist on the disk. (N.B. strictly this depends on the past history of the disk. At some time in the past it may have been used with a different format that was similar, but used 40 track drives. It is to answer questions like this, that the STAT DSK: information is useful.)

14. Now exit from DU (X) and go back into SETUP. Modify the data entry for TRIAL so that it now lines up with the information you have discovered.

15. Use option 3 to set up the BIOS again, this time using the correct parameters.

16. Exit SETUP and try DIR E:. If all is well you should see a correct directory listing without any strange occurrences (like blank files, multiple entries for the same file, or rubbish).

17. Finally TYPE the text file to the screen as a check, and try out (with care!) some of the COM files on the disk.

```
================================================================================
        Gemini Multi-Format-BIOS Setup program     Version 3.3
 D = GEMQDDS     E = TRIAL       F =            G =            H =
(0)  MID5"      (1)  RIGHT5"
             Format Definition

       Format name.....: TRIAL              OS type (CP/M=0)..: CP/M
       Format description...: Attempt at Superbrain

Drive type (8/5/W): 5     Side mode(C/T/S/U): S     Index mark (Y/N)...: N
Tracks-per-inch...: 48                              Format byte.......: E5
Tracks-per-side...: 35    Logical secs/trk..: *     First sector no....: 1
Sides-per-disk....: 2     Reserved tracks...: 2     Phys. sector skew..: 0
Sectors-per-track.: 10    Block size (k)....: 2     Gap 1 length.......: 44
Bytes-per-sector..: 512   Extent mask.......: *     Gap 2 length(34/43): 38
Density (S/D).....: D     Directory entries.: 64    Gap 3 length(>34)..: 35
Data rate (H/L)...: L                               Side fields........: *
Invted data (Y/N).: Y                               Sector size field..: *
Reverse sides(Y/N): N                               Special init.......: N

Sector translate table :  1  3  5  7  9  2  4  6  8 10


================================================================================
```

NOTE. DU was used initially with NO logical skew set in the data file. This was done as a logical skew can be confusing at that stage. If it had been set correctly then everything would have been fine, but if a mistake had been made it can be quite difficult to work out what physical sector you are actually looking at, and how the table should be modified to correct the error.

APPENDIX A

A. Incompatible formats

The Gemini GM849 disk controller card, (the basis of the MFB system), uses the Western Digital WD2793 floppy-disk-controller. The 2793 will handle most of the CP/M formats currently in use, but there are exceptions, which are detailed below.

A.1. Hard sectored disks

The controller cannot read or write hard sectored disks (e.g. North Star Horizon).

A.2. Non-standard recording technique

The controller cannot read or write disks that do not use FM or MFM encoding, together with the appropriate standard for header records and CRC polynomials. (e.g. APPLE disks cannot be read/written).

A.3. Unique hardware

The MFB does not support specialist hardware requirements. (e.g. The SIRIUS records disks at a constant bit density, and achieves this by linking the rotational speed of the drive to the track selected. By doing so it fits more sectors onto the outer tracks of the disk, but they can't be read in conventional constant speed drives.)

A.4. CP/M "compatible" formats

Some systems run CP/M compatible operating systems. The MFB will only be able to support these systems directly if they use the same directory structure as CP/M. For example the TORCH operating system, though billed as 'CP/M compatible', uses a totally different disk structure, and so requires a special utility if files are to be read/written from/to a TORCH disk.

A.5. Asymmetric track or sector ordering

The MFB does not currently support any format where the method of determining the next logical sector varies from track to track. So far two examples have been found:

Alphatronic P3 where the logical progression (expressed as track/side) is 0/0 1/0 0/1 1/1 2/0 2/1 3/0 3/1 ..... Here they have used an "S" type access on the first two tracks, followed by a "T" type access for the remainder of the disk.
(See READ.ME file for a work-around).

Cipher where the order of the sector numbers in the sector skew table changes from track to track. (By doing this Cipher get a slight increase in performance by making an allowance for the track-to-track stepping time of the disk drive, but from a software point of view it would have been simpler if this feature had been incorporated into their Format program, and excluded from the BIOS).

APPENDIX B

B. Incorrect CP/M formats

Several formats have been encountered where the disk parameter block has been set up incorrectly. (The error lies in the extent mask.) The net effect of this is to under utilise the directory entries for large files. Provision has been made in the format data tables used by SETUP to ensure that, if necessary, SETUP can construct an identical parameter block to that used by target system.

In the examples encountered so far, the extent mask has been set to 0 when a figure of 1 should have been used. Some visible symptoms of this error are:

Disk from target system EXM=0 running in MFB system with EXM = 1.	Disk written by MFB with EXM = 1 running in target system with EXM = 0.
In response to DIR command: File appears twice if size>16k.	In response to DIR command: File does not appear if size>16k

.....Files over 16k will not be copied correctly.....

APPENDIX C

C. CP/M directory structure

Each CP/M directory entry consists of 32 bytes. These are used as follows:

byte

0 — Set to E5 if there is no valid entry, otherwise holds the User number (00-1Fh) that the file was created under.

1-8 — Hold the first part of the file name in ASCII.

9-11 — Hold the extension of the file name (e.g. COM). If the $SYS or $R/O attributes are set, then these are reflected in the most significant bit of bytes 9 and 10. (Newer versions of CP/M use the msb of byte 11 as an 'archive' indicator).

12 — Holds the largest extent number of the entry in bits 4-0.

13 — Not used.

14 — Holds the 4 high order bits of the extent number in bits 3-0.

15 — Holds the number of records (sectors) in this extent. (0-128)

16-31 — Holds the numbers of the disk blocks occupied by the file. If the total disk size is >255 blocks, then a maximum of eight 16-bit block numbers can be held. If the total disk size is <256 blocks, then a maximum of sixteen 8-bit block numbers can be held.

e.g. Entry from a system with a 2k block size and <256 blocks.

```
00 57 53 20 20 20 20 20  20 43 CF 4D 01 00 00 0A  *.WS      COM....*
02 03 04 05 06 07 08 09  0A 00 00 00 00 00 00 00
```

File WS.COM, in User Area 0, $SYS attribute set, extents 0 & 1 open, occupying blocks 2->0Ah. Total size is 128+10 sectors. (128 records in extent 0, 10 records in extent 1).

Same file on a system with 4k block size and >256 blocks.

```
00 57 53 20 20 20 20 20  20 43 CF 4D 01 00 00 0A  *.WS      COM....*
10 00 11 00 12 00 13 00  14 00 00 00 00 00 00 00
```

details as above except blocks utilised are 10H->14H.

APPENDIX D

D. .INI file structure

This appendix gives the structure of the .INI files that FORMAT uses (if the 'Special Init' field is set to 'Y') to initialise disks once they have been formatted.

The first 128 bytes of the file are assumed to contain a list of the areas of the disk that are to be initialised. The first byte is taken to be a count of the number of areas to be initialised. This is then followed by a two-byte entry for each area in the form:

Physical Track number Physical side (0 or 1)

The remainder of the file is assumed to be a sequential image of the data that is to be written to the various areas of the disk. Data should be included for each area that is to be initialised.

For example consider a format of:

96tpi double-sided drive with 8 512byte sectors/track

that requires special information to be written into the first sector of the disk, and the last sector of the disk. The file would start with the byte sequence (shown in hexadecimal format):

02 00 00 50 01.....

(ie 2 entries, track 0/side 0, and track 80/side 1).

The data to be written to the first area starts at the second logical sector in the file, and continues for 4k (a total of 32 logical sectors. 8 * 512 = 4k per side). The data to be written to the second area starts at the 34th logical sector - immediately following that of the first area - and also extends for 4k.

Each of the data areas are written sequentially to the disk starting at the first sector of the specified track/side combination, and completing with the last sector of that track/side.