

Some Helpful Hints on Developing a Disk Definition from Scratch

If possible, run programs on the CP/M system to help get the necessary information for building a new definition. The following files may be transferred by RS-232C if a Terminal Program is available, or by PIPMODEM.COM (PIPMODEM.DOC & PIPMODEM.ASM)

1. CPMADR.COM - Finds addresses used by the CP/M system.
2. DPB.COM - Finds Disk Parameter Block information.
DPB.BAS - Same, but runs under MBASIC. (Located in 22DISK.TXT)
3. PROBE.COM - Probes CP/M System and Drive for information.
Note: The previous four programs may not locate and display accurate information for some CP/M systems.
4. SKEW.COM - Finds disk SKEW information.
5. Execute the following "STAT" commands on system in question.
Note: "Control P" will allow a hard copy of the information.

STAT DEV: - Displays current DEVICE information.
STAT VAL: - Displays current STATUS information.
STAT USR: - Displays current USR information.
STAT A*.* \$S - Displays detailed file information.
STAT DSK: - Gives some detailed disk information for logged drive.
STAT A: DSK: - Gives some detailed disk information for A:.
STAT B: DSK: - Gives some detailed disk information for B:.

If you have a functional CP/M system, you can use DDT to find the Disk Parameter Block Address, the Allocation Address, along with the information stored there for the logged floppy drive.

1. Log the floppy drive you wish to locate the DPB data for. I will use A: for this example
2. Run DDT and enter the code below:

```
A: <Carriage RETURN>
DDT
-f0100,0200,00      #fill memory with zero's
-A0100             #assemble program at 0100
0100      mvi    c,0f  #0f in C for Open file
0102      lxi    d,005c #FCB in reg DE
0105      call   5     #call BDOS
0108      rst   7     #back to DDT
0109      mvi    c,1b  #1B in C for Get Allocation Address
010B      call   5     #call BDOS
010E      rst   7     #back to DDT
010F      mvi   c,1F   #get disk parameter address
0111      call  5     #HL will contain the address
0114      rst   7     #back to DDT
0115      nop     #
0016      .       #type PERIOD to EXIT Assembler
```

```

-d0100  #display memory at 0100
 0100   0e0f115c00cd0500FF0E1BCD0500FF0E1FCD0500FF
-g0100  #execute program
*108    #program exits here after file search
-g0109  #execute programs
*010E   #program exits here with Allocation Address
-g010F  #execute programs
*0114   #program exits here with Disk Parameter Address
-xH     #display DPB that is in reg HL
        #<CR> will exit register modify
        #CNTL C will exit DDT and save 1 test101.ddt will save the file
        #save n filename.com saves 1 256 byte block of data.

```

The contents of register HL is the address of Drive A: DPB. If you display that address (mine was F5E3) you will see F5E3: 2400040F0154007F00C00020000200.

The Disk Parameter Block (DPB) for each drive is:

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|SPT.|BSH..|BLM..|EXM..|DSM.. |DRM.|AL0..|AL1..|CKS..|OFF.|
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
 16B 8B   8B   8B   16B  16B  8B   8B   16B  16B

```

RPT = 0024 = The number of 128 byte records per track.
BSH = 04 = The block shift count.
BLM = 0F = The block mask.
EXM = 01 = The extent mask.
DSM = 0054 = Disk storage maximum (the largest block number).
DRM = 007F = Directory maximum (the largest directory entry).
DAB = C000= Directory Allocation Block AL0: = C0 & AL1: = 00
CKS = 0020 = Directory check size.
OFF = 0002 = Track offset (number of reserved tracks).

SPT: Number of sectors per track. May differ from RPT if physical sectors are other than 128 bytes in size.

SSZ: Sector size code:

```

 0 = 128 bytes per sector
 1 = 256 bytes per sector
 2 = 512 bytes per sector
 3 = 1024 bytes per sector

```

NTK: Number of tracks (35, 40, 77, or 80). The number of tracks can be determined by formatting a Floppy, or with Anadisk. Anadisk will also tell you the sector size and SKEW.

Quite frequently, the only information available when developing a definition for 22DISK is a diskette with a few files and nothing more.

Here's a guide on what to do if you'd like to try working up a definition yourself:

1. Get a copy of Sydex's ANADISK. This utility will furnish far more information than will any other program. A typical disk definition follows:

```
BEGIN AMP4 Ampro - DSDD 96 tpi 5.25"
DENSITY MFM,LOW
CYLINDERS 80 SIDES 2 SECTORS 5,1024 SKEW 2
SIDE1 0 17,18,19,20,21
SIDE2 1 17,18,19,20,21
ORDER SIDES
BSH 4 BLM 15 EXM 0 DSM 394 DRM 255 AL0 0F0H AL1 0 OFS 2
END
```

New 22DISK definitions require the following information:

```
DENSITY xx,xx
CYLINDERS x
SIDES xx
SECTORS x,xxxxx
SKEW x
SIDE1 0 x,x1,x2,x3, , ,xn
SIDE2 x x,x1,x2,x3, , ,xn
ORDER x
BSH x
BLM x
EXM x
DSM x
DRM x
AL0 x
AL1 x
OFS x (or SOFS x)
COMPLEMENT (unlikely)
```

New LIBDSK definitions require the following information:

[title]

description = DESC The description of the format as shown by (for example) dskform-help.

sidedness =TREATMENT How a double-sided disk is handled. This can either be alt

(sides alternate – used by most PC-hosted operating systems),

outback (use side 0 tracks 0-79, then side 1 tracks 79-0 – used

by 144FEAT CP/M disks), or outout (use side 0 tracks 0-79, then

side 1 tracks 0-79 – used by some Acorn formats). If the disk is

single-sided, this parameter can be omitted.

cylinders = COUNT Sets the number of cylinders (usually 40 or 80).

heads = COUNT Sets the number of heads (usually 1 or 2 for single- or double- sided).

sectors = COUNT Sets the number of sectors per track.

secbase = NUMBER Sets the first sector number on a track. Usually 1; some Acorn formats use 0.

secsize = COUNT Sets the size of a sector in bytes. This should be a power of 2.
 datarate = VALUE Sets the rate at which the disk should be accessed. This is: HD, DD, SD or ED.
 rwgap = VALUE Sets the read/write gap.
 fmtgap = VALUE Sets the format gap.
 fm = Y or N Sets the recording mode - Y for FM, N for MFM.
 multitrack = Y or N Sets multitrack mode.
 skipdeleted = Y or N Sets whether to skip deleted data.

The LIBDSK Data rate will be one of:

RATE_HD, /* Rate for High-density disc - 1.2Mb in 5.25" 96 tpi drive, or 1.44Mb in 3.5" 96 tpi drive */
 RATE_DD, /* Rate for Double-density disc - 360k in 5.25" 48 tpi drive, or 720K in 3.5" 48 tpi drive */
 RATE_SD, /* Rate for Double-density disc - 180k in 5.25" 48 tpi drive, or 360k in 3.5" 48 tpi drive */
 RATE_ED /* Data rate for 2.8Mb 3.5" in 3.5" 96 tpi drive */

New CPMTTOOLS definitions require the following information:

```

diskdef title
  seclen xxx    #= Sectors xx,1024
  tracks xx     #= (Cylinders * Sides) = 80*2 = 160
  sectrk xx     #= Sectors 5,xxx
  blocksize xxxx #= (128*(BLM+1)) = 2048
  maxdir xxx    #= (DRM+1) = 256
  skew x        #= may be 1 thru 6, or so
  boottrk x     #= OFS = 2
  os x.x        #= 2.2, or 2, or 3
end
  
```

So, if you know the 22DISK parameters, you can easily fill in the details for LIBDSK & CPMTTOOLS. As an example, here is an AMPRO LITTLE BOARD DSDD 96 tpi 5.25" Definition for all three software packages.

```

BEGIN AMP4 Ampro - DSDD 96 tpi 5.25"
DENSITY MFM,LOW
CYLINDERS 80 SIDES 2 SECTORS 5,1024 SKEW 2
SIDE1 0 17,18,19,20,21
SIDE2 1 17,18,19,20,21
ORDER SIDES
BSH 4 BLM 15 EXM 0 DSM 394 DRM 255 AL0 0F0H AL1 0 OFS 2
END
  
```

```

[amp4]
description = Ampro - DSDD 96 tpi 5.25"
sides=alt      #=/* XXX Provisional depending on 22DISK Definition */
cylinders = 80 #= Cylinders = 80
heads = 2      #= Sides = 2
sectors = 5    #= Sectors 5,xxx
secbase = 17   #= First Sectors from (SIDE1 0 17)
  
```

```

secksize = 1024  #= Sectors xx,1024
datarate = DD   #= Data Rate = SD, DD, HD, ED
rwgap = 12      #=/* XXX Provisional */
fmtgap = 23     #=/* XXX Provisional */
fm = N         #= FM or MFM (Likely MFM)
multitrack = N  #= Most likely N
skipdeleted = Y #= Most likely Y

```

```

diskdef AMP4
  seclen 1024  #= Sectors xx,1024
  tracks 160   #= (Cylinders * Sides) = 80*2 = 160
  sectrk 5     #= Sectors 5,xxx
  blocksize 2048 #= (128*(BLM+1)) = 2048
  maxdir 256   #= (DRM+1) = 256
  skew 0       #= may be 1 thru 6, or so
  boottrk 2    #= OFS = 2
  os 2.2       #= 2.2, or 2, or 3
end

```

2. Run ANADISK on the diskette in SECTOR EDIT mode with the display set for "HEX". Note that the interleave (or SKEW) is given, as well as the number, size and addresses of sectors. You now have data for: DENSITY, CYLINDERS, SIDES, SKEW SIDE1 and SIDE2.
3. Locate the directory on the diskette. Almost invariably, it starts on the first sector of a track. If for some reason it can't be found on the first sector of a track, try using the SEARCH feature to find a "FILE" name that exists on the disk. The number of tracks or sectors up the point where the directory begins will give the figure for OFS or SOFS. The directory stands out because it shows file names every 32 bytes, or every other line on the ANADISK display.

Directory was located at Cylinder 0, Side 0, Sector 2:

```

0000 00 32 32 44 49 53 4b 20 20 44 4f 43 00 00 00 80 .22DISK DOC....
0010 02 00 03 00 04 00 05 00 06 00 07 00 08 00 09 00 .....
0020 00 32 32 44 49 53 4b 20 20 44 4f 43 01 00 00 80 .22DISK DOC....
0030 0a 00 0b 00 0c 00 0d 00 0e 00 0f 00 10 00 11 00 .....
0040 00 32 32 44 49 53 4b 20 20 44 4f 43 02 00 00 80 .22DISK DOC....
0050 12 00 13 00 14 00 15 00 16 00 17 00 18 00 19 00 .....
0060 00 32 32 44 49 53 4b 20 20 44 4f 43 03 00 00 80 .22DISK DOC....
0070 1a 00 1b 00 1c 00 1d 00 1e 00 1f 00 20 00 21 00 ..... !.
0080 00 32 32 44 49 53 4b 20 20 44 4f 43 04 00 00 08 .22DISK DOC....
0090 22 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ".....
00a0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00b0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00c0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00d0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00e0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00f0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0100 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0110 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0120 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0130 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0140 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....

```

```

0150 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0160 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0170 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0180 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0190 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01a0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01b0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01c0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01d0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01e0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01f0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....

```

Fig 1

Directory continues on Cylinder 0, Side 0, Sector 3:

```

0000 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0010 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0020 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0030 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0040 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0050 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0060 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0070 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0080 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0090 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00a0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00b0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00c0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00d0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00e0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
00f0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0100 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0110 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0120 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0130 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0140 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0150 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0160 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0170 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0180 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
0190 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01a0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01b0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01c0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01d0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01e0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....
01f0 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 e5 .....

```

Fig 2

- Examine the first directory entry. The second 16 bytes (the second line of the entry) gives a list of allocation unit numbers (AUN) belonging to the file. The first one of these will tell how many allocation units have been reserved to hold the directory. Thus, if the first allocation byte is 04, four allocation units (AL0..AL3) have been allocated to the directory. Another thing to note is whether the allocation unit number appears to be an 8 bit or 16 bit quantity. For example, if an entry has the following form:

00 50 49 50 20 20 20 20 20 43 4F 4D 00 00 00 10 .PIP COM....
04 00 05 00 06 00 00 00 00 00 00 00 00 00 00 00

the allocation unit designators are 4, 5 and 6 and are 16 bit quantities. On the other hand, the following item:

00 50 49 50 20 20 20 20 20 43 4F 4D 00 00 00 10 .PIP COM....
04 05 06 00 00 00 00 00 00 00 00 00 00 00 00 00

shows the same allocation units, but as 8 bit quantities. If 16 bit quantities are involved, more than 255 allocation units are defined for the diskette, if an 8 bit quantity, less than 256 units. This serves as a check for later computations. In any case, the first allocation number of the first directory entry gives the number of bits to be set in AL0 and AL1.

EXAMPLES:

Allocation Unit Number (AUN)
02 AL0 11000000B AL1 0
04 AL0 11110000B AL1 0

The following information shows a detailed breakdown of all the information in the the first and second directory entries.

00 - Valid file (E5 for Erased File)
32 32 44 49 53 4b 20 20 44 4f 43 - File name 22DISK DOC (11 bytes)
00 - Extent (used for large files)
00 - S1 (Digital Research Reserved)
00 - S2 (Digital Research Reserved)
80 - Number of records in this extent
02 00 03 00 04 00 05 00 06 00 07 00 08 00 09 00 - allocation units

Note: 02 is the first AUN.

00 - (Continuation of above file)
32 32 44 49 53 4b 20 20 44 4f 43 - File name 22DISK DOC (11 bytes)
01 - Extent (used for large files)
00 - S1 (Digital Research Reserved)
00 - S2 (Digital Research Reserved)
80 - Number of records used
0a 00 0b 00 0c 00 0d 00 0e 00 0f 00 10 00 11 00 - allocation units

Note: Unused allocation units will be 00.

5. Starting with the first directory sector, advance the sector (and track, if necessary) until you find the data belonging to the first file. Unused directory entries are almost always filled out with a pattern of E5's. The number of sectors dedicated to the directory will give an indication of both the number of directory entries and the allocation unit size.

First Program located at Cylinder 0, Side 1, Sector 2:

NOTE: (This was the 9th sector displayed, so 8 are reserved for directory)

```

0000 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 20 20 20 20 .....
0010 20 20 53 79 64 65 78 0d 0a 0d 0a 0d 0a 0d 0a 0d  Sydex.....
0020 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d .....
0030 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 20 .....
0040 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0050 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0060 20 20 20 20 20 32 32 44 49 53 4b 0d 0a 0d 0a 0d 22DISK.....
0070 0a 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 .
0080 20 20 20 20 20 41 20 43 50 2f 4d 2d 74 6f 2d 44 A CP/M-to-D
0090 4f 53 20 44 69 73 6b 65 74 74 65 20 49 6e 74 65 OS Diskette Inte
00a0 72 63 68 61 6e 67 65 20 55 74 69 6c 69 74 79 0d rchange Utility.
00b0 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d .....
00c0 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d .....
00d0 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 20 .....
00e0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
00f0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0100 20 20 20 20 20 20 20 20 20 20 20 20 20 20 53 79 64 Syd
0110 65 78 0d 0a 20 20 20 20 20 20 20 20 20 20 20 20 20 20 ex..
0120 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0130 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0140 20 20 31 35 33 20 4e 6f 72 74 68 20 4d 75 72 70 153 North Murp
0150 68 79 20 41 76 65 2e 0d 0a 20 20 20 20 20 20 20 20 hy Ave...
0160 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0170 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
0180 20 20 20 20 20 20 20 53 75 6e 6e 79 76 61 6c 65 Sunnyvale
0190 2c 20 43 41 20 20 39 34 30 38 36 0d 0a 20 20 20 , CA 94086..
01a0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
01b0 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
01c0 20 20 20 20 20 20 20 20 20 20 20 20 28 34 30 38 29 (408)
01d0 20 37 33 39 2d 34 38 36 36 0d 0a 0d 0a 0d 0a 0d 739-4866.....
01e0 0a 0c 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a 0d 0a .....
01f0 0d 0a 0d 0a 0d 0a 0d 0a 20 20 20 20 20 20 20 20 .....

```

Fig 3

Suppose that a our hypothetical diskette shows 8 sectors of 512 bytes allocated to the directory. Since a directory entry is 32 bytes long, the number of directory entries possible for this format is:

16 entries per sector X 8 sectors = 128 entries total
or
((Sector Size in Bytes/32) X Sectors to DIR) -1 = DRM

Thus, DRM is one less than this or 127. Since we know that these 8 sectors represent 2 allocation units or 4,096 bytes, the Allocation Unit Size (AUS) is:

4,096 / 2 = 2,048 bytes
or
((Sector Size in Bytes) X Sectors to DIR) / AUN = AUS

Given this, we can use the following table to compute the values for BSH and BLM:

Unit Size	BSH	BLM
1,024	3	7
2,048	4	15
4,096	5	31
8,192	6	63
16,384	7	127

7. We still need a value for EXM and DSM. DSM is simply the total number of allocation units on the diskette, not counting the system OFS area. Let's assume that our diskette has an offset of 2 tracks and is single-sided with 40 cylinders with 8 sectors per track. The value of DSM is then:

$$((40 \text{ cylinders} \times 1 \text{ side}) - 2 \text{ tracks}) \times 8 \text{ sectors} \times 512 \text{ bytes} = 155,648 \text{ bytes} / 2,048 = 76 \text{ Allocation units, total}$$

To get EXM, we then use the following table:

Unit Size	DSM < 256	DSM > 255
1,024	0	na
2,048	1	0
4,096	3	1
8,192	7	3
16,384	15	7

8. That's almost the complete picture, but for the SIDE1 and SIDE2 sector ordering. This is the hardest to determine of any of the parameters. Find a readable text file on the diskette and piece the text back together by observing how words break across sector boundaries. After a bit, a pattern in the ordering will become apparent, such as:

1, 3, 5, 7, 2, 4, 6, 8

9. The ORDER value can be determined on double-sided diskettes by observing whether file data appears to move from one track to the next on the same side (CYLINDERS), or appears to go from one side to the other and then advance to the next cylinder (SIDES).
10. COMPLEMENT is rare. Generally, if a diskette appears to be formatted with values of 1A instead of E5, the data on the diskette is complemented.

11. Last resort for information will require a memory dump from F200 hex through FFFF to provide all the DBP information as hex bytes. It will be necessary to dig through 3584 (E00) bytes of data to locate the DPB information. Most systems follow the layout shown below.

(Hex values starting in memory at Fxxx)

- RPT: (word) total # of logical 128-byte sectors/track
BSH: (byte) data allocation block shift factor,
determined by data block allocation size
1k=3 2k=4 4k=5 8k=6 16k=7
BLM: (byte) data allocation block mask (2**[BSH-1])
1k=7 2k=15 4k=31 8k=63 16k=127
EXM: (byte) extent mask, determined by the data block
allocation size and the number of disk blocks
1k=0 2k=1 4k=3 8k=7 16k=15 if DSM < 256
1k=n 2k=0 4k=1 8k=3 16k=7 if DSM > 256
DSM: (word) disk size in blocks-1; determines the total
storage capacity of the disk size
DRM: (word) directory size-1; determines the total number
directory entries that can be stored on this
drive. DRM: = ((# dir entries) / 4) -1
AL0: (byte) alloc0 (1100 0000) 2 blocks/dir
(1000 0000) 1 block/dir
AL1: (byte) alloc1 (0000 0000)
Note that AL0: and AL1: are taken together
as a single sixteen-bit mask for the number
of blocks of directory allocation.
(1100 0000 0000 0000) 2 blocks/dir
(1000 0000 0000 0000) 1 block/dir
CKS: (word) directory check vector size
fixed media: CKS = 0
removable media: CKS = (DRM + 1)/4
OFF/SOFF: (word) # of reserved system tracks or sectors (offset)
at the beginning of the (logical) disk.
Usually the directory begins at the first
sector of the track defined by this number.

By formatting a disk the number of tracks can be determined, which will give the first byte to look for. The SPT will be followed by 00 and the value for BSH and BLM will be the third and fourth bytes.

BSH and BLM will be one of the following Hex values:

BSH 3 or 4 or 5 or 6 or 7
BLM 7 or 0F or 1F or 3F or 7F

The above information should be fairly easily to locate.

Updated - 5 May 2018