AllDisc : A Variable Format Disc Input/Output Subsystem

(Version 1.2)

For NasBus/80-Bus computers

Disclaimer

Whilst considerable effort has been made to ensure that AllDisc will function as described in this document, its use is entirely at the risk of the purchaser. AllDisc is an extremely complex and flexible program for transferring data using a variety of hardware and exposes the user to greater risks of loss or damage than is normal. The author can accept no liability for any loss or damage caused by using AllDisc.

1. What it is

AllDisc is a set of substitute disc drivers for those in any standard CP/M system and will work with any of the standard NasBus/80-Bus Floppy Disc Controllers (Lucas Logic, GM809/829, MAP-80). Different FDCs can be mixed as long as their base port addresses do not clash. It is also possible to drive a GM833 RAMDisc card, but this version does not yet support a hard disc. It may not necessarily function with non-standard CP/M systems such as those which include CCPZ, ZCPR, BDOSZ or SYS.

AllDisc can read and write a wide variety of disc formats including those of most of the popular CP/M computers as long as the drives are suitable and the the disc format (soft-sectored only, IBM formatting protocols) is suitable.

An archive file of popular formats can be maintained for easy recall.

If 96 tpi drives are in use then it can also read and write 48 tpi discs. However, the facility to write 48 tpi discs on 96 tpi drives should be used with discretion. Although the facility is provided for convenience and, indeed, will often work well, you may find difficulties in reading that disc on a 48 tpi drive.

It is just over 3K in length, including a 1K sector read/write buffer, and sits below the CCP. It is located on loading by examining the BDOS jump address at location 0006H which is then overwritten to ensure that the new drivers do not lie in the TPA. The TPA is thus reduced in size by just over 5K (since the CCP takes up 2K). In order that all facilities can be fully used, a minimum TPA corresponding to a 46K CP/M system is required before loading AllDisc.

If you do not understand the abbreviations CCP, BDOS and TPA then do not worry. Just follow the instructions which follow in the next sections.

2. Getting started

The disc supplied contains the following files:

ALLDISC.COD	ALLDISC.COM	OLDBIOS.COM	VFORMAT.COM
FORMATS.DAT	CONFIGDR.GEM	CONFIGDR.NAS	MCLEAR.COM
DU.COM	DU.DOC	CRCK.COM	CRCKLIST.CRC

CRCK.COM is a program to check the integrity of all the files on the disc. It is a public domain program from the CP/M User Group that performs CRC checking and is provided free of charge. To use it you should enter the command "CRCK d:*.*" where d is the drive containing your AllDisc disc. Then compare to the CRC totals in CRCKLIST.CRC which can be viewed by "type CRCKLIST.CRC" or sending to the printer. It is recommended that CRC checking is provided on all disc made up using AllDisc to enable verification of the copied files on the destination computer.

If you are running a Gemini video card then REName CONFIGDR.GEM to CONFIGDR.COM. If on a Nascom/Lucas video system then REName CONFIGDR.NAS to CONFIGDR.COM. This is simply so that the 'Clear to end of screen' (ClrEOS) and 'Clear to end of line' (ClrEOL) codes function correctly. Otherwise the two files are identical. If you are running any other video system then you can select either and patch locations 0123-012BH to your ClrEOS sequence and 012C-0134H to your ClrEOL sequence. In both cases, the first byte contains the number of bytes in the sequence. Alternatively, you can patch the call addresses at 0139H (ClrEOS) and 013CH (ClrEOL) to your own routines in the area 0140-01BFH. If your screen display is not 25*80 then you will also need to patch location 010AH to your screen width and location 010BH to your screen height.

ALLDISC.COD contains the relocatable code for the disc drivers. It should come set up so that drives A and B correspond to the format of the disc on which it is supplied. (If it is not, or you wish to configure it for a different format then you will need to run CONFIGDR before loading it, carrying out the initial system configuration from the F(ile option rather than the more usual M(emory option. See the next section.)

ALLDISC.COM is the means by which AllDisc is invoked. Typing ALLDISC in response to the CP/M prompt will load ALLDISC.COD into memory at the top of your TPA. Once AllDisc is in place and configured for the discs currently in use, then CP/M should function exactly as before <u>in most circumstances</u> except for a reduction in available RAM by just over 5K. ALLDISC.COD must lie on the current default drive.

OLDBIOS.COM is the means by which you can return to your original BIOS. Typing OLDBIOS at the CP/M prompt will remove AllDisc from operation.

CONFIGDR.COM is the command (used only when AllDisc is in place) which enables you to reconfigure the drives to different formats and different FDC cards. It is fully described in the next section.

MCLEAR.COM is a simple program to erase the directory on drive M: which should normally be the GM833 RAMDisc. This is because it may start up with rubbish in the directory which would prevent its proper functioning. MCLEAR simply writes 'E5' bytes into the directory so that it appears as a freshly formatted disc. It should only be used when AllDisc is in operation and with

the default directory size for RAMDisc.

DU.COM has nothing to do with AllDisc but is a public domain program by Ward Christianssson of the US CP/M User Group. It is supplied free as a utility to help in examining difficult formats or reading and writing individual 128-byte records. See the appendix for some examples of its use.

DU.DOC is a text file containing instructions on how to use DU.

The file FORMATS.DAT contains archived formats which can be recalled or added to by CONFIGDR. It must lie on the current default drive if the A(rchive option is used (see next section).

The command VFORMAT will format a disc in the specified drive according to the format for which that drive is currently configured. You can specify the gap3 size between sectors, and a physical sector skew if you wish. If you specify too large a gap3 then you may have trouble with the first or last sectors. Too small a gap3 may result in difficulty reading and writing on another computer. The gap3 size defaults to the minimum value (10 single density, 24 double density).

When formatting a 48 tpi disc on a 96 tpi drive, take care to use a completely blank disc. If the disc had been previously formatted at 96 tpi, then a 48 tpi drive may pick up garbage.

As each track is written every sector is verified in turn. If a read fails, then a message is written to that effect and the contents of the ports X0-X5 (where X0 is the base port of the controller card in use) are displayed in hex. After a fixed number of retries (see next section) the track is rewritten up to five times. If it still proves impossible to verify a particular track then the message 'Bad track' is written and the program moves on to the next track.

Double sided discs are formatted one side at a time, independently, whatever the logical to physical translation used for the second side.

3. Configuration

Throughout CONFIGDR, a tree of menu prompts is used. These take the form of a series of commands listed on the top line of the screen. The relevant command is selected by typing the first letter of the command (which is separated by a '(' character). The Escape key, marked ESC, will take you back from the current command level to the previous level except for the final exit from CONFIGDR, when Q for Q(uit is used to prevent an accidental exit without saving by pressing <ESC> one time too many.

CONFIGDR will configure the drives either in a current active AllDisc in memory or in the ALLDISC.COD file on disc. In the latter case, the new version is saved in ALLDISC.NEW and can be RENamed ALLDISC.COD for future use when required. The choice between editing F(ile or M(emory is made in response to the very first prompt when CONFIGDR is executed. You cannot change this choice except by Q(uitting and re-executing CONFIGDR. If the F(ile option is used then ALLDISC.COD must lie on the default drive.

After responding to the initial prompt, the drive parameters are immediately loaded from file or memory into a set of internal variables and a new prompt line appears with the following options:

3.1 G(et

will reload the drive parameters discarding any unS(aved changes.

3.2 S(ave

will save any new parameters set by the E(dit command (see 3.3). If a F(ile configuration is in progress, the result will not be written to disc until CONFIGDR is Q(uit.

3.3 E(dit

produces a new prompt line with the options:

3.3.1 D(rive

enables you to select a drive from 1 to 6. The current parameters associated with that drive are reported. These do not necessarily correspond to any specific CP/M drive identifier A to P. If they do it appears as parameter W. These parameters are generally self-explanatory but are explained in more detail in subsection 3.3.2.

3.3.2 C(hange

enables you to change any drive parameters one at a time once a drive has been selected. Familiarity with the FDC documentation and CP/M disc structure will be helpful. Type the key letter (A-W) associated with the parameter you wish to alter and then the new value. <ESC> will return you to the E(dit prompt line.

Parameter A is the drive number (1-6) of the currently selected drive.

Parameters B-C are the logical tracks per disc and sectors per track. On single-sided discs, these will be the physical parameters. On double-sided discs, one of these two parameters will be double the single-sided value (see parameters L & M). The number of sectors per

track cannot exceed 64.

Parameter D is the physical sector size in bytes. This should not be confused with the "record" size of 128 bytes used internally by CP/M. Software documentation (including DU and that from Digital Research) very often confuses the two and is a hang-over from the days when they were the same. You will have to use your common sense as to which is being referred to.

Parameters E-H are specific to the particular implementation of CP/M. A CP/M 'block' (also sometimes referred to as a 'group') contains eight times the 'Block size' (measured in Kbytes) 128-byte 'records'. 'Dir Blocks' is the number of blocks allocated to the directory. 'Reserved tracks' specifies the number of tracks skipped at the beginning of the disk before the directory. 'Skew' is a number between 0 and the number of sectors per track less two. It specifies how many sectors to skip when locating the next logical sector on the same track. It operates in addition to any formatted skew. If it is non-zero then the logical to physical translation table is displayed. (If you have only a 16 line display, then this may not be complete.) On discs which use the 'cylinder' method to access the flip side (see parameter L), it is assumed that each side is separately skewed. If you come across a format where the skewing mixes both sides, please let me know.

Parameter I can be 'S' for single (FM) or 'D' for double (MFM) density recording format.

Parameter J if 'T' then data will be complemented (a la SuperBrain) before reading or writing. Should be 'F' for normal operation.

Parameter K will always be '0' for single-sided discs. Double-sided discs can have '0', '1' (so that each side can be a separate logical disc) or 'B' for both. If you have single-sided drives then you will be able to read and write the first side of a double-sided disc as long as it does not use the 'cylinder' method (see parameter L) for accessing the flip side.

Parameter L is relevant only if K is 'B'. If L is 'S' then 'sector' mode is in operation. This means that parameter C (logical sectors per track -- SPT) will be twice the number of sectors per track on a single side and sector numbers greater than SPT/2 will be found on the flip side. This is sometimes known as the 'cylinder' method -- a 'cylinder' being a pair of tracks, one on each side, at the same head position. If L is 'T' then parameter B (tracks per disc -- TPD) will be twice the number of tracks per side and track numbers greater than or equal to TPD/2 will be found on the flip side.

Parameter M is relevant only when K is 'B' and L is 'T'. If M is 'N' for normal then logical track TPD/2+T is physical track T on the flip side (as used by Gemini QDDS) and if M is 'R' for reverse then logical track TPD/2+T is physical track TPD/2-T-1 on the flip side (as used by Lucas Logic).

Parameter N can be either 0 or 1. This is the sector number of

the first sector ID on each track. Nascom normally use 1, Gemini normally use 0.

Parameter O is only reported if a Gemini controller is in use, and is only relevant if it is a GM829. It indicates whether or not the 8" bit (bit 5) of the drive select byte (for a higher clock rate) is to be set. It may be either 'T' or 'F'.

Parameter P indicates whether a Nascom ('N') or Gemini G809/829 ('G') FDC card is in use for this drive. A MAP-80 card operates as a Gemini card and should be specified as 'G'.

Parameter Q is the base port of the relevant FDC card. (Normally E0 for a 2-drive system.) If two FDC cards were in use then this might be C0 for some drives. If FB then RAMdisc is assigned with a default format.

Parameter R specifies the least significant four bits of the drive select byte. Bit 0 if set indicates drive 0 on the relevant FDC card. Bit 1 indicates drive 1, etc.. Obviously, only one of the four bits should be set.

Parameter S specifies the type II flags. (See the Western Digital manual.) Normal setting is 1000 for Gemini FDCs (this sector length flag works for most formats) or 0010 for Nascom /Lucas FDCs (side compare enabled). It is very unlikely that you will need to change these settings. Side selection is handled automatically by the AllDisc software.

Parameter T specifies the type I flags. (See the Western Digital manual.) The bits you are most likely to wish to change are the last two (bits 0 & 1). These refer to four possible stepping rates. The smaller the value, the faster the stepping rate. Older drives may require this to be 11 (binary) whilst newer drives may work well with 00. The value set should be the smallest that will function correctly. Faster speeds improve disc access times and are quieter.

Parameter U is usually 'F'. It may be set to 'T' in order to read and write 48 tpi discs on 96 tpi drives.

Parameter V may be either 'T' or 'F'. If 'T' then any read/write errors will be reported on the screen together with the contents (in hex) of ports X0- X5 (where X0 is the base port of the FDC in use).

Parameter W specifies which CP/M drive identifier, A - P, if any, is associated with this drive. Typing a space will send it off-line.

3.3.3 A(rchive

enables access to the file FORMATS.DAT (which must lie on the current default drive) containing any standard formats which have been archived. Parameters B-N inclusive can then be saved or retrieved as a set with an identifier of up to eight characters. With this command it is possible to F(etch a format from the file and assign it to the currently selected drive, P(ut the format of the currently selected drive in the file, L(ist the identifiers of all formats in the file and D(elete any format from the file.

Note that P(utting will overwrite any existing format with the same identifier, otherwise it will add a new entry. Note also that the parameters O-W are peculiar to the drive rather than the format and may still need to be reconfigured after a F(etch. This will obviously be the case with parameter U when switching to or from a 48 tpi format on a 96 tpi drive.

3.3.4 R(e-tries

enables you to specify the maximum number of re-tries that will be attempted when reading or writing a sector.

Appendix. Some Formats and Hints

Only the formats listed in FORMATS.DAT have been tested and proved useable. Mostly these are various Nascom and Gemini formats. Quantum and Kenilworth computers use these same formats.

Here are some more formats that have been culled from various sources. Their reliability, however, is in doubt. To get them correct you will have to experiment, preferably with the aid of a utility such as DU to read the disc sector by sector and display the contents on the screen. Many of the parameters can be ascertained by judicious use of the "STAT DSK:" command on the original machine together with intelligent deduction.

							Parameter						
Computer	**8/5**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**
Altos	5	160	9	512	4?	1?	3	0	D	F	B	T	?	1
	8	77	48	128	2	2	2	0	D	F	0			1
	8	77	15	256	2	1	6	3	S	F	0			1
BT M3300	5	70	15	256	2	1	6	0	D	F	B	T	?	1
Cromemco	5	40	18	128	1	2	3	4	S	F	0			1
DEC Rainbow & Profess.	5	80	10	512	2	2	2	1	D	F	0			1
DEC VT180	5	40	9	512	1	2	2	1	D	F	0			1
Digico Prince	5	160	5	1024	2	4	1	0	D	F	B	T	?	1
IBM PC	5	40	8	512	1	2	1	0	D	F	0			1
	5	80	8	512	?	?	1	0	D	F	B	T	R	1
ICL PC 96tpi	5	80	20	512	2	2	1	0	D	F	B	S		1
Kaypro	5	40	10	512	1	2	1	0	D	F	0			0
McCombo	5	35	18	256	2	1	3	4	D	F	0			1
Memory	5	80	5	1024	2	1	1	1	D	F	B	T	?	1
NCR Decision Mate 5	5	80	8	512	2	2	3	0	D	F	B	T	N	1
Osborne	5	40	10	256	2	1	3	1	S	F	0			1
	5	40	5	1024	2	1	3	0	D	F	0			1
Philips P2000	5	80	32	256	4	1	1	1	D	F	B	S		1
Rair	5	70	30	128	1	2	3	1	D	F	B	T	?	1
RML	5	40	16	128	1	2	3	2	S	F	0			1
SuperBrain	5	35	10	512	2	1	2	1	D	T	0			1
	5	70	10	512	2	1	2	1	D	T	B	T	N	1
TMK 320P	5	80	16	256	2	1	4	4	D	F	0			1
Tuscan	5	40	10	512	2	1	2	1	D	T	0			1
Toshiba	5	70	16	256	1	2	3	3	D	F	B	T	?	1
Transtec	5	80	10	512	2	1	2	0	D	F	0			1
Televideo	5	80	18	256	2	1	2	0	D	F	B	T	?	1
Wang PC	5	70	9	512	2	1	2	0	D	F	B	T	?	1
Xerox 820	5	40	18	128	1	1	3	4	S	F	0			1
Zorba	5	80	10	512	2	1	1	0	D	F	B	T	?	1

It is usually obvious from these parameters whether a disc is formatted at 48 tpi or 96 tpi. If you wish to operate 48 tpi discs on a 96 tpi drive remember to set parameter U to True.

Discs with hardware-dependent formats, such as Apple and Sirius, cannot be read or written using AllDisc.

There are some additional formats which will give trouble. An example is the ICL PC format listed above. As it stands it is not quite correct. This is because the directory (and the block count) starts not at the beginning of track 1 but at the end (i.e. the first 8K -- 4 blocks -- are unused). This runs contrary to the normal conventions used by CP/M implementers. However, all is not lost.

All that is required is to do a mass sector copy of the whole ICL disk. To READ, copy from track 1, physical sector 17 (CP/M "sector" 65) to a new disc (or overwriting the the same disc if you have a back-up copy), starting at track 1 sector 1. (You can either write your own utility to perform this copy or use DU with the repeat facility to read into the buffer a block (group) at a time, starting at block 4, and write it out again, starting at block 0.) Then read the new disc according to the format above.

To WRITE, simply reverse this procedure -- but if you are rewriting to the same disc, start at the end!

If the directory is offset by a whole number of blocks from the begining of the track, then an alternative method is possible. This consists of moving only the directory to the beginning of the track rather than moving the whole disc contents. However, since the block numbers will then be out by the number of blocks the directory was offset, this offset must be added to the block numbers in each directory entry. (Again this can be done using DU.) To use this method, you will need some familiarity with the CP/M directory format (see DU.DOC).

DU-V86.DOC

By Ward Christensen (revised 10/18/83)
additional notes by Ron Fowler, Irv Hoff, and Jeffrey Nonken

```
; This version of DU is compatible with CP/M 1.4, 2.x and 3.x and does
; not require alteration for various hardware configurations.  It ad-
; justs itself automatically to the correct number of sectors, tracks,
; directory size, etc.
;
```

INDEX

1.0 INSTALLATION:

Because of the automatic adaption feature, no conditional assembly options are included. The only alteration that needs to be done is to use DDT to set the byte at 103H for your clock speed. Use 0 for 2MHz, 1 for 4MHz, 2 for 6MHz. (This only affects the 'Z' SLEEP command.)

2.0 USE:

An initial command string may optionally be placed as an operand of the original 'DU' command, i.e.:

```
A>DU G0;D;G2;=OK<D><A><1A>;D
```

for example, if you want to only map the disk, and then exit:

```
A>DUU M;X
```

Once 'DU' is running, it expects single-letter commands much like 'SID' OR 'DDT'. For ease of use, multiple commands may be placed on one line, separated by ";". In addition, a given command or string of commands may be repeated -- either indefinitely (until ^C is pressed) or a given number of times.

To avoid an accidental ^C from dropping out of 'DU', only an explicit "X" command will exit 'DU'.

2.1 COMMANDS, BY FUNCTION

HELP:

```
?       displays the help guide
```

POSITIONING:

```
Gnn     by allocation group
Snn     by sector
```

```
        Tnn     by track
        +nn     going ahead nn sectors
        -nn     going back nn sectors

I/O:
        <       puts current sector "away" into a buffer
        >       recalls previously saved sector
        K       writes "yanked" sectors to a file
                   (see "saving sequential memory" in notes below)
        R       reads sector
        W       writes sector
        Y       "yanks" current sector into sequential memory

DISPLAYING:

        £       shows disk parameters
        A       dump sector in ASCII
        D       dump the sector (hex + ASCII)
        G       shows current group, track, sector
        H       dump sector in hex
        M       maps the disk -- where the files are located
        Mxx     map starting at group xx
        Vnn     views (like CP/M type) nn sectors

CHANGING:

        CAnn,VAL change data in ASCII (with <xx> escape to hex)
        CHnn,VAL change data in hex
        Unn     change user to nn

SEARCHING:

        =Abc    scan for Abc (IN ASCII) from current sector on (very
                   slow, allow up to 15 minutes to scan an entire disk.
                   Either finds the answer or says: "out of bounds".
        FNAME   find a file in the directory
        F       find next occurrence (extent) of same name

MISC:
        (       toggles the map display to show/not show erased files
        /nn     repeat previous command nn times  (repeats indefinitely
                   if nn omitted)
        Bnn     boot nn sectors per track
        LX      log in disk X
        P       printer toggle
        Q       before a command does it 'quietly'
        X       exit to CP/M
        Znn     sleep (nn tenths of a second) to allow viewing data
                   before it scrolls off
```

2.2 ALPHABETIC COMMAND SUMMARY

```
        £       PRINTS THE DISK PARAMETERS
        +       Advance 1 sector (if below track 2, this advances to
                   next numerical, if 2 or more, advances based on
                   CP/M's normal sector scrambling algorithm, i.e.,
                   allows + to the next logical sector of the file.
        -       backs up 1 logical sector
                   NOTE:  + and - may take an amount:  For example,
                          +15 advances 15 sectors
        /       repeats entire command -- defaults to "forever"
        /nn     nn may be 2 TO 65535
        (       toggles the map display to show/not show erased files.
                   When showing erased files '*' indicates that block
                   duplicates a block in another file.  It may not be
```

```
                (starts at 3000H, increments for each yank)
        Z       sleep -- causes the porgram to pause -- such as to look
                at a dump.  Z is 1 second
        Znn     nn tenths of a second  Z50 = 5 seconds
```

3.0 NOTES

* Multiple commands: May be separated by ";"

EXAMPLE: The following commands will erase the B: disk directory to all E5's:

```
        LB              log in B: drive
        GO              position to directory
        CHO-7F,E5       fill with E5
        <               save the sector
        >;W;+;/16       restore, write, next, repeat 16
```

---- this could be shortened to:

```
        LB;GO;CHO-7F,E5;<
        >;W;+;/16
```

* DUMP COMMANDS: All dump commands (D, A, H) may be optionally followed by a starting and ending address:

```
        DO,7F           the same as just 'D'
        D3,5
        A2O,3F
```

* LOGGING IN DISK: If you have a disk with a 'blown directory', try logging in a good disk of the same density, then put in the 'blown' disk without logging it in. You are opening yourself to possible problems because of the buffering of physical sectors in the 'BIOS'. The best technique, (but not guaranteed), would be to seek to the unused inner tracks of the first disk, do the read, then change disks. That way, if it write anything, you won't have destroyed anything -- assuming the disk is not completely full. Another technique (assuming the second disk does not contain a CP/M system, would be to seek to track 1, do the read there, then change disks to the blown one.

* SAVING SEQUENTIAL MEMORY:
Since CP/M v 3.0 does not have a SAVE function, one has been added.
Syntax is:

```
        Kdu:filename.ext
        ^^^     ^    ^
        |||     |    +-- file extension (0-3 characters)
        |||     +------- file name (1-8 characters)
        ||+-------------- user £ (or none)
        |+--------------- drive designation (A-P or none)
        +---------------- DU command
```

Drive and user may be omitted. If so, omit the colon as well. Drive must be specified if the user is. If the user £ is omitted, the current user is used. If the drive is omitted, the current CP/M default drive is used.

This function saves the current contents of sequential memory into a disk

```
            possible to restore this program without errors.  If
            there are no '*' in this complete file, it can be
            correctly restored.
<           saves current sector in a 'save' buffer
=Abc        ASCII search, starting at current sector.  <xx> hex
            may be imbedded or used alone.  To find: "IN OFEH"
            use:  =<DB><FE>  (Ignores bit 7 unless using <xx>.)
            Since ";" is a command delimiter, you have to use
            <3B> to search for a ";".  Also, since "<" is a hex
            escape character, use << to mean a single "<".

               NOTE:  This is a very slow routine.  It can take 15
                      minutes or longer to search an entire double-
                      density double-sided disk so be patient.  It
                      either finds the string or says:  "OUT OF
                      BOUNDS".
>           gets saved buffer.  < and > maybe be used to move
              a sector to another place.
?           displays the help guide
A           dump sector (ASCII only)
Bnn         boot nn sectors per track -- not all disks have 26.
CHADDR,VAL,VAL,VAL...    change hex values in sector
CAADDR,CHAR STRING...    change ASCII calues in sector
               NOTE:  <xx> may be hex imbedded in the ASCII
                      string:  CA00,OK<OD><OA><1A>
               ---->  W writes changes to disk
                      note that the 'C' command echoes the over-
                      laid data for verification.
CHADDR-ADDR,BYTE         repeats a change
CAADDR-ADDR,BYTE         repeats a change
D           dump sector (hex + ASCII)
FNAME       print directory for file "NAME", then positions to
            its directory sector.
F           find next occurrence of name in directory
Gnn         position to group nn and read
G           shows current position
H           dump sector, hex only
Kdu:filename.ext    save a file from "yanked" sectors.
               drive, user are optional.  Resets "yank" address.
               see "saving sequential memory" in notes below.
L           re-logs in the current disk -- you may pull out a
            disk, put in a new one, and "L" just to log it in.
            (see "logging in disk" in notes below)
LX          logs in disk 'X', such as: LB, LC, LA, etc.
M           dumps a map of the group allocations for files
Mn          shows which file is allocated to group "N"
N           resets CP/M via BDOS -- this may make it possible under
            some implementations of CP/M to change the disk
            format, i.e., density, sides, etc.
P           toggles the printer on/off
Q           quiet -- preceedintg any command, suppresses CRT output
R           reads into memory the sector currently positioned at.
               NOTE:  'R' (read) is implicit in the G, +, and -
                      commands but NOT in the 'S' and 'T' commands
Snn         position to sector nn, and read
TNN         seek to track nn (no read)
Ux          logs user 'x' for next 'F' command
V           views the current sector -- assumes ASCII data
Vnn         views nn sectors
W           writes the current sector to disk
               NOTE:  may NOT be used after an 'F' command as
                      CP/M was used to find file in the directory
X           exit back to CP/M (must press return).  ^C was too
            easy to hit over modem lines -- requires two bytes:
            (X,CR) to exit.
Y           "yank" the current sector into sequential memory
```

file. The contents of sequential memory are determined by the 'yank' function, and the pointer of that function is used here. If nothing has been yanked, you get an error. Once the file has been saved, the 'yank' pointer is re-initialized to its original value (3000H). Control is returned to DU.

4.0 INTERPRETATING DIRECTORY DATA

4.1 SINGLE DENSITY

The following explains the format of a CP/M directory entry as shown by 'DU', using either the 'F' (find file) command, or just doing 'D' (dump) of the directory sectors, which are located in groups 0 and 1 on a single-density disk.

SAMPLE RESULT OF 'FSID.COM' COMMAND:

```
40  00534944 20202020  20434F4D 0000003A  *.SID     COM...:*
50  33343536 3738393A  00000000 00000000  *3456789:........*
```

FIRST LINE -

```
40  00534944 20202020  20434F4D 0000003A  *.SID     COM...:*
||  |||                          | ||    ||    |         |
||  ||^---hex file name/type--^ ||    ||    ^file name^
||  ||                          ||    ||     in ASCII
||  ||                    extent-^^    ||
||  ||                                 ||
||  ||           file size in sectors-^^
||  ||
||  ^^-00 = file active  other values (E.G 03) = user £
||     E5 = file erased
^^-displacement of line in directory sector
```

SECOND LINE -

```
50  33343536 3738393A  00000000 00000000  *3456789:........*
    |                                  |   |
    |                                  |   ^- allocation groups
    ^-----allocation group numbers-----^         (just happened to
                                                 be printable)
```

4.2 DOUBLE DENSITY

The following is a sample of 'FSID.COM' running on a double-density system:

```
:FSID.COM
00  00534944 20202020  20434F4D 0000003A  *.SID     COM...:*
10  38003900 3A003B00  00000000 00000000  *8.9.:.;.........*
G=0000:00, T=2, S=1, PS=0
```

The primary difference is that the groups now occupy 2 bytes, i.e., 38 00" "39 00" ... this follows the Intel and CP/M convention of putting 16-bit values high-byte-first. This it means group 0038, 0039 etc.

Note that in double-density, each group stood for 2k not 1K, so there were half as many groups for the same file.

Be very careful when patching a directory under double-density. For exmaple:

CH10,38,39,3A,3B...

This might try to access group 3938 with resultant angry noise from the disk stepper as it attempts to find where it should go for the data.

* * * * *

```
type kenpatch.asm
;NCPATCH1.ASM   20th Jan 84
;Configuration Overlay for Wordstar 3.3
;Using Inverse Video
;
;Sign-on Message Change
;
        org     18ah
        db      '     Kenilworth 83G          '
;
;
;Set Screen Height and Width ( 25 x 80 )
;
        org     232h
        db      19h,50h
;
;
;Cursor Positioning Codes
;
        org     234h
        db      02h,1bh,3dh
;
        org     247h
        db      00,20h,20h,00
;
;
;Erase to end of line
;
        org     250h
        db      02h,1bh,2ah
;
;
;Delete line
;
        org     257h
        db      01h,0bh
;
;
;Insert line
;
        org     25eh
        db      01h,0eh
;
;
;Inverse Video ON
;
        org     267h
        db      02h,1bh,41h
;
;
;Inverse Video OFF
;
        org     26eh
        db      02h,1bh,4eh
;
;
;Terminal Init. (Select Dim and Clock)
;
        org     275h
        db      08h,1bh,74h,45h,1bh,41h,1bh,61h,66h
```

```
;
;
;Terminal Un-init. (Deselect Clock Display & Cls)
;
        org     27eh
        db      04h,1bh,74h,44h,1ah
;
;Set Delay Times
;
        org     2afh
        db      03h 01  ;Cursor blink (Single Position)
        db      03h 01  ;Cursor blink (Dual Position)
        db      09h 0C  ;Between Prefix key and its Menu
        db      03h     ;After Sign-on, NEW-FILE, ABANDON
        db      09h     ;Horizontal Scroll Delay
;
;
;Patch for reading printer status. Used when
;Printing and Editing at the same time.
;
        org     2e0h
        db      0dbh,0b4h,1fh,0c9h
;
;
;Change Backspace key so that it also deletes
;
        org     4a3h
        db      43h,68h
;
;
;Make DEL work the same as Backspace
;
        org     545h
        db      7fh
;
;
;Patch to make the Cursor Keys work as well as
;the usual Diamond of CTRL E,S,D,X.
;
        org     655h
        db      1eh,00,0b4h,64h
        db      1fh,00,9ah,64h
        db      1ch,00,0dah,63h
        db      1dh,00,0d0h,63h
;
;
;Patch to make WS use the routine at 2E0h
;when editing and printing at the same time.
;
        org     718h
        db      0ffh,0c3h,0e0h,02h
;
end.

B>
```

PARAMETERS

A - DRIVE N° 1 TO 6
B - TRACK PER DISK
C - SECTORS PER TRACK
D - SECTOR SIZE
E
F
G
H
I - S=SINGLE (FM) D=DOUBLE(MMF)