

THE **ALTERNATE** SOURCE

and Allan J. Domuret present:

DDSD/CMD

**Automatic
Density
Recognition
for
NEWDOS/80
Version 2.0**



DDSD/CMD
Automatic Density Recognition
For: NEWDOS/80 Version 2
And TRS-80 Model I or III

INSTRUCTIONS FOR USE

Your DDSD/CMD diskette is formatted single density on 35 tracks using the Newdos-80 FORMAT command. Before doing anything with it, copy the DDSD/CMD file to your operating system and put the original away. Single drive owners will not be able to access the distributed formatted-only diskette; we apologize for this inconvenience. Affected persons will need to return the original DDSD diskette and a diskette containing NEWDOS/82, and we will perform the transfer of files for you. For multiple drive owners, instructions for copying are in your Newdos-80 manual, pages 2-09 through 2-16. Model I owners will use the files DDSD1/CMD and DSET1/CMD; Model III owners will need to convert DDSD3/CMD and DSET3/CMD. It is presumed that you will name the files DDSD/CMD and DSET/CMD on your system diskette.

Using DDSD/CMD is no more complicated than simply running it from DOS. Go ahead and run it, now!

Mix single and double density diskettes on an appropriate drive and try a few DIRectory reads. Note also that you can mix Double Density (DD) diskettes with track zero formatted in DD or Single Density (SD). (This feature relates to the PDRIVE setting, TI=CK or TI=C. See pages 2-37 of your NEWDOS/80 Version 2 Documentation.) Your NEWDOS-82 no longer requires numerous and time consuming PDRIVE changes.

DDSD/CMD requires no disk ZAPs of any kind, it does not conflict with fast CPU clock systems, and, if desired, DDSD/CMD can be established as an 'AUTO' file. DDSD/CMD will function on any 5 inch floppy drive when used with the TRS-80 Model I expansion interface.

DDSD/CMD will work in either 32K or 48K disk systems equipped with one or more drives. It will check for high RAM drivers (as might be used for certain printers, etc.) and, if any are present, it will load itself in memory just below them so as not to interfere. In the absence of high RAM drivers, DDSD/CMD will load itself to the top of available memory and it will protect itself by automatically setting the appropriate high memory value (that is, "Memory Size") in memory location 4049 hex (4411 hex for the Model III) as required by the TRS-80. It will NOT be necessary to respecify MEMORY SIZE when going into BASIC unless a need for additional protected memory exists.

DIRCHECK AND SUPERZAP PECULIARITIES

If NEWDOS-82 DIRCHECK or SUPERZAP is used, DDSD/CMD will recognize their presence and will compensate as required. However, some DIRCHECK or SUPERZAP misbehavior might be anticipated if diskettes with significantly different DD formats are used alternately in the same drive.

For example, if DIRCHECK is active and is given a DD diskette with track zero formatted in SD (PDRIVE setting, TI=CK), DDSD/CMD will work as advertised. If another DD diskette is subsequently given to DIRCHECK without recycling, but with track zero formatted this time in DD (PDRIVE setting, TI=C), DIRCHECK will complain of numerous diskette errors. Similar difficulties may be encountered with SUPERZAP. If you find yourself in this position, simply call up MINIDOS (with the 'DFG' keys), do a DIR for that particular drive, then upon 'MDRET' everything will work fine. This DIR read will establish the correct PDRIVE codes for the current diskette which will then enable DIRCHECK or SUPERZAP to function correctly.

Last minute additions to the DDSN documentation:

DDSD is distributed on a formatted (only) NEWDOS/80 Version 2.0 35 track single density diskette. This diskette was created with the NEWDOS/80 SYSTEM switch RN=Y. This means that the diskette can be read either on a Model I or Model III TRS-80. In order to read the disk on a Model III, you will need to PDRIVE the Model III drive containing DDSD as follows:

TI=A,TD=A,TC=40,SPT=10,TRR=3,GPL=2,DDSL=17,DDGA=2

This most likely coincides with the current setup of your DRIVE 5; if so, you can execute the following command from DOS READY:

PDRIVE 0 n=5 A<ENTER>

where "n" is a number (1, 2 or 3) corresponding to the drive number holding the master DDSD diskette. If you have a single drive, consult your NEWDOS/80 manual for information on SPDN and DPDN.

The files on the DDSD master disk are:

DDSD1/CMD	DSET1/CMD	DDSD1/ASM	DSET1/ASM
DDSD3/CMD	DSET3/CMD	DDSD3/ASM	DSET3/ASM

"1" and "3" indicate which system the program should be run on, either a Model I or a Model III.

In order for DDSD to function properly, you must have your NEWDOS/80 SYSTEM switch AN equal to 2 or more.

Your DDSD package includes the source code for both DDSD and DSET. Our thanks to Allan Domuret for allowing us to offer this to end users.

Always use DSET before performing a COPY or FORMAT to make sure the drive is configured as you last remember.

rules established by the competing DOS authors (NEWDOS, LDOS, DOSPLUS, etc.) to enable transportability of SD diskettes between the two TRS-80 Models. SD transportability is recommended because, in most cases, DD transportability is virtually impossible.

Since our purpose is to make DDSD/CMD function with NEWDOS-80, Version 2, it becomes necessary for the user to become familiar with the various ZAPs and references (explanations of incompatibilities induced by other DOS systems, plus those induced by the implementation of competing SD vs DD Data Address Marks (DAM)) made throughout the NEWDOS-80, Version 2 manual.

Briefly, old DAMs (Data Address Marks) used by TRSDOS, NEWDOS 2.1, and the older VTOS, in the directory sectors were an 'FA', but the DD disk controller chips cannot recognize an 'FA' DAM. Therefore, all DD DOS systems have resorted to using an 'F8' DAM for compatibility with DD disk controller chips. This DAM re-write is, among other things, part of what the various DOS conversion utilities accomplish in order to make the SD diskette readable on the Model III.

The DAM problem is also the reason why Model I DD conversion boards retain the services of two separate Disk Controller Chips: the old WD-1771 is used in SD operations and maintains compatibility with old SD diskettes which use the 'FA' DAM. The DD Controller Chip (usually a 1791 or 1793) can perform SD disk operations, but it refuses to recognize the former 'FA' DAM. Consequently, the competing DOS systems have adopted an 'F8' DAM (in the diskette Directory) for DD operations.

To further complicate matters, Radio Shack in their DD Model III TRSDOS has resorted to numbering disk sectors from one to eighteen (1 - 18) rather than the more traditional zero to seventeen (0 - 17) sector count favored by, and still used by, most other DOS systems. Among other things, Model III TRSDOS uses 'F8' and 'FB' DAMs in unusual ways on their DOS diskettes. (By the way, it should be mentioned that both the 'FB' DAM and the 'FE' IDAM (Identification Data Address Marks), are still legal in both SD and DD). Last but not least, Radio Shack has messed around with their directory design and file management structure to the extent that their own Model III software virtually assures them of isolation from the rest of the software world.

In short, the world of TRS-80 DOS systems gets more complicated rather than more standardized. As stated once before, the time has arrived for the TRS-80 user to decide on one DOS system and stick with it. Program swapping can best be accomplished in SD, and both DD and SD compatibility can be improved upon by staying with one of the better DOS systems such as NEWDOS, LDOS, DOSPLUS, etc.

Disclaimer: No intentional omissions of competing DOS system acknowledgement is intended here. The three systems mentioned by the author -- Allan J. Domuret -- are the only systems he has experience with and so he can not safely or accurately make any statements about the capabilities or functioning of competing DOS systems. The statements made are his own and do not necessarily represent those of any dealers who might market this program.

DSET/CMD

DSET/CMD was created in response to suggestions received by numerous individuals who were involved in evaluating the original DDSD/CMD Automatic Density Recognition program.

As is the case for DDSD/CMD, its intended use is with NEWDOS-80, Version 2 only. Additionally, DSET/CMD will not function without its parent DDSD/CMD being active in the system.

DSET/CMD has two simple functions:

1. To inform the user of the current PDRIVE density settings only. (If you need a complete PDRIVE readout, do the usual

These DIRCHECK or SUPERZAP problems may occur during a single performance of either of these utilities only when DD diskettes of substantially different format are mixed on the same drive. If your DD formats for a particular drive stay consistent, everything will work fine. Note that it is permissible to intermix DD and SD diskettes during the same performance of either DIRCHECK or SUPERZAP.

It is recommended that you experiment with some different diskette formats until you become familiar with the substantial convenience provided by DDSD/CMD.

SOME SUGGESTIONS

DDSD/CMD is pretty much automatic, but by understanding and adhering to a few simple rules, its usefulness and reliability will be substantially extended. Please observe the following:

1. REDEFINE ALL PDRIVE PARAMETERS TO DOUBLE DENSITY SETTINGS BEFORE USING DDSD/CMD.
2. WHEN DOING A DISKETTE FORMAT, NEWDOS-82 WILL USE THE PDRIVE VALUES AS CURRENTLY MAINTAINED IN MEMORY. IF YOU ORIGINALLY HAD A DOUBLE DENSITY PDRIVE CODE ESTABLISHED FOR, SAY, DRIVE ONE, BUT YOU SUBSEQUENTLY ACCESS A SINGLE DENSITY DISKETTE, DDSD/CMD WILL CONVERT THE PDRIVE PARAMETERS FOR DRIVE ONE FROM DOUBLE DENSITY TO SINGLE DENSITY. IF LEFT UNCHANGED, A SUBSEQUENT FORMAT OPERATION ON THAT DRIVE WILL BE PERFORMED IN SINGLE DENSITY. TO RE-ESTABLISH THE ORIGINAL DOUBLE DENSITY PDRIVE SETTING FOR THAT DRIVE BEFORE FORMATTING, SIMPLY INSERT AND READ A DIRECTORY ON A DISKETTE WITH THE DESIRED DD FORMAT, OR REBOOT.

SOME TECHNICAL DETAILS

The reason for suggesting that the PDRIVE settings be established in double density (See bold type paragraph number one, above) is that DDSD/CMD reads the PDRIVE settings currently in RAM and it uses this data to create two PDRIVE code tables, one with single density parameters and one with double density parameters. Most single density PDRIVE parameters are fairly standard, but the number of PDRIVE parameter combinations available for double density formats multiplies rapidly. In short, it is easier for DDSD/CMD to convert your double density PDRIVE settings into a standard single density PDRIVE code than it is to convert single density PDRIVE settings into double density code; there is no way that DDSD/CMD can determine what you intended for a double density PDRIVE setting if you only give it single density information to begin with.

DDSD/CMD has been tested extensively under a variety of conditions except that it was not tested on a TRS-80 equipped with eight inch or hard drives. If you experience any problems when operating in a normal floppy disk environment (five inch floppies), please contact the author or The Alternate Source by mail or phone and provide all possible details. You will be provided with free updates and or fixes, except for mailing and handling costs, whenever they are implemented.

In the event that Apparat introduces future NEWDOS-82 ZAPs which interfere with DDSD/CMD, we will make every attempt to re-establish the functioning of DDSD/CMD with our own counter-ZAPs. We will attempt to send our counter-ZAPs to legitimate (and registered) DDSD/CMD owners, but if you do not hear from us in a reasonable amount of time, please send a query and we will get back to you.

SUPPLEMENTAL DOCUMENTATION

In consideration of the incompatibilities of DD and SD diskettes created by various DOS systems, it is strongly suggested that the user should learn and observe the SD and DD disk swapping and conversion

NEWDOS PDRIVE command.

The command format is as follows:

DSET 3S,1D,0S,2D <Enter>

or, simply: DSET <Enter>

The command explanation and a few simple rules are provided. DSET is the filespec; the S or D commands a Single or Double density code setting in RAM (no disk reads or writes); and the numbers (0 - 3) represent the drive numbers. (Four drives maximum are currently allowed by NEWDOS-82).

Note that the S or D codes can be entered in any drive sequence; that is, entries do not have to be made sequentially from 0 to 3. After the mandatory blank space following the filespec, commas or blanks can be used as desired -- either will be accepted as parameter separators. Entries can be made for 1 to four drives (use the standard TRS-80 drive numbers of 0 to 3), or if no drive parameters are specified, entering DSET by itself will display the current density status of all active drives. (If your display gives a drive status reading for, say, four drives when your system is equipped only with two, check your SYSTEM AL=# setting as described on pages 2-47 of the NEWDOS-82 documentation).

A few more rules: the Model I and Model III versions are NOT interchangeable. Additionally, DSET/CMD will function ONLY if its corresponding DDSD/CMD is active within the system. Finally, if you are prone to getting into the machine code of either of these programs, there is a risk that you will, to put it simply, muck things up. DSET searches through DDSD/CMD using certain parameters as its clue as to the location of its PDRIVE SD and DD tables. Anything that upsets the position or ordering of these parameters may render everything useless. Experiment if you must, but be careful.

TECHNICAL COMMENTS

The DD PDRIVE codes maintained by both DDSD/CMD and DSET/CMD are identical to those originally defined via the PDRIVE command by the user. Whatever PDRIVE codes the programs find within the system will be the same codes stored for subsequent reference purposes. When diskettes of different format (SD vs DD, or a SD track zero on a DD formatted diskette) are alternated within a drive, the original system PDRIVE settings are always maintained intact for reference by the programs, but various combinations of PDRIVE parameters will be tried until something is found that works. In short, when toggling between SD and DD on a particular drive with DSET/CMD, you are simply re-establishing your original DD PDRIVE codes when going into DD mode.

Thank you for your interest in our products.

Allan J. Domuret

(NEWDOS-80 is a trademark of Apparat, Inc.)
(TRS-80 is a trademark of Tandy Corp.)

Comments, Criticisms, Suggestions, Problems, Etc.

can be addressed to the author of DDSD, Allan J. Domuret, 7825 Willowcrest Way, Fair Oaks, California, 95628 or to The Alternate Source, 704 North Pennsylvania Avenue, Lansing, Michigan, 48906. Phone numbers are: Allan, (916) 967-7331 and TAS, (517) 482-TAS8.

DDSD/CMD

allows you to forget about PDRIVE configurations when using NEWDOS/80 Version 2.0! Set each drive configuration for double density use, then AUTO the DDSD/CMD program. Now you can switch from single density to double density as often as you like, without ever having to change the PDRIVE table! DDSD/CMD will figure out the density of each disk you try to access, and will store that information in its own memory. If you need to change densities, don't worry -- DDSD will know. Included with DDSD is a program called DSET, which does allow you to specify a drive as single density or double density, should you so desire. For instance, if you need to format a disk, the format will be single or double, depending on the last density used. If you don't recall the last density used, just DSET 1S or DSET 2D; this tells DDSD that you want drive one to be single density, or drive two to be double density, respectively. Of course, more than one specification can be entered at a time. Another bonus: the DDSD package includes fully commented source code! This allows you to modify the program as you see fit, study the techniques used, etc. PDRIVE tables have always been a headache, so take two DDSD's, and call if you have any further problems. DDSD was written by Allan J. Domuret.

May, 1982

DDSD/CMD -- Distributed by: The Alternate Source
704 No. Pennsylvania
Lansing, MI 48906

Programmed by: Allan J. Domuret
7825 Willowcrest Way
Fair Oaks, CA 95628

Documentation update for DDSD/CMD and DSET/CMD, Versions 1.2 and 3.2 (TRS-80 Models I and III, respectively).

Version 1.2 and 3.2 Changes to DDSD/CMD

1. Since NEWDOS/80 V2 users continue to experience problems associated with FORMAT and/or COPY operations when DDSD/CMD is active, a change has been implemented in which DDSD/CMD automatically deactivates itself whenever NEWDOS/80 V2 gets ready to do a FORMAT operation of ANY kind. For example: DDSD deactivates with an ordinary diskette FORMAT operation, with a single-track or multiple-track FORMAT, or with a COPY-with-FORMAT operation. This fix should eliminate once and for all the FORMAT-associated problems previously experienced while using DDSD/CMD with NEWDOS/80 V2.

With this fix programmed into DDSD, NEWDOS/80 should FORMAT exactly as commanded. Remember, however, that NEWDOS/80, unless commanded otherwise, will FORMAT using default values from its PDRIVE table as maintained in RAM (Memory). And, it is possible that the PDRIVE parameters in RAM might have been modified by DDSD/CMD prior to the intended FORMAT operation. For example, with DDSD active, if you did a single density disk read or write in, say, drive 1, to be followed by a FORMAT operation in the same drive, NEWDOS/80 will FORMAT in single density unless the FORMAT command specifies otherwise. This happens because DDSD/CMD established SD PDRIVE parameters in the PDRIVE table for the SD disk access, and NEWDOS/80, when subsequently commanded to do a FORMAT operation, will default to these PDRIVE parameters.

When confronted with this condition, DSET/CMD can be used to either determine beforehand a disk drive's density status, or it can be used to re-establish the desired SD or DD PDRIVE parameters. For example, in the situation described above, DSET/CMD can be used to re-establish or re-affirm DD PDRIVE parameters in the DCT before starting the FORMAT operation. (See your documentation for proper use of DSET/CMD.)

Upon completion of the FORMAT operation, DDSD/CMD will automatically reactivate itself.

2. DDSD/CMD can be turned ON, OFF, or it can be purged from the system using three new codes implemented in the new (version 1.2 and 3.2) DSET/CMD file. See the DSET/CMD supplementary documentation below.

Version 1.2 and 3.2 Changes to DSET/CMD

1. DSET/CMD has been modified to be callable as a "CMD" operation from within BASIC. For example, from BASIC, enter

CMD "DSET"

or:

The usual DSET drive status information will be displayed without disturbing BASIC and its program contents.

Credits for this fix to John Stedeford, Twin Falls, Idaho.

2. Three new commands have been implemented in DSET/CMD

- a. DSET OFF : Turns DDSD/CMD off. DDSD/CMD remains intact in RAM, but it is still protected by its HIMEM setting and it is reusable by invoking the next DSET command.
- b. DSET ON : Turns DDSD/CMD back on.
- c. DSET DELETE : Deletes, or purges, DDSD/CMD from RAM. Two conditions have been allowed for:
 - (1) If DDSD/CMD was lowest in RAM (e.g., no other files were RAM-resident lower than DDSD/CMD), its RAM space will be made available automatically by a reset of the HIMEM parameter in (4049H) or (4411H) for Models I and III, respectively.
 - (2) If the HIMEM parameter is established such that DDSD/CMD is higher in RAM than other modules, or it is higher than user-established HIMEM or "Memory Size" settings, DDSD/CMD will be deactivated, but HIMEM parameters will not be disturbed, thus not exposing those modules or disturbing user-established HIMEM or "Memory Size" settings.

Under this condition, DSET ON will still turn DDSD/CMD back on because the HIMEM parameters remain intact.

IMPORTANT NOTE!

For users who continue to experience FORMAT or COPY-with-FORMAT problems with DDSD/CMD active, be certain you have ALL current Apparat ZAPs installed. At the time of this writing (May, 1982), I have ZAPs for the Model I up to 057, and for the Model III up to 052. There are MANY ZAPs which directly affect NEWDOS/80 FORMAT and COPY commands. In particular, ZAPs 057 for Model I and 052 for Model III are directly related to the "Track Count Too High" problem some of you have experienced while using DDSD/CMD. Contact Apparat if you are not current with their ZAPs.

In reference to another minor discrepancy which some users have attributed to DDSD/CMD, be aware that when a NEWDOS/80 'DIR' command is given, NEWDOS/80 takes the track count from its RAM PDRIVE table (or what I have referred to as the DCT, or Drive Code Table), and NOT from the disk directory being read. If a 34 or 35 track diskette directory is read in a 40 track drive which is PDRIVE'd for 39 or 40 tracks, the 'DIR' command will report the PDRIVE'd 39 or "40 tracks" rather than the actual 34 or 35 tracks from the diskette being read. This is explained in the NEWDOS/80 documentation on p. 2-20, quote: "The values for track count and free granules are based on the current active PDRIVE specification for that drive and if those specifications are not proper, these displayed values

may be in error."

Conclusion: the inconsistent track count display during a DIR command is NOT a DDSD-related discrepancy..

Optional PDRIVE ZAPs:

The following ZAPs to SYS16/SYS will accept the usual PDRIVE entries as described on page 2-33 of the NEWDOS/80 V2 documentation, but with an additional feature that a 'B' suffix, in lieu of the 'A' suffix described on page 2-37, can be used to temporarily put new PDRIVE settings in RAM without writing them to disk. For example:

PDRIVE 0,1,TI=CK,TC=39,A

will put the PDRIVE settings in RAM and will write them to disk. But with

PDRIVE 0,1,TI=CK,TC=39,B

the new PDRIVE settings will go only to the NEWDOS/80 PDRIVE table in RAM; they will NOT be written to disk. Also, the new PDRIVE settings will NOT be made available to DDSD/CMD.

ZAPs for Model I:

SYS16/SYS, 02/F8:

from: 7E FE 41 12 20 06 23
to: 7E C3 C0 51 00 00 23

04/D4:

from: all zeros
to: FE 41 12 20 03 C3 F1 4F FE 42
C2 F7 4F 3E 41 12 77 C5 E5 01
00 07 21 E2 4D 71 23 10 FC E1
C1 C3 F1 4F (end)

For Model III:

SYS16/SYS, 02/E8:

from: 7E FE 41 12 20 06 23
to: 7E C3 C6 51 00 00 23

SYS16/SYS, 04/DA: all zeros

to: FE 41 12 20 03 C3 E2 4F FE 42
C2 E8 4F 3E 41 12 77 C5 E5 01
00 07 21 D3 4D 71 23 10 FC E1
C1 C3 E2 4F 02 (end)

DSET/CMD, For Use Only With DDSD/CMD

Automatic Density Recognition

for: Newdos-80 Version 2

And TRS-80 Models I or III

DSET/CMD was created in response to suggestions received by numerous individuals who were involved in evaluating the original DDSD/CMD Automatic Density Recognition program.

As is the case for DDSD/CMD, its intended use is with NEWDOS-80, Version 2 only. Additionally, DSET/CMD will not function without its parent DDSD/CMD being active in the system. Again, separate versions are required for the Model I and Model III.

DSET/CMD has two simple functions:

1. To inform the user of the current PDRIVE density settings only. (If you need a complete PDRIVE readout, do the usual NEWDOS PDRIVE Command.)
2. To allow toggling between DD and SD in RAM without doing a NEWDOS PDRIVE command.

The command format is as follows:

DSET 3S,1D,0S,2D (Enter)

or, simply: DSET (Enter)

The command explanation and a few simple rules are provided. DSET is the filespec; the S or D commands a Single or Double density code setting in memory (DSET/CMD does not perform disk reads or writes); and the numbers (0 - 3) represent the drive numbers. (Four drives maximum are currently allowed by NEWDOS-82).

Note that the S or D codes can be entered in any drive sequence; that is, entries do not have to be made sequentially from 0 to 3. After the mandatory blank space following the filespec, commas or blanks can be used as desired -- either will be accepted as parameter separators. Entries can be made for 1 to four drives (use the standard TRS-80 drive numbers of 0 to 3), or if no drive parameters are specified, entering DSET by itself will display the current density status of all active drives. (If your display gives a drive status reading for, say, four drives when your system is equipped only with two, check your SYSTEM AL=# setting as described on P. 2-47 of the NEWDOS-82 documentation).

A few more rules: the Model I and Model III versions are NOT interchangeable. Additionally, DSET/CMD will function ONLY if its corresponding DDSD/CMD is active within the system. Finally, if you are prone to getting into the machine code of either of these programs, there is a risk that you will, to put it simply, muck things up. DSET searches through DDSD/CMD using certain parameters as its clue as to the location of its PDRIVE SD and DD tables. Anything that upsets the position or ordering of these parameters may render everything useless. Experiment if you must, but be careful.

Technical Comments

The DD PDRIVE codes maintained by both DDSD/CMD and DSET/CMD are identical to those originally defined via the PDRIVE command by the user. Whatever PDRIVE codes the programs find within the system will be the same codes stored for subsequent reference purposes. When diskettes of different format (SD vs DD, or a SD track zero on a DD formatted diskette) are alternated within a drive, the original system PDRIVE settings are always maintained intact for reference by the programs, but various combinations of PDRIVE parameters will be tried until something is found that works. In short, when toggling between SD and DD on a particular drive with DSET/CMD, you are simply re-establishing your original DD PDRIVE codes when going into DD mode.

The Alternate Source
704 No. Pennsylvania Ave.
Lansing, MI 48906

**** ZAP 01/DDSD/CMD **** 06/24/82 **** Models I & III ****

DDSD/CMD recognizes a FORMAT-busy condition by checking two NEWDOS/80 bytes. The first check is for a 58H in (4369H); the second is for bits 1 and 2 set in (5399H). With both of these bytes set as such, DDSD turns itself off to avoid introducing spurious parameters into the PDRIVE table during a FORMAT operation. However, it turns out that some users are getting a 48H rather than 58H in (4369H) during a FORMAT. (Evidently, this is a variation introduced by certain SYSTEM parameter specifications.) Thus, the 'CP 58H' may in some cases cause DDSD to not recognize a FORMAT-busy condition. If your FORMAT operations occasionally generate spurious results, ZAP 001 should take care of the problem. Apply this ZAP ONLY if you are experiencing FORMAT problems.

**** For the Model I ****

DDSD12/CMD) 02/A8: change: 43 FE 58 C0
Model I) to: 43 FE 48 C0

**** For the Model III ****

The corresponding FORMAT-busy byte for the Model III is (4289H).

DDSD32/CMD) 02/A8: change: 43 FE 58 C0
Model III) to: 43 FE 48 C0

//

In the event the original ZAP bytes are not found at the track/byte locations shown above, use the SUPERZAP search feature to locate the bugged byte. From the SUPERZAP menu, perform the following command sequence:

1. User -- DFS
2. SZAP -- FILESPEC ?
3. User -- DDSD12/CMD:0 (Enter the appropriate filespec)
4. SZAP -- RELATIVE-SECTOR-WITHIN-FILE?
5. User -- 2

If the bugged-bytes are not found where expected, enter the following:

User -- F,FE,58 [Enter]

SUPERZAP will locate the specified bytes and it will identify them with a flashing cursor. Write down the relative byte offset indicated by the cursor, then perform the required ZAPs.

