by Kim Watt of

Breeze/QSD inc.
11500 Stemmons Expwy., Suite 125
Dallas, Texas 75229

Disk Program

Distributed by:

*TRS-80 is a product of Radio Shack, division of Tandy Corporation

Over the years, I have written a number of programs and it seemed that I constantly had to use various utility programs that were on the market from a number of different sources. Each of these programs performed a very specific function and I soon had invested several hundreds of dollars for these utility programs and seemed to be constantly having to buy more of these types of programs as the MICRO-MARKET expanded into ever higher levels of programming. As the programs on the market became more complex (and better), I was able to improve several of my existing utilities, but none of them were able to be the OVERALL utility that I was looking for.

Finally, after discussing this problem on numerous occasions with different people who were involved with TRS-80's, I decided to write a program in machine language that would be the LAST WORD in utilities and the resulting program was called SUPER-UTILITY. It was a 24K program that was introduced on the market in 1980, and was very well received all over the world.

During the following year, I received a lot of mail regarding that program and also requests for me to implement various other additional subroutines that people thought might make it even more versatile. I made several upgrades to the program trying to be receptive to their wishes, but then came DOUBLE DENSITY. After getting many requests to make it work with double density, I upgraded it again, but only so the ZAP option would work as I was running out of room.

We decided that an entire new program would have to be written. We made lists of all of the requests for additional options that had been received and we then proceeded to spend the next several months making the new program. The final result of our effort is called SUPER-UTILITY PLUS, and we hope that it will be everything that you ever wanted in a utility program.

I would like to take this time and thank JIM FRANK, the author of the manual, STEVE KING, SKIP TAYLOR, DENNIS BRENT, RALPH BURRIS, TOM BROWN, ROY SØLTØFF, JOHN LONG, PAUL WIENER, and JOE SIMON for their assistance as TEST SITES for this program, for without their donations of untold hours, this release would have been many months delayed. It is through people like these that Super-Utility has evolved into the highly sophisticated program that it is today.

==========
DOCUMENTATION FOR TRS-80
S U P E R - U T I L I T Y P L U S
==========

1.0 INTRODUCTION:

The following documentation is written in layman terms in an effort to assist novice as well as professional programmers in completely understanding the full potential of this extremely powerful utility program. However, this documentation is not written with the intention of teaching you how disk storage systems are organized and stored. If in depth information on this subject is desired, there are several excellent books on the market to assist you. INSIDE SUPER UTILITY, by Kim Watt, is aimed at the novice and expert alike, and is packed full of useful information, including detailed information on the differences in DOS structures.

Breeze/QSD offers this program for sale "AS IS" and assumes no liability whatsoever in the event of damages of any kind as a result of the use of this program.

1.1 REGISTRATION:

Upon receiving your S U P E R - U T I L I T Y P L U S program, please fill out the attached registration form and mail it to Breeze/QSD, Inc., 11500 Stemmons Fwy, Suite 125, Dallas, Texas 75229. Breeze/QSD will notify all registered owners regarding updates, enhancements, fixes, and new releases of coming programs. If you would like a backup copy of SUPER-UTILITY PLUS include $5.00 ($10.00 U.S. for foreign orders) when sending in your registration form. Be sure to copy your serial number from the diskette label onto the registration form.

1.2 MINIMUM HARDWARE REQUIREMENTS:

This utility requires a minimum of 48K RAM and one disk drive to operate. The SPECIAL TAPE COPY UTILITY requires two tape recorders. The efficiency of the program is considerably enhanced with two drives.

1.3 WHAT IS "S U P E R - U T I L I T Y P L U S:"

SUPER-UTILITY PLUS was written by KIM WATT of Breeze/QSD and is the most powerful program of its kind on the market at this time. This program is a machine language, stand alone program that has its own I/O routines, does not use any ROM or DOS calls, and works on SINGLE or DOUBLE DENSITY systems, on both the Mod I and III. It is the result of many months of hard work by Kim Watt and performs such a wide range of varied tasks, that it may truely be called "The King of Utilities". SU+ does not require that the disk be in any drive after initialization of the program and user may custom configure the program to suit his individual system requirements. SU+ is a single RAM resident program that performs ALL the following functions.

The ZAP utility is a program that does everything that similar products do, but it will do it on SINGLE or DOUBLE DENSITY systems and will work on most major operating system that are presently on the market. It has DUAL CURSORS (one for ASCII and one for the HEX side of the readout) and allows the user to go to the heart of the disk and read and/or modify data in HEX, ASCII, DECIMAL, BINARY, or OCTAL regardless of whether it is a STANDARD disk or not. The screen printout on ZAP is similar to Superzap

(one sector at a time in HEX and ASCII), but also tells user the true and relative track and whether the disk is IBM format or not. ZAP also has a search routine that will locate the highest or lowest configured track on the disk and others that will search the disk for a BYTE LIST, ASCII STRING, WORD LIST, or even ENCRYPTED CODE. ZAP also allows you to DISPLAY DISK SECTORS, VERIFY DISK SECTORS, COMPARE DISK SECTORS, COPY SECTOR DATA, ZERO DISK SECTORS, COPY DISK SECTORS, EXCHANGE SECTORS, REVERSE SECTOR DATA, SECTOR SEARCHES, READ ID ADDRESS MARKS, or ALTER DATA ADDRESS MARKS.

PURGE has a full screen editing kill control that allows you to kill files by positioning cursor and pressing one key. Also, PURGE has several sub-utilities that allow you to zero out unused directory entries or zero out unused disk granules. In addition, user may kill files by naming the common category of the files (EXAMPLE: /CMD /BAS /TXT <I>nvisible, <V>isible, etc. or even kill files that begin with a specified sequence of letters), and also may compute existing passwords, change the disk name, date, passwords, auto command, or even file parameters (name, passwords, protection levels). Lastly, PURGE contains a complete disk directory that indicates all active and non-active files on the disk.

FORMAT is a utility that allows the user to format a disk with STANDARD FORMAT, FORMAT WITHOUT ERASING EXISTING DATA, SPECIAL FORMAT, (custom format your disk most any way you want it) BUILD A FORMAT TRACK and optionally write it back to any track on your disk, and even contains a software BULK ERASE utility. The total formatting capabilities of this program are just about UNLIMITED and you may even REFORMAT OVER a disk or ADD TRACKS to an existing disk without destroying existing disk data.

The DISK COPY UTILITY will copy any standard disk from 1 to 96 tracks, with or without formatting. The SPECIAL DISK COPY enables the user to make a backup of MOST TRS-80 readable disks that are presently on the market, regardless of any efforts that have been made to protect them from being "BACKED UP". NOTE: (This program will NOT copy itself). This programs' only intended use is for you to make BACKUPS of your legally purchased programs. Please DO NOT USE this utility to make "BOOTLEG COPIES" for others as authors of quality programs deserve their royalities.

The TAPE COPY UTILITY enables the user to perform a wide variety of actions that include the ability to READ, WRITE, or VERIFY tapes and even includes a BIT BY BIT copying routine that will backup most ANY TRS-80 readable tape regardless of protection attempts made by authors. This utility also is for your own use only.

The DISK REPAIR UTILITY allows you to automatically repair the HIT and GAT sectors, and will automatically repair a BOOT. This utility also does a complete DIRECTORY CHECK and will advise you of errors that exist. In addition, this utility allows the user to RECOVER KILLED FILES, (if the file was killed by this utility or by most anything except TRSDOS), READ PROTECT or UN-READ PROTECT the directory, MOVE IT to a different location on the disk, or CLEAR UNUSED ENTRIES. Lastly, this utility advises you of all inactive files that are on the disk.

The MEMORY UTILITY supplies the ability to DISPLAY, MOVE, TEST, COMPARE, ZERO, EXCHANGE, INPUT or OUTPUT a byte to any port, EXCHANGE, JUMP TO, REVERSE, FILL, STRING SEARCH, or even LOAD/WRITE AN ENTIRE TRACK or SECTORS to/from memory.

The FILE UTILITIES consist of the abilities to DISPLAY FILES SECTORS, COMPARE FILES, COPY FILES, DISK DIRECTORY, FREE SPACE, FILE LOCATIONS,

DRIVE STATUS, CREATE FILES, CLEAR FILES from disk, and ENCODE/DECODE PASSWORDS or HASH CODES. These utilities give you a wide range of powerful commands at your disposal to perform just about any function that you want with files up and including the complete REORGANIZATION of your ENTIRE DISK with all the files re-written in their MOST CONTIGUOUS order.

CONFIGURE SYSTEM gives you the ability to CUSTOM CONFIGURE SU+ to your system. You may select SINGLE or DOUBLE DENSITY, in any combination, of 5" drives, select your operating system boot of your choice, upper or lower case, high speed clock, or even configure your printer.

1.4 LOADING AND INITIALIZATION INSTRUCTIONS:

To initialize the program, insert disk into drive Ø and press the RESET BUTTON. There will be a short delay while the program and configuration data is loaded, and the main menu will then be displayed to the screen (* Mod I users see footnote 21). If you hold down the <CLEAR> key while the program loads, the configuration data will NOT be read from the diskette. If you hold down either the right or left arrow keys during initialization, a list of system labels and entry points will be displayed to the screen. If the right AND left arrows are held down together during initialization, a copy of the labels will also go to the printer. These entry points are useful for utilizing the 1K patch area provided for user routines. These entry points may be moved on updated versions of SU+. A TECHNICAL REFERENCE MANUAL, by Kim Watt, that explains the usage of all the system entry points and calls is available from Breeze/QSD.

1.5 BACKUP:

As previously stated in the registration information, Breeze/QSD will send every registered owner of SUPER-UTILITY PLUS, ONE backup copy, upon request, for $5.ØØ ($1Ø.ØØ U.S. for foreign orders). If your ORIGINAL DISK should become PHYSICALLY DAMAGED, where it won't allow us to re-copy SUPER-UTILITY PLUS on it, send THE ORIGINAL DISK back to Breeze/QSD, along with $5.ØØ (to cover a new disk, postage and handling) and a new copy will be sent to you. (NOTE: YOU MUST RETURN THE ORIGINAL DISK!).

1.6 COMPATABILITY:

This program will operate with standard MODEL I and III TRS-80 MACHINES (specific versions for each machine), equipped with 5" TRS-80 compatible drives (up to 96 track), and will support TRS232 serial or parallel lineprinters for screen printing.

1.7 GENERAL PROGRAM INFORMATION:

****

This program has a default setting of 35 track SINGLE DENSITY. In order to perform operations on a disk that exceed these parameters, it is necessary to RESET the SYSTEM CONFIGURATION. This can be done in various different ways. The program disk itself may be "ZAPPED" so that it will automatically be set to your particular system or you may use TEMPORARY OVERRIDE COMMANDS that are discussed further in both this section and in the CONFIGURE SYSTEM SECTION of this manual. Either of these methods will work equally as well but we recommend that you "zap" the disk to meet the individual requirements of your system and use the OVERRIDE OPTIONS for special situations. Be sure to read the CONFIGURE SYSTEM section (2.9) and fully understand it prior to proceeding.

****

THROUGHOUT THIS PROGRAM, ALL OF THE FOLLOWING NOTATIONS WILL APPLY:

Any time numerical input is required from the user, you may enter in either HEX, DECIMAL, BINARY, or OCTAL. All numbers are assumed to be decimal unless you append the letter "H", "D", "B", or "O" to the end of the number, in which case it will be interpreted as HEX, DECIMAL, BINARY, or OCTAL. The default base may be changed, and is discussed in section 2.9.

Any time you are requested to enter several numbers on the same line, you must separate the numbers with one or more commas or spaces. All numbers displayed on the screen will be in DECIMAL unless the letter "H" is appended to the end of the number, in which case the number is in HEX. The only exception to this rule is that the displays in Zap are ALL in HEX unless otherwise noted.

Anytime that keyboard input is required, you may press <ENTER> and the default value (that is equal to the first possible selection) will be inserted.

This program recognizes upper and lower case and comes set so that pressing <SHIFT> and a letter key will result in a CAPITAL letter being displayed just like a typewriter. It may also be reversed by pressing <SH><Ø> at which time unshifted letters will be displayed as CAPITALS and small letters maybe entered by pressing the <SHIFT> key. To change back again, merely press <SH><Ø> again.

Throughout the program the <BREAK> key may be used to interrupt ANY routine in progress except cassette and printer I/O, in which <CLEAR> will interrupt. This includes all Disk I/O operations. The <BREAK> key will immediately return you to the last menu that was displayed and <SH><BREAK> will return you to the MAIN MENU. When the key is pressed, the word BREAK will appear at the top right of the screen to acknowledge the key. The menus will be displayed as soon as the <BREAK> key is released. In some routines, such as the screen printer or dual mode a 1K buffer is used and it may be emptied by pressing <CLEAR> to interrupt the routine and return you to the immediate mode that you left.

If any Disk I/O errors are detected, you will be notified and given the opportunity to either <R>etry, <S>kip this sector, <Q>uit, <C>ontinous, and <N>on-stop. This applies to both Read AND Write operations. Setting Continuous will simulate pressing the Retry key each time a sector error

occurs. It will return back to prompted mode as soon as the current sector is sucessfully read or is skipped. Selecting Nonstop will simulate Continuous, but will not revert back to prompted when the current sector is read or skipped. Holding down the <CLEAR> key will revert back to a prompt in the case of Continuous or Nonstop being selected.

Any time that information is being rapidly displayed, the screen may be paused at any time by pressing the <SPACEBAR>. Display will resume by pressing the <ENTER> key. Optionally, the <SHIFT> <SPACEBAR> may be pressing to pause, and display will resume as soon as it is released.

The screen printer may be activated anytime the keyboard is being scanned by pressing <SHIFT><CLEAR>. A 1k buffer is used for printer spooling. You may empty the buffer at any time by pressing the <CLEAR> key. Due to similarity between the two commands, when you are screen printing, be sure to release the <CLEAR> key BEFORE the <SHIFT> key, or some of the characters may be lost.

Whenever there may be differences in the manual regarding operation with the different DOS's, a * followed by a footnote number will be supplied. Unless the DOS you are using is mentioned, you will not need the information provided. If you are using the DOS listed, refer to the proper footnote listed at the rear of this manual.

The term PAGING MODE shall refer to anytime you are viewing a full page of data and are NOT in the MODIFY MODE. The term MODIFY MODE shall refer to when you are viewing a full sector of data AND have pressed the "M" key so as to be able to MODIFY data.

In the MEMORY UTILITY section of SU+, all changes are IMMEDIATE and are NOT held in a buffer area. Changes made in DISK MODIFICATION OPTIONS are held in a buffer area and are NOT WRITTEN TO THE DISK until the <U>pdate option is selected.

With so many different operating systems presently available on the market and many of them designed differently, it has become a problem to give SU+ the ability to interact with all of these different systems. This problem has been solved by allowing you to temporarily RECONFIGURE SU+ at any time the program requires drive input. This may be done by selection of any of the following options.

1.8 OVERRIDE COMMAND LIST:

1. (A or a) following a drive number, tells SU+ that the disk in the selected drive is SINGLE DENSITY and contains an operating system other than LDOS (Mod I TRSDOS compatable).
2. (B or b) indicates a Mod III TRSDOS format diskette.
3. (C or c) indicates a Mod I or III SINGLE DENSITY LDOS DISK.
4. (D or d) indicates a Mod I or III DOUBLE DENSITY LDOS DISK.
5. (E or e) indicates a Mod I or III SINGLE DENSITY DOSPLUS DISK.
6. (F or f) indicates a Mod I DOUBLE DENSITY DOSPLUS DISK.
7. (G or g) inducates a Mod III DOUBLE DENSITY DOSPLUS DISK.
8. (H or h) indicates a Mod I DOUBLE DENSITY DOUBLEDOS DISK.
9. (I or i) indicates a Mod I or III NEWDOS 8Ø version 2 DOUBLE DENSITY with SINGLE DENSITY TRACK Ø DISK.
10. (J or j) indicates a Mod I or III NEWDOS 8Ø version 2 DOUBLE DENSITY with DOUBLE DENSITY TRACK Ø DISK.

In ALL routines throughout the program that require SOURCE and DESTINATION information, if a single drive is used, the DOS SPECIFIER MAY NOT BE DIFFERENT. If, for example, you wish to copy files from a single density disk to a double density disk, 2 drives are required.

The following will contain examples showing some of the possible uses of the OVERRIDE COMMANDS.

EXAMPLE #1 - : You wish to examine track #5, sector #2, of a 40 track DOUBLEDOS disk in drive #1 for a friend and SU+ is configured for your 35 track SINGLE DENSITY NEWDOS+ system. From the MASTER MENU, type "1" and press <ENTER>. You will now be asked to select from the ZAP MENU. Press <ENTER> again and SU+ will DEFAULT TO THE FIRST POSSIBLE ENTRY when <ENTER> is pressed alone. You will now be in DISPLAY SECTORS and will be asked to input the DRIVE, TRACK, and SECTOR to be displayed. Type "1h=40 5 2" and press <ENTER> at which time said sector will be displayed to the screen. Another option that is available to you when you are asked to input the DRIVE, TRACK, and SECTOR information is you may input the drive number, followed by a space, and then followed by an:

1 - <UP ARROW> to display the HIGHEST CONFIGURED TRACK on the disk.
2 - <DOWN ARROW> to display the LOWEST TRACK on the disk.
3 - <L> to display the SAME SECTOR as was just displayed, but will allow you to select a different drive number. (* see footnote 1)
4 - <D> to display the configured directory track.

EXAMPLE #2 - : Super-Utility Plus is configured to your 40 track NEWDOS DOUBLE DENSITY system and you want to examine track 2, sector 4, of a 40 track LDOS DOUBLE DENSITY SYSTEM DISK in drive #2. From the MASTER MENU, type "1" and press <ENTER> or press <ENTER> by itself to default to the ZAP MENU. Next, type " 2d 2 4 ", press <ENTER>, and the sector will be displayed.

NOTE: The use of these OVERRIDE COMMANDS may seem a little technical at first, but once you become familiar with them, you will find that they give you the power to read/modify almost any disk on the market without having to spend the time or go to the trouble of reconfiguring your system. These options WILL NOT allow you to read/modify data in double density unless you have the double density modification in your computer.

Whenever PURGE is used to KILL FILES, it does so by resetting bit 4 in the PDE and XDE's. The HASH TABLE BYTES are then reset to Ø and all allocated space for the file is then released. Admittedly, this system makes it a little difficult to tell the active from the inactive files when ZAPPING, but if you want to restore a file using the DISK REPAIR utility, it will be restored to it's EXACT previous condition.

2.Ø SUPER-UTILITY PLUS OPERATIONS MANUAL:

Upon initialization of SU+, the MAIN MENU containing the following ten options will be displayed.

1. Disk Zap
2. Disk Purge
3. Disk Format
4. Disk Backup
5. Disk Repair
6. Tape Utilities
7. Memory Utilities
8. File Utilities
9. Configure System
Ø. Exit Program

Selection ?

2.1 DISK ZAP:

Upon selection of DISK ZAP, a menu consisting of the following twelve possible options will be shown on the screen.

1. Display Sectors
2. Verify Sectors
3. Compare Sectors
4. Copy Sectors
5. Copy Sector Data
6. Zero Sectors
7. Reverse Sector Data
8. Exchange Sectors
9. String Search
Ø. Sector Search
A. Read ID Address Marks
B. Alter Data Address Marks

Selection ?

2.1A DISPLAY SECTOR:

Upon selection of option #1, press <ENTER> at which time you will be prompted to input the DRIVE, TRACK, and SECTOR number that you wish to examine (* see footnote 2) (NOTE: user may input selections in HEX, DECIMAL, OCTAL, or BINARY). If selection is to be in hex, follow the numbers with a "H". If selection is to be in decimal, merely type in the numbers. (NOTE: Throughout this program, data inputs must be separated with either COMMAS or SPACES). At this time, the selected sector will be displayed on the screen (see below for example printout and the following page for explanation).

```
      00 00FE 11CD C901 FD21 0000 3A02 4257 1EE04 .......!..:.BW..
HEX   10 0651 CDAB 4220 580A E650 FE50 2056 2A16 .Q..B X..P.P V*.
DRV   20 5155 7C07 0707 E607 6707 0784 5F0E FFD9 QU\.....G...._...
 1    30 CD7E 42FE 0128 1AFE 0228 0FFE 2030 35CD .^B..(...(.. 05.
TRK   40 7E42 47CD 7E42 10FB 18E6 CD7E 42CD 6642 ^BG.^B.....^B.FB
 00   50 E9CD 7E42 47CD 6642 0505 CD7E 4277 BE20 ..^BG.FB...^BW.
TRU   60 1823 10F6 18CA CD7E 426F CD7E 4267 C921 .#.....^BO.^BG.!
 00   70 9942 186E 219F 4218 6921 F842 1864 D90C .B.N!.B.I!.B.D..
SEC   80 200D CDAB 4220 E81C 7BD6 0A20 025F 140A  ...B ..{.. ._..
 00   90 D9C9 2045 7272 6F72 0017 4469 736B 0017 .. ERROR..DISK..
STD   A0 5359 5300 C506 0A10 FEC1 C9CD AF42 C83E SYS..........B.>
ISD   B0 0132 E137 ED53 EE37 21EC 3736 18CD A442 .2.7.S.7!.76...B
      C0 CB46 20FC 3688 CDA4 42C5 1803 0F30 0C7E .F .6...B....0.^
      D0 CB4F 28F8 3AEF 3702 0318 F47E 36D0 C1E6 .O(.:.7....^6...
      E0 9FC9 CDEF 4221 9242 CDEF 42CD 4000 767E ....B!.B..B.@.V^
+00   F0 E7C8 CD33 0023 18F7 174D 454D 004B 696D ...3.#...MEM.KIM
```


FIELD #2: This contains the RELATIVE BYTE NUMBERS in the sector. (NOTE: A relative byte is it's offset from the beginning of the sector.)

FIELD #1: To the left of relative byte 1ØH (upper left hand corner of the screen) you will see the word "HEX", "ASC", "BIN", "OCT", or "DEC". These words indicate whether the program is presently capable of modifying in HEX, ASCII, BINARY, OCTAL, or DECIMAL (NOTE: See MODIFY MODE instructions). The user may change modes by pressing the letters "H", "A", "B", "O", or "D" at any time PRIOR TO ENTERING THE MODIFY MODE. (To change modes when in the modify mode, press <SHIFT><ENTER> followed by "H", "A", "B", OR "O","D"). To the left of relative byte 2ØH the word "DRV" appears, and it stands for drive. To the left of relative byte 3ØH, a number between Ø and 3 appears and it indicates the drive # that is being read. To the left of relative byte 4ØH, the word "TRK" will appear and it stands for track. To the left of relative byte 5ØH a number appears that tells the relative track # (in DECIMAL) that is being read. To the left of relative byte 6ØH, The word "TRU" will appear and it stands for true. (Below it the true track number as it is read from the disk is given.) NOTE: (The explainations on relative bytes 4ØH, 5ØH, 6ØH, AND 7ØH may seem strange because the TRUE and RELATIVE track numbers are usually the same. However, on certain protected disks there are differences!) (* ND8Ø v2 Double Den users see footnote 3). To the left of relative byte 8ØH, the word "SEC" is shown and it stands for sector. To the left of relative byte 9ØH, the sector number that is currently being viewed is shown in decimal. To the left of relative byte AØH, the word "STD", "RPT","DDT", or "UDF" is shown, indicating whether the displayed sector contains <S>tandard, <R>ead protected, <D>eleted data, or <U>ser defined data (* Mod III users see footnote 4). To the left of relative byte BØH the letters "ISD", "NSD", "IDD", or "NDD" will appear. (The first letter will tell if the sector is <I>BM or <N>on IBM format. The "SD" or "DD" that follows indicates whether the sector is <S>ingle <D>ensity or <D>ouble <D>ensity). When you are in the DECRYPTING MODE, the letters "DCR" appear to the left of relative byte CØH, and its associated prompt at DØH. To the left of relative byte FØH, the numbers "+ØØ" appear and these are also explained in the DECRYPTING SECTION of this manual.

FIELD #3: This consists of 8 pairs of columns that contain the sector data (in HEX), and any byte may be modified by entering the MODIFY MODE and typing over the existing code that is presently highlighted by the flashing cursors (See MODIFY MODE instructions).

FIELD #4: this is an ASCII representation of the printable ASCII bytes contained in FIELD #3. All bytes less than 2ØH will be displayed as periods.

2.1B MODIFY MODE INSTRUCTIONS:

The MODIFY MODE has several options available and should be thoroughly understood prior to any "ZAPPING". Once your chosen sector has been displayed on the screen, any data byte shown may be modified using HEX, DECIMAL, BINARY, OCTAL, or ASCII input. To enter MODIFY MODE check the word located to the left of relative byte 10H to determine which mode the program is presently in and then make your mode selection by pressing "H", "A", "B", "O", or "D". After selecting the desired mode, press "M" and you will observe the dual cursors on the screen. Said cursors may be moved to any data byte on the screen by using the UP, DOWN, RIGHT, and LEFT ARROWS. <SHIFT><ARROWS> will move the cursor to extreme column or row ends. Holding down an arrow key will cause the cursors to continue moving in the direction of the arrow until released. Once the cursor is on the byte that is to be modified, type over the existing byte and the changes are automatically updated to both FIELD #3 and FIELD #4.

Upon completion of the desired changes push <ENTER> and you are then given the options of being able to <U>PDATE data, <R>ETURN to modify mode, or <C>ancel changes. NOTE: (All changes are kept in a buffer and are not written back to the disk unless the <U>PDATE option is selected, and you may return to the menu at any time by pressing <BREAK>, or return to the MAIN MENU by pressing <SHIFT><BREAK>). If you select the <C>ANCEL option then all changes are cancelled and you are returned to the same sector as it was prior to entering the MODIFY MODE. If you choose the <R>ETURN to modify mode option, you are returned to the same sector as it existed prior to pressing <ENTER>, and you will still be in the MODIFY MODE.

2.1C MODIFY MODE COMMAND LIST:

The following command list of options apply only while you are IN THE MODIFY MODE. Also see PAGING MODE COMMAND LIST as there are differences and you should familiarize yourself with these differences BEFORE getting into various ZAPPING procedures. Be sure to check the manual instructions on each of the utilities prior to using them.

1 - Right Arrow	Moves cursor one byte to the right
2 - Left Arrow	Moves cursor one byte to the left
3 - <SH> Right Arrow	Moves cursor to last byte right side of display
4 - <SH> Left Arrow	Moves cursor to last byte left side of display
5 - Up Arrow	Moves cursor up one row
6 - Down Arrow	Moves cursor down one row
7 - <SH> Up Arrow	Moves cursor to top row
8 - <SH> Down Arrow	Moves cursor to Bottom row
9 - <Clear>	Returns cursor to relative byte 00H of display
10 - <P>	Place cursor on field #3 byte and after pressing "P", follow with numeric input. Whatever byte that was UNDER the cursor will be recopied to the bytes following the number of times of the inputed numeric value (not active in ASCII modify)
11 - <Break>	Return to DISK ZAP MENU
12 - <SH><Break>	Return to MAIN MENU
13 - <SH><Clear>	Activate screen printer
14 - <SH><Enter>	Reset MODIFY MODE (Hex, Dec, Bin, Oct, Ascii)
15 - <G>	Plus relative byte number moves cursor to the specified RELATIVE BYTE LOCATION (not active in ASCII modify)
16 - <L>	Plus numeric input, moves the cursor to the NEXT OCCURANCE of the NUMERIC INPUT (not active in ASCII modify)
17 - ">"	Insert data and move the rest of the sector bytes to the right
18 - "<"	Delete the data that is being covered by the cursor and all remaining sector data is moved to the left.

2.1D PAGING MODE COMMAND LIST:

1 - Right Arrow	Pages one sector higher and will advance track
2 - Left Arrow	Pages one sector lower and will retard track
3 - <SH> Right Arrow	Pages one sector higher but won't advance track
4 - <SH> Left Arrow	Pages one sector lower but won't retard track
5 - Up Arrow	Pages one track higher
6 - Down Arrow	Pages one track lower
7 - <SH> ")"	Pages to highest sector on track
8 - <SH> "("	Pages to lowest sector on track
9 - <SH>Up Arrow	Pages to highest configured track
10 - <SH>Down Arrow	Pages to track #0
11 - <Break>	Returns to the menu
12 - <R>	Pages to Track 0, Sector 0
13 - <M>	Activates modify mode
14 - 0 to 9	Pages to a selected sector
15 - <SH><Break>	Returns you to MAIN MENU
16 - <SH><Clear>	Activates screen printer
17 - <Clear>	Allows reselection of a new DRIVE, TRACK, and SECTOR
18 - <T>	Allows reselection of a new TRACK and SECTOR
19 - <S>	Allows reselection of a new SECTOR
20 - <@>	Enter DECRYPTING MODE
21 - ">" or "."	Pages to next higher VALID SECTOR
22 - "<" or ","	Pages to the next LOWER VALID SECTOR
23 - <H><D><B><A><O>	Select modify type

Normally, when you are in the PAGING MODE, you will use the RIGHT and LEFT ARROW KEYS to page forward or backward to different sectors, however, there may be times when you will run into a disk that has been "PROTECTED" by assigning non-standard track and sector numbers. These disks may be identified by using the ADDRESS MARKS program in the ZAP utility. SUPER-UTILITY PLUS has a routine that will automatically page forward or backward, until it finds a VALID sector on a disk. This routine is activated by using the ">" key for forward or the "<" key for backward.

2.1E DECRYPTING:

The PAGING MODE gives you the ability to do various "BIT" ROTATING and SHIFTING operations. These options are activated by pressing <@> and then following it with the specific operation desired.

<1> - <R> : This option allows you to "ROTATE" the bits either to the right or left. This option does the same thing as the RLCA or RRCA command

in ASSEMBLY PROGRAMMING. If you wish to ROTATE LEFT 4 times, press <@> and then press <R>. (This means that you have decided to ROTATE) Now press <R> again (To ROTATE to the <R>ight), and then enter a number to indicate the number of times to ROTATE. If you had wanted to ROTATE LEFT, the LAST <SH><R> would have been a <L>.

<2> - <S> : This option allows you to "SHIFT" the bits either to the <L>eft or <R>ight and follows the same input as in the ROTATE instructions. THis option does the same thing as the SLA and SRL commands in ASSEMBLY LANGUAGE PROGRAMMING.

<3> - <A> : This option allows you to "AND" a value and requires a two digit input.

<4> - <O> : This option allows you to "OR" a value and requires a two digit input.

<5> - <X> : This option allows you to "XOR" a value and requires a two digit input.

<6> - <+> : This option allows you to "ADD" a value and requires a two digit input.

<7> - <-> : This option allows you to "SUBTRACT" a value and it also requires a two digit input.

One other major ability of the DECRYPTING program is to allow you to have the computer check sector data by incrementing or decrementing the data. This utility will perform these functions, update the ASCII side, HEX side, or BOTH sides of the screen and will RE-UPDATE the screen at a speed that is selected by you from several times per second to once a second. The following will be a list of possible options that are available.

<1> - <@>	Enter or Reset the DECRYPTING MODE
<2> - "*"	Update ASCII side only
<3> - ":"	Update HEX and ASCII sides
<4> - <ENTER>	Turn off or reset routine
<5> - <Up Arrow>	Increment value
<6> - <Down Arrow>	Decrement value

DECRYPTING MODE continued on next page.

The DECRYPTING MODE routine also has a DISPLAY SPEED ADJUSTMENT option included and the speed of updating the displays may be set by including a number from Ø to 255 when setting up for decrypting. EXAMPLE: You want to check to see if there are any words on a sector that have been encrypted by incrementing or decrementing their value. First, display the sector to be checked on the screen and then enter the DECRYPT MODE by pressing the <@> key at which time you will see the letters "DCR" appear to the left of relative byte #CØH. (DCR means that you are presently in the DECRYPTING MODE). Next, press either ":" if you want both the HEX and ASCII sides of the display to be updated or press "*" if you want only the ASCII side updated. Follow this by pressing either the UP or DOWN ARROW to increment or decrement the display. Lastly, you may enter a number from Ø to 255 to determine the speed of the screen updating. (a "1" is the fastest and "255" is the slower speed of updating the screen). This routine starts by adding or subtracting "1" to each byte of the display and automatically stops when it reaches FFH but it may be interrupted at any time by pressing the <AT> key, or paused with <SPACEBAR>. It is not necessary to supply a speed setting for this routine and the default speed value is approximately one updating each 1/2 second. An example of a minimum input selection after pressing the <@> key would be <UP ARROW> followed by <ENTER> which would begin incrementing the displayed sector at a rate of approximately 2 times per second.

2.1F VERIFY SECTOR

Selection of this option requires that you input the drive, track, and sector where you wish to begin verifying and the number of sectors that are to be verified. Pressing <ENTER> alone at the sector count question will default to the total number of sectors on the diskette. The routine will then read and verify each individual sector and advise you of the location of any sectors (if any) that it wasn't able to verify.

2.1G COMPARE SECTORS:

Choosing option #3 requires input of the the drive, track, and sector numbers of the SOURCE DISK. You will then be asked for a SECTOR COUNT. Press <ENTER> and then enter the the information on the drive, track, and sector numbers of the DESTINATION DISK. You will then be asked "Prompt for disk mounts?". If you answer YES, you will be prompted when to swap disks. NOTE: (All remaining memory is used as a buffer and the number of swaps is determined by how much RAM is available). Push <ENTER> and you are then told if the compared sectors are IDENTICAL. If the compared sectors are NOT IDENTICAL, you will be given the location of the first mismatched byte, and upon completion, the number of DATA mismatches, and number of DAM mismatches.

2.1H COPY SECTORS:

Selection of this option also requires you to input the drive, track, and sector numbers of the SOURCE DISK and the number of sectors to be copied. Push <ENTER> and you will be prompted to input the DESTINATION drive, track, and sector numbers. You will then be asked if you want to be prompted for disk mounts. This utility operates with the same input routines as COMPARE SECTORS for one or multiple drive systems and it too uses all available memory on copies.

2.1I COPY SECTOR DATA:

Selection of this option requires input of the SOURCE drive, track, and sector. You will then be asked to supply the RELATIVE BYTE # where the copy is to BEGIN and then the number of bytes to be copied. Upon completion of these inputs, you will then be requested to enter the drive, track, and sector information on the DESTINATION drive. You will be then be asked if you want to be "prompted for disk mounts". Upon completing the data transfer, you will be advised if any I/O errors occured during the transfer. This utility will allow you to copy from 1 to 65535 bytes of data at a time to the location of your choice.

2.1J ZERO SECTORS:

Upon selection of this option, you are prompted to input the drive, track, and sector where you want to begin zeroing and the number of sectors that are to be ZEROED. WARNING!!! UPON PRESSING <ENTER> THE SELECTED SECTORS ARE IMMEDIATELY ZEROED AND THERE IS NO CHANCE OF DATA RECOVERY FROM THE SELECTED SECTORS. NOTE: (the selected sectors will be rewritten with all ZERO'S, but in STANDARD DAM'S when this routine is finished.)

2.1K REVERSE SECTOR DATA:

This utility simply takes the sectors of your choice and completely REVERSES the data on it so that the data that WAS contained in relative byte #ØØ will NOW be contained in relative byte #FF etc. The ability to automatically perform this function has definite possibilities pertaining to disk protection of your machine language programs.

2.1L EXCHANGE SECTORS:

Option #8 asks you to input the SOURCE drive, track, and sector numbers along with the sector count to be exchanged. Then you are requested to supply the DESTINATION drive, track, and sector where you want the exchange to BEGIN. Upon completing these steps, you will be given the option of having "disk mount prompts" and the selected sectors will then be transfered between the two disks. You will also be advised if any I/O errors occured during the transfer of information.

2.1M STRING SEARCH:

This utility will allow you to search the DISK for a given ASCII STRING, BYTE LIST, or WORD LIST. If the DISK is to be searched, it will require input of the drive and sector where you want to BEGIN searching. You will then have to input the number of sectors to be searched. Next, input the string to be searched for and lastly, you will be asked to supply a "Replacement string". (If you do not want to replace the string with another, press <ENTER>). If you DO wish to replace the string with another of your choice, then simply enter the string and press <ENTER>. Upon completion of the routine, the location of each match will be displayed to the screen along with the total number of matches that were found. NOTE: The ENTIRE string MUST be contained in a single sector for this routine to work properly, so if it doesn't find the string, try entering a portion of it. If MEMORY is to be searched, then go to the MEMORY UTILITY section of this manual. You may also use this utility to search for a BYTE or WORD LIST by first typing a "#" or "##" PRIOR to entering the two numbers. The following three explainations of the different types of searches, will include ASCII, BYTE LIST, and WORD LIST searches. ASCII is self explanatory, but the other two may not be so readily understandable to some users. A BYTE LIST is a list of numbers, where each numbers value is from Ø to 255 and may represented by ONE BYTE. A WORD LIST is a list of 2 BYTE NUMBERS whose values may be from Ø to 65535, and is entered in LSB-MSB format.

STRING SEARCH continued on next page.

This utility may be used to search the disk in three different types of searches. The following will attempt to explain how each of the three types of search is used.

<1> - The ASCII STRING SEARCH may be used to locate any combination of ASCII bytes that are on the disk. First, you will have to enter the drive, track, and sector where you want the search to begin and the sector count (number of sectors to be searched. - Pressing <ENTER> will result in the entire disk being searched). Next, you are requested to enter the STRING that is to be searched for and upon entering it, the "REPLACEMENT STRING. If you DO NOT want to replace the string with a different one, simply press <ENTER> as the default value is "NO". The disk will then be searched and you will told the location of any "MATCHES" on the disk.

<2> - The BYTE LIST SEARCH may be used to locate a byte or byte list on a disk. First, you will be required to input the drive, track, and sector where you want the search to begin and the sector count (Pressing <ENTER> defaults to entire disk). Next, when you are requested to enter the STRING, enter "#" and the byte list to be searched for (the byte list may contain a single byte or several that are separated by a space or comma. An example entry after inputing the drive, track and sector information would be "#2ØH" or "#2ØH,1ØH,7FH"). Lastly, you are asked if you want to REPLACE the string and you may do so if you wish by following the ASCII search directions.

<3> - The WORD LIST SEARCH option allows you to search for 2 byte word lists on the disk. When you are requested to input the STRING, enter "##" PRIOR to the word list. A word list is simply 2 bytes of data instead of 1 as in the BYTE LIST SEARCH. When you are using this option, enter the 2 bytes in REVERSE ORDER as that is the way it is stored on the disk. An example of this type of search input after entering drive, track, and sector information would be "## 7FØØH" or "## 7FØØH,ØØ3CH". This utility has the same replacement options as in <2>.

<4> - If the string that is to be searched for is ENCRYPTED, then BEGIN YOUR STRING using the <@> KEY as the FIRST LETTER OF THE STRING and the routine will then search for the string in its ENCRYPTED FORM.

There is one more option that may be used in any of the above searches, and its called the "I DON'T CARE" option. When inputting the string to be searched for in any of the above options, you may use the character "?" in place of any character in the string and the routine will then return the location of any string that matches the rest of your string but is different at the "?" characters location. EXAMPLE: You want to find all the locations of the ASCII strings "Jim" and "Tim". When requested to input the STRING, enter "?im" and upon completion of the routine, all matches of ASCII words with the letters "im" that are preceded by another letter will be returned to you.

2.1N SECTOR SEARCH:

Use of this option will require input of a SOURCE drive, track, and sector. Upon completion, you will then be asked to supply the drive, track, and sector numbers where to search is to BEGIN. Finally, input the number of sectors to be searched and you will then be told if there are any "MATCHES" of your source sector and if so, where their locations are on the destination disk.

2.1O READ ID ADDRESS MARKS:

This routine may be used for a number of different things, but its main function is advise you if a disk contains FALSE or NON-STANDARD SECTORS and to assist you in determining the way that a disk was formatted. The utility starts reading the disk at relative track #ØØ and will advance ONE track higher each time the <UP ARROW> is pressed. The display will move quite rapidly but may be stopped at any time by pressing the <SPACEBAR> and may be continued by pressing <ENTER> again. This routine may be exited by pressing <BREAK>. Additionally, you may proceed immediately to either the HIGHEST or LOWEST tracks on the disk by pressing either <SH><UP ARROW> or <SH><DOWN ARROW>. If the computer finds an UNFORMATTED track, the display will print " ID ADDRESS READ ERROR" or "TRACK NOT FORMATTED".

You will observe 7 columns of information scrolling down the screen. 3 more columns of information may be obtained by pressing "X". The following is a list of what each of the columns are and what information is supplied by each one, starting from the left hand side of the screen.

continued on NEXT PAGE

SOURCE	TRACK	HEAD	SECTOR	LENGTH	CRC1	CRC2	CKCRC	IBM	DATA	#
:1S= 00	00	00	06	01	5BH	75H	YY	Y	STD	
:1S= 00	00	00	03	01	A4H	80H	YY	Y	STD	
:1S= 00	00	00	09	01	4BH	4BH	YY	Y	STD	
:1S= 00	00	00	01	01	C2H	E2H	YY	Y	STD	
:1S= 00	00	00	07	01	68H	44H	YY	Y	STD	
:1S= 00	00	00	04	01	3DH	17H	YY	Y	STD	
:1S= 00	00	00	05	01	0EH	26H	YY	Y	STD	
:1S= 00	00	00	02	01	97H	B1H	YY	Y	STD	
:1S= 00	00	00	08	01	78H	7AH	YY	Y	STD	
:1S= 00	00	00	00	01	F1H	D3H	YY	Y	STD	
:1S= 00	00	00	09	01	4BH	4BH	YY	Y	STD	
:1S= 00	00	00	01	01	C2H	E2H	YY	Y	STD	
:1S= 00	00	00	07	01	68H	44H	YY	Y	STD	
:1S= 00	00	00	04	01	3DH	17H	YY	Y	STD	

1 - SOURCE: This column advises you of what drive you are presently viewing and whether the track you are viewing is <S>ingle or <D>ouble density. :ØS = DRIVE#Ø with a SINGLE DENSITY track being displayed. If an error occurs while attempting to read the address marks, a message will be displayed, and the density will be reversed.
2 - TRACK: This column advises you of what track # (as read from the disk) is now on display.
3 - HEAD: This is the head number that is recorded on disk.
4 - SECTOR: Displays the sector numbers as they are read from the disk. These numbers are not listed in any particular order and will appear in sequencial groups of 3 or 4. If there are any NON-STANDARD sector numbers, they should be quite easy to spot.
5 - LENGTH: IF IBM, "Ø"=128 else LENGTH * 256 - If NOT IBM, LENGTH * 16 "Ø"=4Ø96. NOTE: The length numbers indicate the recorded length in the ID field and not necessarily the actual number of bytes in the sector.
6 - CRC1: Checksum of the bytes in the ID FIELD.
7 - CRC2: Checksum of the bytes in the DATA FIELD.
8 - CKCRC: Computes and compares CRC1 & CRC2 with actual data that is on the disk. The first "Y" indicates if CRC1 is correct, and if so, the second "Y" indicates if CRC2 is correct. If CRC1 is non valid, a "*" will be printed in column #8.
9 - IBM: This column advises you if the sectors are formatted in IBM format or not. If the sector IS in IBM format, a "Y" will appear. If the sector IS NOT in IBM format, a "N" will show.
1Ø - DATA: The DATA ADDRESS MARK (DAM) column will contain one of four possible words which will advise you if the sector that is presently being viewed has been formatted to store Std (STANDARD), Rptc (READ PROTECTED), Ddt (DELETED DATA), or Udf (USER DEFINED) data. (* Mod III users see footnote 4)

2.1P - ALTER DATA ADDRESS MARKS:

Selection of this utility will require you to input the drive, track, and BEGINNING SECTOR where you want to begin altering the data address marks. Next, enter the number of sectors that are to be changed and you will then have to select the type of data address mark that you want on the effected sectors. Your choice includes S>td (STANDARD), R>ptc (READ PROTECTED), D>dt (DELETED DATA), or U>df (USER DEFINED). (Mod III users see footnote 4)

2.2 PURGE OPERATIONS MANUAL:

Upon selection of the PURGE UTILITY, the ten following options will be displayed to the screen.

1. Kill Selected Files
2. Kill by Category
3. Remove System Files
4. Remove All Passwords
5. Disk Directory
6. Zero Unused Entries
7. Zero Unused Granules
8. Change Disk Name
9. Change File Parameters
0. Check Directory

Selection ?

2.2A KILL SELECTED FILES:

Selection of this option requires input of the drive number. The directory is then read (* ND80 v2 users see footnote 5) and you will be told the NAME and DATE of the DISK, along with the NUMBER of TRACKS and FREE GRANULES (* ND80 v2 and Mod III TRSDOS users see footnote 6). Press <ENTER> and the entire disk directory will be displayed including ACTIVE and NON-ACTIVE (see following example display). The ACTIVE files are contained with RIGHT AND LEFT ARROWS (left and right square brackets in the Mod III) and the NON-ACTIVE files are contained in GRAPHIC BLOCKS. In the upper left corner of your video display you will notice a CURSOR and it may be moved around the screen by pressing the ARROW KEYS. If you are examining a disk with more files on it than can be displayed on a single screen (Double Density and ND80 extended directories), you may TOGGLE to PAGE #2 by pressing <SH><UP ARROW>. To return to PAGE #1, press <SH><DOWN ARROW>.

At this point, you may either KILL files by placing the cursor on the file to be killed and then pressing <K>ill, or files that have already be killed may be RESTORED by pressing <R>estore. You may also kill a file and <C>lear its directory entry by pressing C. Pressing <CLEAR> at any time will zero ALL unused directory entries. These changes are held in a buffer until you press <W>rite to disk, at which time you will be asked if you really want to <W>rite the changes back to the disk (answer "Y" or "N"). If you wish to <A>dvance to the next specified drive WITHOUT writing back the directory, press A. You may return to the PURGE MENU at any time by pressing <BREAK> or return to MAIN MENU by pressing <SH><BREAK>.

When requested to input the drive #, you may input MORE THAN ONE if desired, and the above directions would still apply except the SECOND SELECTED DRIVE would be displayed upon completion of the FIRST DRIVE changes. Press <W> to write the current directory and advance to the next drive, or <A> to advance to the next drive WITHOUT updating the current directory.

The following is a COMMAND SUMMARY of KILL SELECTED FILES.

<1> - <K>	KILL file that cursor is on
<2> - <C>	KILL file and zero directory entry
<3> - <R>	RESTORE file that cursor is on
<4> - <W>	WRITE changes back to disk
<5> - <N>	NO action, go to next file
<6> - <A>	ADVANCE to next drive

2.2B KILL FILES BY CATEGORY:

Use of this utility requires input of the drive(s) number(s) and after doing so, you then have to supply the "COMMON CATEGORY" of the files that are to be killed. Common category refers to the filename EXTENSION (/BAS, /CMD, /TXT, etc). NOTE: (Don't forget the "/" separater). NOTE: (if you specify "/SYS" as the common category, ALL SYSTEM FILES INCLUDING BOOT and DIRECTORY will be killed !! and this could be fatal to your disk. Use option 3 to safely remove the system files).

This utility also contains a "WILD CARD" option where you may input a LETTER ALONE and all files that START WITH THE LETTER will be KILLED. If you accidentally purge a file(s), you may recover it/them by going to the KILL SELECTED FILES option and using the <R>estore command on the file(s) that you want to restore.

You may also kill classes of files depending on directory bits. For the common category, enter a <SPACE> as the first character, followed by either V, I, S, or P to kill Visible, Invisible, System, or files with a protection level other that 0 respectively. (* Mod III TRSDOS users see footnote 8).

2.2C REMOVE SYSTEM FILES:

The only input needed for this utility is the drive(s) number(s). Press <ENTER> and ALL SYSTEM FILES (except BOOT and DIRECTORY) are instantly "KILLED". NOTE: Any files that are killed with "SUPER-UTILITY PLUS" may be reinstated at any time by using the disk repair option as long as the sectors or unused directory entries haven't been "zero'd" or the granules haven't been allocated to a new file. (EXCEPTION: Mod III TRSDOS system files that are killed cannot be restored).

2.2D REMOVE ALL PASSWORDS:

This utility simply requires you to input the drive(s) number(s). Press <ENTER> and all files will be set to NIL passwords (both UPDATE and ACCESS passwords), and the protection level will be set to 0 (full access), and the MASTER PASSWORD will then be "PASSWORD"

2.2E DISK DIRECTORY:

The disk directory utility, after input of the drive(s) number(s), will supply you with the NAME and DATE of the disk, number of TRACKS and FREE GRANULES, and the number of FREE FILES that are available on the disk (* see footnote 6). In addition, you will be shown EVERY VALID FILE on the disk. This will include each files PROTECTION LEVEL (if any), and whether it is a SYSTEM FILE, VISIBLE, or INVISIBLE. The method of advising you of the status of each file is in standard TRSDOS format (S=SYSTEM, I=INVISIBLE, P+NUMBER=ATTRIBUTE LEVEL).
EXAMPLE: "SYSØ/SYS SIP=7" indicates that the file "SYSØ/SYS" is a <S>ystem file that is <I>nvisible, and has a <P>rotection level of 7.

If the disk contains a large number of files that cannot be displayed on a single screen, press <ENTER> and you will be shown the next page of files. If more than one drive was originally selected, the information for the second drive will automatically start upon completing the first drives information. (* Mod III TRSDOS users see footnote 7).

2.2F ZERO UNUSED DIRECTORY ENTRIES:

This utility is excellent for "cleaning up" the directory and only requires input of the drive(s). Having done so, press <ENTER> and all non active files are immediately ZEROED out. WARNING !!!! Once this done, you will not be able to recover non active files without engaging in the wonderful pastime called "mucking thru the disk".

2.2G ZERO OUT UNUSED GRANULES:

This utility is also an excellent for "cleaning up" a disk, and it too only requires input of the drive(s). Press <ENTER> and all NON-ALLOCATED GRANULES are immediately ZEROED out. Once it has been used, there is absolutely no chance of data recovery from the AFFECTED GRANULES of the disk. NOTE: (all ZEROED sectors are written as standard data regardless of their original format.

2.2H CHANGE DISK NAME:

This routine asks for input of the drive(s) and you will then be told the existing NAME of the disk and asked to supply a new one (Pressing <ENTER> alone, defaults to "NO CHANGE"). Next, you will be told the existing DATE of the disk and requested to supply another one. Next, you will be given the MASTER PASSWORD of the disk and given an opportunity to change it. Pressing <ENTER> here will set the master password to PASSWORD. Lastly, you will be told if the disk is set for an AUTO COMMAND and you may also alter that to a DIFFERENT program. Pressing <ENTER> here will set the auto command to NIL.

2.2I CHANGE FILE PARAMETERS:

Use of this option requires the user to input a complete filename, and upon completing the input, the directory will be read, and the filename located. Next you are asked for the new name. Press <ENTER> alone to leave it unchanged. Next you will be asked for the new ACCESS password. Pressing enter will set it to NIL. Then you will be asked the same about the UPDATE PASSWORD. Then you are prompted for the Protection Level that the file is to be set at. It may be from Ø to 7 (excluding 3 which is unused). Finally, you are requested to input the files Attributes. Specify V (for visible, non-system), I (for invisible, non-system), or S (for invisible, system). At this time the updated information will be written back to the disk.

2.2J CHECK DIRECTORY:

Input the drive(s) to be checked and you will then be advised of the disk NAME, DATE, number of TRACKS, FREE GRANULES, and FREE FILES (* see footnote 6). In addition, a very complete check of the directory of the selected disk is performed and you will be told if any errors exist in the HIT or GAT SECTORS of it. If errors are discovered in either of these places, you may have SU+ automatically repair these problems for you by going to the MAIN MENU and selecting the DISK REPAIR option. Pressing <BREAK> will return you to the PURGE MENU and pressing <SH><BREAK> will return you to the MAIN MENU. You will also be notified if any files contain invalid directory information, and check for properly linked extended directory entries.

2.3 DISK FORMAT UTILITY OPERATIONS MANUAL:

Selection of the DISK FORMAT utility will result in the following six options being displayed to the screen.

1. Standard Format
2. Special Format
3. Format Without Erase
4. Build Format Track
5. Write Format Track
6. Software Bulk Erase

Selection ?

2.3A STANDARD FORMAT:

Upon selecting this utility, you are asked to supply the Drive(s), NAME, DATE, and MASTER PASSWORD of the disk(s) to be formatted, and if you wish to use the CONFIGURATION for which the system has been set. (NOTE: if the set configuration is not valid, i.e., the directory track indicated is outside the disk boundaries, then a NO answer will be forced.

If your response is <Y>es, the routine will first attempt to read the disk that is to be formatted and if the first sector is READABLE, the directory will attempt to be read and you will be given EXISTING DISK INFORMATION. (* see footnote 20) If the directory WASN'T READABLE you will also be advised. You will then be given the options to <C>ontinue with the formatting or <S>kipping the formatting and returning to the FORMAT MENU or <K>eep same density.

If your response is <N>o, it will result in the prompt ":1 DEN, TRKS, DIR, ST TK?" being displayed. At this point you may CUSTOM FORMAT the disk by selecting the DENSITY of the disk, number of TRACKS, track number where the DIRECTORY will reside, and the STARTING TRACK number where the formatting is to begin. EXAMPLE: You have a DISK that you wish to format as 40 TRACK, DOUBLE DENSITY, and you want the DIRECTORY on TRACK #20. You would now input "D, 40, 20", and then press <ENTER> to accomplish the format.

There are 2 OVERRIDE COMMANDS that may be used in this routine when you are asked "USE CONFIGURATION ?", and they are the "*" and "!". The "!" will force the routine to format the disk WITHOUT attempting to read the directory, and the "*" will force the routine to NOT VERIFY the disk during formatting. The diskette will simply be formatted, and the directory and boot written. Using either of these two override options will start the formatting process to begin as soon as you press <ENTER> from answering the "USE CONFIGURATION ?" prompt. Using ALL DEFAULT VALUES in responding to the prompts will result in a disk with the name "DataDisk", a password of "PASSWORD", and a date of "00/00/00". These values may be altered using HARD CONFIGURATION. If you specified a starting track OTHER THAN 0, then you will be prompted if you want the system information written to the disk (the BOOT and DIRECTORY). When formatting is completed, you will be asked if you wish to REPEAT? (answer "Y" or "N").

You may add tracks to an existing disk easily with this routine (example: change a 35 track to a 40 track disk, or an 80 to an 82 track.)

Answer <N>o to use configuration, specify the total number of tracks, and a starting track of the current track count. When you are asked if the directory is to be written to the disk, BE SURE that the CONFIGURED directory track is the same as the REAL directory track. If not, the disk may be rendered unusable. A safer way is to specify NO to the question, then use the REPAIR GAT table option to open up the additional tracks.

2.3B SPECIAL FORMAT:

This utility was designed to allow you to CUSTOM FORMAT a disk in ways that are only limited by your imagination. You may assign as many sectors as you wish to any track and make any of them either VALID OR FALSE. Further, you may make tracks SINGLE or DOUBLE DENSITY in any combination or even have the routine assign ALL RANDOM TRACKS AND SECTOR NUMBERS to your disk.

The main purpose of this utility is to assist professional programmers who wish to protect their work and at the same time, give the novice some understanding of how various protected disks are put together.

You will first be asked to input the DRIVE NUMBER that contains the disk that is to be formatted, and upon doing so and pressing <ENTER>, you will see the work "TRACKS ?". Input the number of tracks that are to be on the disk and press <ENTER>. Next, the prompt "T=00, S=00" will appear on the screen and you may begin inputting TRACK and SECTOR information as explained in the following manner.

The "T=00" tells you the REAL TRACK NUMBER that is now being built is "00". The "S=00" tells you that the NUMBER of SECTORS that have ALREADY BEEN BUILT on the track. There are several different commands that may be used in building custom formatted tracks and sectors and the following is a list of those SPECIAL FORMAT COMMANDS.

1. <S> - This letter, when followed by a number, tells the routine that this <S>ector is to be called by the NUMBER that follows it. EXAMPLE: "S5" means that the sector is to be called SECTOR #5.
2. <T> - This letter, when followed by a number, tells the routine that this sector is to called by the NUMBER that follows it. EXAMPLE: "T2" means that the SECTOR that is now being built, is to be called track #2.
3. <L> - This letter, when followed by a number, tells the routine that the SECTOR that is being built is to be the <L>ength of the NUMBER that follows the "L".
4. <M> - This letter, when followed by the letter "S", "U", "R", or "D" tells the routine that the SECTOR DATA MARK is to be <S>tandard, <U>ser defined, <R>ead protected, or <D>eleted. (* Mod III users see footnote 4).
5. <C1> - This letter and number, when followed by another NUMBER, determines the ID FIELD (CRC1) NUMBER. If this number is anything other than "F7", the sector will be considered FALSE and nothing will be written to the DATA FIELD, however the sector will still reside on the track.
6. <C2> - This letter and number, when followed by another NUMBER, determines the DATA FIELD CHECKSUM of the sector.
7. <I> - This letter, if followed by a "Y", tells it that "Yes, the sector is IBM". If followed by a "N", then the sector is "NOT an IBM sector".
8. <R> - If this letter is used as the ONLY INPUT for the sector, then everything will be <R>andom. If the <R>andom command is used on all

the sectors and the "REPEAT ?" prompt is answered <Y>es, then each disk produced will be completely identical.

9. <D> - This letter, when followed by "D" or "S", tells the routine that "the TRACK is <D>ouble density", or "the TRACK is <S>ingle density".

Pressing <ENTER> without inputting any data, will advance to the NEXT TRACK and start again at Ø sectors. Lastly, you may format some tracks as single density and others as double density as the utility will allow SINGLE OR DOUBLE DENSITY FORMATS, BUT YOU MAY NOT CHANGE THE DENSITY IN THE MIDDLE OF A TRACK.

EXAMPLES OF CUSTOM FORMATTING AFTER THE "T=ØØ, S=ØØ" PROMPT.

EXAMPLE #1. TØØ S12 MU L64 IN DD C1ØØ <ENTER> In the above sector description, "TØØ" means that the <T>rack NUMBER is "ØØ", "S12" means the <S>ector is to be called "12", "MU" means the DATA <M>ark is to be <U>ser defined, "L64" means the sector is to have a <L>ength of "64", "IN" means the sector is <N>ot to be <I>BM type of format, "DD" means that <Y>es, it is to be <D>ouble density, and lastly, "C1ØØ" means that the CR<C1> ID FIELD is to contain the value "ØØ" and will make this a FALSE SECTOR.

EXAMPLE #2. T24 S123 LR MR <ENTER> In this sector description, "T24" means that the <T>rack number FOR THIS SECTOR is "24" and the "S123" will cause the <S>ECTOR to be called "123". "LR" tells it that the <L>ength of the sector is to be <R>andom and "MR" tells it that the data <M>ark also is to be <R>andom.

EXAMPLE #3. R <ENTER> This ONE LETTER DESCRIPTION tells the routine to make EVERYTHING IN THE SECTOR COMPLETELY <R>ANDOM.

2.3C FORMAT WITHOUT ERASE:

This utility requires you to input the DRIVE(S) NUMBER(S) that you want formatted just as was required in the standard format utility. (However, in this routine, you may RE-FORMAT a disk that has existing data on it and still leave said existing data intact). If you wish to REFORMAT the disk and leave the SAME NUMBER OF TRACKS, press <ENTER> to proceed. If you wish to INCREASE the number of tracks on the disk, use the following procedure.

When you FIRST INPUT the DRIVE to be REFORMATTED, also input a "=" and the number of tracks desired. EXAMPLE: To make a disk in drive #Ø, with LESS THAN 4Ø tracks into a disk WITH 4Ø tracks, input the following when FIRST prompted to input the DRIVE. "Ø=4Ø <ENTER>". The program will then proceed to REFORMAT the disk WITHOUT LOSS OF EXISTING DATA THAT WAS ORIGINALLY ON THE DISK.

This program will read each track, one sector at a time and then re-format and re-write the same sectors back on the track. If it reaches a track that is not formatted, it merely continues formatting until the specified track count is correct and then "FIXES" the directory. Lastly, this utility is also very handy for RE-VITALIZING those older disks that have been sitting for long periods of time, but it is a good idea to do a DIRECTORY CHECK (See DISK REPAIR UTILITY) when DECREASING the number of tracks on a disk as any data that was resident on the tracks that had been eliminated would be lost.

2.3D BUILD FORMAT TRACK:

This utility will BUILD a formatted track in memory and allow you to make the MODIFICATIONS of your choice. Upon choosing this option, you will be asked for the Dos Specifier to be used, which will imply the density. Next, the prompt "TRACK ?" is displayed, to which you input the TRACK NUMBER that you want it to be called and press <ENTER>. You will then be prompted to press <ENTER> again and the BUFFER AREA that holds the track will then be displayed to you for examination or modification. After making any desired changes by using the MODIFY MODE (you remember the modify mode instructions don't you?). When all changes are complete, press <BREAK> to return to the FORMAT MENU and then go to the WRITE FORMAT TRACK option to transfer the track to the disk.

2.3E WRITE FORMAT TRACK:

This routine will WRITE A TRACK FROM MEMORY TO DISK that was prepared with the BUILD FORMAT TRACK option, READ TRACK OPTION, or memory modify. Upon selecting this utility, you will be asked to input the DRIVE and TRACK NUMBER where you want the prepared formatted track to go on the disk. Remember that you were given the option of NAMING THE TRACK NUMBER in the BUILD FORMAT routine, and NOW you are required to supply the DRIVE and TRACK number where the track is to go (they do not have to be the same!). The routine will take the prepared track and put it at the location you are now indicating, BUT it will still have the NUMBER from the ORIGINAL preparation.

2.3F SOFTWARE BULK ERASE:

This neat little utility will take a disk and COMPLETELY REMOVE ALL DATA FROM IT. If you have ever had problems with SOME OPERATING SYSTEMS that just don't seem to like to cooperate with BACKUPS or FORMATTING because of data that is already on the disk, you're going to LOVE this routine. Simply place the offending disk in the drive of your choice, advise the utility of the DRIVE NUMBER, press <ENTER>, and then just sit back and relax. This program will completely eliminate the need for the BULK ERASERS or MAGNETS that you have been using in the past to correct these problems and were afraid to bring into the COMPUTER ROOM for fear of damaging valuable disk data. The diskette will be formatted to contain ALL Ø's. This will remove all traces of sectors from the disk.

2.4 DISK COPY UTILITY OPERATIONS MANUAL:

Selection of this utility results in the following two options being displayed to the screen.

1. Standard Disk Backup 2. Special Disk Backup

Selection ?

2.4A STANDARD DISK BACKUP:

This routine will, in effect, make a copy of almost any IBM formatted disk as long as the track and sectors are numbered sequentially from Ø on up. Most protected disks may be "BACKED UP" by use of this program and you may determine if a disk is numbered sequentially by use of the ZAP or READ DISK MARKS programs that are included in this program package. Use of this utility requires you to input the source and destination drives.

ONE DRIVE SYSTEM: If the source and destination drives are the same, you are prompted when to make the necessary disk SWAPS to complete the disk copy. NOTE: The entire unused memory is used as a buffer during this routine.

TWO DRIVE SYSTEM: When source and desination drives are different, simply place a blank disk in the destination drive and the disk to be copied in the source drive. Press <ENTER> and the copy will be automatically completed.

This routine uses ALL AVAILABLE MEMORY during data transfer from source to destination disks. Prior to any copying, the destination disk is formatted to the number of tracks specified if requested.

2.4B SPECIAL DISK BACKUP:

Selection of this utility requires input of the SOURCE DRIVE. Upon pressing <ENTER>, the program will begin scanning the SOURCE DISK and you will be able to observe the screen as it constantly updates during the scanning process. You are also told the number of tracks on the source disk along with the type of format (IBM NON-IBM). This routine is COMPLETELY AUTOMATIC and requires no further user input during its scanning operation. After the SOURCE DISK has been "FIGURED OUT", you will be asked to what drive(s) you want the SOURCE DISK to be copied. Upon completion, you will then be asked if you desire any more copies and if you do, simply input the drive(s) number(s) where you want the additional copies made. NOTE: The SCANNING PROCESS may take a little time but once it has finished, it doesn't have to scan again for additional copies.

2.6 DISK REPAIR UTILITY OPERATIONS MANUAL:

Selection of this utility will result in the following ten options being displayed to the screen.

1. Repair GAT Sector
2. Repair HIT Sector
3. Repair BOOT Sector
4. Read Protect Directory
5. Un-Read Protect Directory
6. Recover Killed Files
7. Move Directory
8. Display Directory
9. Check Directory
Ø. Clear Unused Entries

Selection ?

2.6A REPAIR GAT SECTOR:

If you find that an error exists in the "GAT" (GRANULE ALLOCATION SECTOR), select this option and input the drive(s) number(s). You will then be asked the the GAT table, or the entire sector is to be repaired. If the sector contains garbage, then select ALL to completely rebuild it. If just the allocation table is wrong, then select GAT table only. This will allow you to keep the existing name, date, auto command and master password. Press <ENTER> and the GAT SECTOR will IMMEDIATELY BE REPAIRED.

2.6B REPAIR HIT SECTOR:

If it is found that an error exists in the "HIT" (HASH INDEX TABLE), select this option and input the drive(s) number(s) to be repaired. Press <ENTER> and the directory will be read and the "HIT" table will be COMPLETELY RECONSTRUCTED without any further user intervention. (* Mod III TRSDOS users see footnote 9).

2.6C REPAIR BOOT SECTOR:

If you have a disk that will not BOOT, insert the disk into a drive, select this option, input the drive #, and press <ENTER>. At this point, a new BOOT will be re-written to the disk. NOTE: (the BOOT that is written to the disk will be either the one that you selected when you CONFIGURED SUPER-UTILITY PLUS or the FAST BOOT that came with this program if you DID NOT choose to use your own. This routine DOES NOT log the boot to the directory as most standard DOS-type disks will already have the BOOT logged in the directory). When using this option, 2 BOOTS will be written to the disk. BOOTA will be written to sector Ø, and BOOTB will be written to sector 1. This is to allow compatability with the Mod III systems, as the Mod III will boot sector 1, as compared to sector Ø for Mod I. (* Mod III TRSDOS users see footnote 1Ø).

2.6D READ PROTECT DIRECTORY TRACK

If you try to boot in a diskette, and the drive seems to be making a lot of passes and finding nothing, there is a possibility that some or all of the directory track is not read-protected. This does not mean that it cannot be read, but is a type of marking that sets it apart from the remainder of the disk which contains data. Whenever DOS tries to read a directory, and cannot verify that all sectors are read-protected, it will search the disk trying to find a track that is read protected. You may repair this condition by simply entering the drive(s) number(s) and pressing <ENTER>, at which time, the directory will automatically be READ PROTECTED. If the directory cannot be located, you will be asked to supply the directory track and the number of sectors. (* Mod III users see footnote 11).

2.6E UN-READ PROTECT DIRECTORY:

Selection of this routine will require that you input the drive(s), and upon pressing <ENTER>, the directory will immediately be UN-READ PROTECTED (set to standard data).

2.6F RECOVER KILLED FILES:

Selection of this utility allows you to reinstate previously killed files that have been killed by SUPER-UTILITY, or most non TRSDOS systems. NOTE: (you cannot recover files with this utility killed under TRS-DOS, as the entire directory entry is ZEROED out when the file is killed.)

You are first asked to input the drive(s) and you are then told the DISK NAME, DATE, NUMBER OF TRACKS, NUMBER OF FREE GRANULES, and NUMBER OF FREE FILES that are available. Press <ENTER> and you will be presented with a full screen display that consists of all VALID and NON VALID files that are presently on the disk. All ACTIVE FILES will be contained with a <RIGHT> and <LEFT> ARROW (Mod III will be shown with the left and right square brackets), and all NON ACTIVE (previously killed) files will be contained in RECTANGULAR GRAPHIC BLOCKS. To RECOVER KILLED FILES, move the cursor to the file that is to be recovered by using the ARROW KEYS and once the cursor is LOCATED ON TOP of the file, press <R>estore and you will see the graphic blocks replaced by the right and left arrows on the screen. When the files to be recovered have all been changed in this manner, press <W>rite, and you will then be asked if you want to WRITE the changes back to the disk (answer "Y" or "N"). NOTE: If the disk you are examining contains more files than can be displayed on a single screen, press <SH><UP> or <SH><DOWN> ARROWS to go to next page of display. Also, you may not be able to restore a killed file if you have SAVED other programs on the disk AFTER killing it, as the file may have been WRITTEN OVER.

2.6G MOVE DIRECTORY:

This utility will move the directory to any track on the disk as long as the track IS NOT ALLOCATED to another file. (* DOUBLEDOS users see footnote 12). After selecting this routine, input the drive(s) that you want to move the directory and you will then be asked to input the TRACK WHERE YOU WISH THE DIRECTORY TO BE MOVED. If the selected track is ALREADY ALLOCATED, you will be told and then requested to supply a different track number. Upon entering the track and pressing <ENTER>, the directory will be moved and the BOOT SECTOR will also be updated. As always, you may exit this routine by pressing <BREAK> to return to DISK REPAIR MENU or <SH><BREAK> to return to MAIN MENU.

2.6H DISPLAY DIRECTORY:

This option asks for input of the drive(s) number(s) and it then gives you a full screen display of all ACTIVE FILES on the disk (* Mod III TRSDOS users see footnote 7). Also the status of each file (system, visible, protection level, ect.) is supplied. In addition, you are given the disk name, date, and the number of free granules and files.

2.6I CHECK DIRECTORY

This utility was designed to thoroughly examine the directory for errors and then report them (if any) to the user. Upon entering the drive(s) number(s) and pressing <ENTER>, you will be told the number of TRACKS on the disk, the number of FREE GRANULES, the number of FREE FILES, and you are told if there are any ERRORS in the directory. NOTE: If any errors are found to exist in the HIT or GAT sectors on the directory, use the DISK REPAIR utility to automatically repair them.

2.1J CLEAR UNUSED ENTRIES:

This option simply asks you to supply the drive(s) number(s) and after doing so, press <ENTER> and ALL NON-ACTIVE FILES on the disk will be ERASED from the directory. This routine is great for cleaning up the directory, but once you have used this routine on a disk, you will not be able to go back a recover any NON-ACTIVE files that may have been on the disk.

2.5 TAPE COPY OPERATIONS MANUAL:

Selection of this utility will result in the following four options be displayed to the screen.

1. Read Tape
2. Write Tape
3. Verify Tape
4. Copy Tape

Selection ?

2.5A READ TAPE:

This utility will READ a TAPE into memory, at which time, you may elect to EXAMINE OR MODIFY the bytes and write them back to tape. If you are using a Mod I, you will first be asked whether DECK 1 or DECK 2 is to be used. Input the DECK (recorder) number that contains the tape (1 or 2) that is to be read and press <ENTER>. At the bottom of the screen you will then be shown the bytes that are being read as they are received from the recorder. The bottom three lines contain the data as it is read from the tape. The top one is as ASCII representation of the bytes and the bottom two lines are the data in HEX numbers. When the routine is finished reading the data, you will be prompted to press <ENTER> and the TAPE data will be presented to you in standard ZAP format so that you may use the MODIFY MODE to make any changes. If changes are made, you may make another tape of the program that INCLUDES YOUR CHANGES by pressing <BREAK> after you have made the changes, and then using the WRITE TAPE OPTION. When using the MODIFY MODE to change MEMORY, all changes are IMMEDIATE and therefore the <U>pdate option that is used when changing a DISK is not necessary when using these options.

2.5B WRITE TAPE:

On the Mod I, this utility asks you the DECK (RECORDER) that you want to record to, and upon answering, you will be prompted to PRESS ENTER TO BEGIN. Upon Pressing <ENTER>, the MEMORY that was READ IN BY THE READ TAPE OPTION will be written back to the specified deck along with any changes that you made.

2.5C VERIFY TAPE:

This option asks you to input the DECK NUMBER (recorder) that holds the tape that is to be VERIFIED. After supplying the information and pressing <ENTER>, you will be given information on the LAST TAPE THAT WAS READ by the READ TAPE OPTION and also the BYTE COUNT and CHECKSUM of that tape. Press <ENTER> to proceed and the tape will be VERIFIED against the information that is contained in memory from the READ TAPE OPTION. If there are any differences, you will be given the number of errors that were encountered during the VERIFY.

2.5D COPY TAPE:

Upon choosing this utility, you are first asked which tape player you wish to use (1 or 2). Input the SOURCE recorder number and place the SOURCE tape (rewound to the beginning of the program) in the SOURCE recorder and press the PLAY button down. Next, unplug the larger of the two gray plugs (AUX) that goes into the SOURCE recorder. Next, unplug BOTH gray plugs from the DESTINATION recorder (AUX and MIC) and then insert the AUX plug from the Source recorder into the AUX jack of the DESTINATION recorder.

To begin copying, press down the PLAY and RECORD buttons on the DESTINATION recorder (it will immediately start running) and then press <ENTER>.

When active data from the source recorder starts pulsing into the computer, you will observe a large GRAPHICS BLOCK begin flashing in the upper right hand corner of the screen, and the copy is complete when the flashing stops. To stop the routine and return to the menu, press <CLEAR>.

This utility takes a data pulse from the source tape, "cleans it up", and then immediately rewrites it to the destination tape without loading the program into memory. the flashing stops. To stop the routine and return to the menu, press <CLEAR>.

2.7 MEMORY UTILITY OPERATIONS MANUAL:

Selection of this utility will result in the following fifteen options being displayed to the screen.

1. Display Memory
2. Move Memory
3. Exchange Memory
4. Compare Memory
5. Fill Memory
6. Reverse Memory
7. Test Memory
8. Jump to Memory
9. String Search
0. Input Byte from Port
A. Output Byte to Port
B. Memory to Sectors
C. Sectors to Memory
D. Memory to Track
E. Track to Memory

Selection ?

2.7A DISPLAY MEMORY:

Selection of this routine will require you to input the memory ADDRESS where you wish to BEGIN VIEWING. Be sure to append the letter H to your input if you are entering in HEX. Press <ENTER> and you will see the MEMORY ADDRESS and the FOLLOWING 255 BYTES on the screen. The screen format is in standard ZAP FORMAT and you may go into the MODIFY MODE and perform modifications, however, changes that are made while in this routine are IMMEDIATE and are NOT HELD IN A BUFFER. If you elect to enter into the MODIFY MODE (by pressing "M"), you will find that are a few COMMAND KEY differences between using the MODIFY MODE in this utility and in DISPLAY SECTORS if the ZAP UTILITY.

The following will a list consisting ONLY of the COMMAND KEY DIFFERENCES between this routine and DISPLAY SECTORS using the MODIFY MODE. The COMMAND KEYS that are NOT LISTED here and are contained in the MODIFY MODE COMMAND LIST and will also work in this routine.

1 - <UP ARROW>	Pages next higher 256 bytes
2 - <DOWN ARROW>	Pages next lower 256 bytes
3 - <SH><RIGHT ARROW>	Same as <UP ARROW>
4 - <SH><LEFT ARROW>	Same as <DOWN ARROW>
5 - <CLEAR>	Reselect new memory address
6 - <SH><UP ARROW>	Pages to highest memory
7 - <SH><DOWN ARROW>	Pages to lowest memory
8 - <RIGHT ARROW>	Rotates one byte of memory on the lower right hand section of screen and rotates one byte off upper left hand section
9 - <LEFT ARROW>	Rotates one byte of memory on the upper left hand section of screen and rotates one byte off the lower right hand side

While in the modify mode, if you attempt to move the cursor past the edges of the screen, additional data will be brought into view as needed.

2.7B MOVE MEMORY:

This routine asks you to input the STARTing and ENDing locations the memory to be MOVED, and also the STARTing location of where it is to be MOVED TO. EXAMPLE: To move memory from location 5000H to 6000H and move it so that it will reside STARTING at 8000H, would require the following response to the program requests of "START, END, START?". RESPONSE: "5000H,6000H,8000H" and then press <ENTER>. After the routine moves the selected memory to its new location, you will then also be advised of the number of bytes that were moved.

2.7C EXCHANGING MEMORY:

This utility requires that you input the STARTing and ENDing location of the bytes to be exchanged and also the STARTing address of the location of where you want the EXCHANGE to BEGIN. EXAMPLE: To EXCHANGE the memory locations that reside from 5000H to 6000H with the memory locations that BEGIN at 8000H, the following input would be required. RESPONSE: "5000H,6000H,8000H".

2.7D COMPARE MEMORY:

Selection of this utility requires you to input the STARTing and ENDing address of the MEMORY LOCATIONS that are to be COMPARED and then , the STARTing address of where you wish the compare to start. This routine will then advise you if it locates any differences between the two "BLOCKS OF DATA" and tells you the location of any BAD BYTES that it discovers. This utility requires the same type of data input as the the aforementioned 2 examples in sections 2.7B and 2.7C. NOTE: If several NON-COMPARES are discovered, they will scroll very rapidly on the screen and you may PAUSE them by holding down the <SPACE BAR>.

2.7E FILL MEMORY:

To FILL MEMORY with a user specified byte, select this option and you will then be requested to input the STARTing and ENDing point of WHERE you want memory to be filled and FILL byte that is to be used. EXAMPLE: To fill memory from A000H to B000H with the byte FFH, you would respond with the following input. RESPONSE: "A000H,B000H,FFH".

2.7F REVERSE MEMORY:

This routine asks you to input the START and ENDing points of the memory locations that you want REVERSED. Press <ENTER> and the entire BLOCK of data will immediately be put in reverse order from before and you will also be told the number of bytes that had been changed.

2.7G TEST MEMORY:

This utility tests every byte of memory between the inputed addresses by loading each byte with ØØH to FFH and then verifies each change.

To use this routine, input the STARTing and ENDing addresses and you will observe that the screen is constantly being updated with a display of the byte address being tested while it is in progress. The only RAM addresses that MAY NOT BE TESTED are displayed on the screen as they are the locations of the test routine itself and attempting to test in this area will cause a SYSTEM CRASH.

If you attempt to test ROM (Read Only Memory), you will get all ERRORS as ROM cannot be "written to" by this routine. Also, if any problems are detected in RAM ADDRESSES, you will be given a display of the BAD BYTE LOCATIONS and told BIT BY BIT where the problems are located.

2.7H JUMP TO MEMORY:

This program will allow you to JUMP TO any memory location and selection of it only requires that you input the ADDRESS of where you wish to transfer control, and then press <ENTER>.

2.7I STRING SEARCH:

Selection of this routine requires that you input the STARTing and ENDing points of the memory locations to be searched. Next, you are asked to supply the STRING that is to be searched for, and you will then be asked if you want to enter a REPLACEMENT STRING. If you want the string replaced with another one, simply enter the REPLACEMENT STRING and press <ENTER> at which time each occurance of the ORIGINAL STRING will be replaced with the REPLACEMENT STRING.

If you wish to leave the ORIGINAL STRING in its original form and merely want to locate it, then press <ENTER> by itself when you are requested to supply the REPLACEMENT STRING. Upon completion of the routine, you will be told the exact memory locations (if any) where any MATCHES exist and if there are several of them, you may pause the scrolling by holding down the <SPACE BAR>.

2.7J INPUT A BYTE FROM PORT:

Selection of this option will merely allow you to input and display a byte of data from any port. You only need to supply the PORT number from which you wish the input.

2.7K OUTPUT A BYTE TO PORT:

Selection of this option will allow you to output the byte of your choice to a port. You will be requested to input the PORT number where you want the byte sent and the BYTE that is to be sent to the port.

Two shorthand notations notations apply here. If you have a HIGH-RES board installed, you may turn it on with HY, and off with HN.

If you have a high speed clock installed, you may turn it on and off with SY and SN respectively.

2.7L MEMORY TO SECTORS:

This program requires input of the block start address that is to be DUMPED from memory to the disk. You must then specify if the sector to be written to is in IBM format or not. You will then be asked for the DRIVE, TRACK, SECTOR of where to BEGIN THE DUMP, and then the SECTOR COUNT (number of sectors to be saved).

Upon completing the data input, press <ENTER>, at which time the specified block of memory will be saved to the disk at your selected locations. NOTE: (Memory saved to the disk with this utility is not updated to the directory, and as a result, will be invisible without ZAP. Data stored to the disk with this utility may be loaded back to memory by using the SECTORS TO MEMORY UTILITY.)

2.7M SECTORS TO MEMORY:

First you must specify if the read is to be in IBM format or not. Then this option asks you to input the DRIVE, TRACK, and BEGINNING SECTOR. Next, the SECTOR COUNT (number of sectors that are to be loaded into memory). You are then requested to input the ADDRESS where to BEGIN storing the data (this is the FIRST memory location where the block will reside in memory). EXAMPLE: To load 1Ø sectors of data, from drive #Ø, track #14, sector #Ø, to RAM MEMORY, starting at location AØØØH, the following responses are required.

1 - RESPONSE to DRIVE, TRACK, SECTOR ....."Ø,14,Ø"
2 - RESPONSE to SECTOR COUNT ....."1Ø"
3 - RESPONSE to ADDRESS....."AØØØH"

2.7N MEMORY TO TRACK:

This utility asks for input of the beginning ADDRESS in memory where you wish to start the transfer of data to the disk. Next, you are asked to supply the DRIVE and TRACK number where the data is to be placed. Press <ENTER> and a track of data that resides FROM the beginning address will be written to the disk at the specified drive and track number.

2.7O TRACK TO MEMORY:

This utility will allow you to READ a COMPLETE TRACK of a disk, including GAPS, SYNC FIELDS, ADDRESS MARKS, and all DATA. To implement this routine, input the DRIVE, TRACK, and SECTOR NUMBER where you wish to begin examining. NOTE:(The SECTOR NUMBER is ignored). Next, you will be asked if you want it "SYNCHRONIZED" or not. If you want the controller to synchronize each address mark it finds, press <Y>es. If you want the controller to display the bytes "AS ASSEMBLED", press <N>o.

You will also be given the location of where the track data is presently located, and you may view the data by simply pressing <ENTER>.

2.8 FILE UTILITIES OPERATIONS MANUAL:

Selection of this utility will result in the following fourteen options being displayed on the screen.

1. Display File Sectors
2. Compare Files
3. Copy Files
4. Disk Directory
5. Free Space
6. Offset File
7. File Locations
8. Drive Status
9. Sector Allocation
Ø. Build File
A. Clear File
B. Disk Allocations
C. Compute Hash Code
D. Compute Passwords

Selection ?

2.8A DISPLAY FILE SECTORS:

Selection of this routine will require input of the FILENAME that is to be displayed (Don't forget DOS specifiers, drive numbers, and track counts if any) (TRSDOS Mod III users see footnote 14). Press <ENTER> and you will be given the END OF ALLOCATION (EOF) SECTOR NUMBER, END OF FILE (EOF) SECTOR NUMBER, and then requested to supply your CHOICE of which sector is to be viewed.

Standard operating system disks contain TRACKS on which data is stored (from 35 to 8Ø) and each track is broken down into GRANULES which are further broken down into SECTORS. On SINGLE DENSITY DISKS, each TRACK contains 2 GRANULES and each granule contains 5 SECTORS. On DOUBLE DENSITY DISKS, each TRACK contains 3 GRANULES and each granule contains 6 SECTORS. On TRSDOS Mod III, each track contains 6 granules of 3 sectors each. (* ND8Ø v2 users see footnote 13). You will notice that the EOF and EOA usually are not the same. The reason for this is that operating systems allocate space for data storage in GRANULES and so the EOF SECTOR may be located anywhere in the LAST ALLOCATED GRANULE.

To display the first sector in the file, press <ENTER> or you may input the sector number of your choice and <ENTER> (remember that Ø is the first RELATIVE sector). To automatically go to the EOF SECTOR, press "E" or to proceed immediately to the EOA SECTOR, press "A". Pressing <CLEAR> will allow you to RESELECT the sector of your CHOICE. You may also page thru the file by using the RIGHT or LEFT ARROW KEYS or UP and DOWN ARROWS (as they both perform the same function in this option) to go forward or backward thru the file one sector at a time. At any time you may enter into the MODIFY MODE by pressing "M" if you want to make corrections to the file.

You probably have already read the manual section on DISPLAY DISK SECTOR (2.1A) where the DISPLAY FIELDS are explained, but if you haven't, please do so PRIOR to continuing with this explanation.

The printout in this routine contains all of the information that is supplied in DISPLAY DISK SECTOR, but since we are now dealing with FILES, additional information is now available. To the left of RELATIVE BYTE #CØH, you see "FPDE" which means that the sector being viewed is in a FILE PRIMARY DIRECTORY ENTRY (don't worry about the "8" that is sitting right after "FPDE" as it will be explained later). If the letters had been "FXDE", it would mean the the sector is a FILE EXTENDED DIRECTORY ENTRY. To the left of RELATIVE BYTE #DØH, The letters "RSEC" appear and they stand for RELATIVE SECTOR. To the left of RELATIVE BYTE #EØH, a DECIMAL NUMBER is displayed and it tells you the RELATIVE SECTOR NUMBER of the file that you are viewing.

```
.   S00 C9E5 6069 2929 2929 4485 4FE1 2318 DB21 ..@I))))D.O.#..!
HEX Y10 D206 1100 4001 3600 EDB0 AF06 2712 1310 ....@.6.....'...
DRV S20 FCF3 C375 00CD A051 210F 43F3 2819 CBFE ...U...Q!.C.(...
 0  130 210F 403E C322 1643 3215 433E CD22 3140 !.@>."..C2.C>."1@
TRK /40 FB32 3040 C31F 4ECB BEAF 3215 433E 3E21 .20@..N...2.C>>!
 16 S50 A3EF 18E9 CDA0 5111 D94C 3E0E CA13 44C3 ......Q..L>...D.
TRU Y60 1044 237E FE20 28FA FE28 C023 CDDF 5078 .D#~. (..(.#..PX
 16 S70 E7C9 0119 B551 434D 440A 444F 5320 5245 .....QCMD.DOS RE
SEC  80 4144 590D 5748 4154 3F0D 434D 4402 0200 ADY.WHAT?.CMD...
 04  90 4E23 CD1F 5128 D818 CCCD 0451 18D1 0100 N#..Q(.....Q....
STD  A0 007E FE59 2810 FE4E 280F FE4F C023 7EFE .~.Y(..N(..O.#~.
ISD  B0 4628 06FE 4EC0 01FF FF01 B7FA 5023 7EFE F(..N.......P#~.
FPDE8C0 29C8 FE2C C818 F601 0000 7ED6 30D8 FE0A )..,.......~.0...
RSECED0 D0E5 6069 2929 0929 0600 4F09 444D E123 ..@I)).)..O.DM.#
0004 E0 18E8 0100 007E FE27 C023 7ED6 3038 0AFE .....~.'.#~.08..
+00 @F0 0A38 0ED6 07FE 1038 087E FE27 23C8 2BAF .8.....8.~.'#.+.
```

Lastly, there is one more COLUMN of information that needs to be explained and it is RIGHT NEXT TO, and IMMEDIATELY TO THE LEFT OF the RELATIVE BYTE NUMBERS. This COLUMN contains the FILENAME that is being viewed, and will also tell you if you are viewing a sector that is located BEFORE or AFTER the EOF SECTOR. The FIRST LETTER of the FILENAME itself sits IMMEDIATELY to the left of relative byte #ØØH, and the FOLLOWING LETTERS of the filename sit DIRECTLY UNDER IT. In the SAME COLUMN and to the left of RELATIVE BYTE FØH, you will see ONE OF THREE SYMBOLS. If the symbol is a "-", then you HAVE NOT reached the EOF SECTOR. If the symbol is a "+", then you HAVE ALREADY PASSED the EOF SECTOR. If the symbol is a "@", then you are viewing the EOF SECTOR and you will ALSO NOTICE that there is a HEX number IN this column and to the left of RELATIVE BYTES #DØH & #EØH (I told you we would get back to that 8 and the E). These numbers are representative of the FIRST AVAILABLE BYTE AFTER THE FILE ENDS (so the last byte in the file would be one byte LESS). If the numbers contained are "ØØ", then the actual file ended with RELATIVE BYTE #FFH in the PRECEDING SECTOR.

EXAMPLE: In the preceding example DUMP, the FILENAME is called "SYS1/SYS", the sector being viewed is a "FPDE", the "@" symbol is visible which tells you that the EOF SECTOR is presently being viewed, and the HEX NUMBERS "8" and "E" appear next to relative bytes CØH & DØH, which tells you that RELATIVE BYTE # "8EH" is the FIRST AVAILABLE BYTE in the sector and that # "8DH" was the LAST BYTE USED in the file.

2.8B COMPARE FILES:

Selection of this utility will require you to input the SOURCE FILENAME and then the COMPARE FILENAME. The routine will then compare the two programs BYTE BY BYTE and you will be told the location of any MISMATCHES that it finds. (NOTE: if you are doing a single drive compare, the DOS specifiers MUST BE THE SAME. Two drive users may compare files on different supported formats.

2.8C COPY FILES:

This utility asks you to input the DRIVE(S) number(s) from which you wish to COPY FILES, and upon doing so and pressing <ENTER>, you will be given a full screen directory of all valid files on the disk. In the LOWER LEFT HAND CORNER of the screen, you will see a FILENAME and IMMEDIATELY to its RIGHT a number will appear that tells you the number of GRANULES in that file. To the right of that, the word "COPY ?" appears, and you are then required to answer "Y" if you want that file copied or "N" if you don't want it copied. If you want to copy ALL the files, hold down the "Y" key and it will repeat. Each file on the disk will be displayed to you is this manner, and upon completion, you will be asked to input what DRIVE(S) NUMBER(S) where you want the files copied.

First all the data will be copied across the diskettes, then the directory of the destination disk will be updated all at one time. If you press <BREAK> at any time BEFORE the destination disk's directory is written back, then the diskette will essentially remained unchanged.

2.8D DISK DIRECTORY:

This utility, after input of the the DRIVE(S) NUMBER(S) to be displayed, will supply you with the NAME, DATE, NUMBER OF TRACKS, NUMBER OF FREE GRANULES, AND FREE FILES that are available on the disk. (* ND8Ø v2 and TRSDOS Mod III users see footnote 6). In addition, you will be shown every VALID FILENAME on the disk, including the PROTECTION LEVEL of each (if any), and whether each file is VISIBLE, INVISIBLE, or a SYSTEM FILE (* TRSDOS III users see footnote 7).. The method of advising you of the status of each file is in STANDARD DOS format (S=SYSTEM, I=INVISIBLE, P+NUMBER=ATTRIBUTE LEVEL). EXAMPLE: "SYS5/SYS SIP=7" indicates that the filename "SYS5/SYS" is a <S>ystem file that is <I>nvisible and has a <P>rotection level of "7".

2.8E FREE SPACE:

Selection of this routine automatically reads ALL DRIVES presently on the system, and you will be given the NAME, DATE, TRACKS, FREE GRANULES, and FREE FILES that are on EACH DRIVE. The DOS specifier must be set correctly on all mounted drives for the information to be located.

2.8F OFFSET A FILE:

This utility will allow you to LOAD A FILE into memory at the location of your choice or load a file into memory at one location and then have it move to ANOTHER LOCATION and EXECUTE, or you may even DISABLE INTERRUPTS. You will first be asked to supply the FILENAME of the file that is to be OFFSET and upon pressing <ENTER>, you will then be told the LOAD MODULE RANGE of the file (the PRESENT START, END, AND ENTRY POINT of the file). Next, you will be asked to supply the NEW LOAD ADDRESS (where you WANT it to reside), and the LOW OFFSET (the LOWEST memory location that may be offset ** default=4ØØØH **). Next, you will be asked of you want to ADD APPENDAGE (the appendage is 14 bytes long and it performs a BLOCK MOVE of a file from one location in memory to another ** answer "Y" or "N" **). You will then be told the NEW MODULE LOAD RANGE and asked if you wish to DISABLE INTERRUPTS (answer "Y" or "N"). Press <ENTER> and the file will be written back to the disk and from this point on, it will load into memory at your specified locations.

2.8G FILE LOCATIONS:

This routine will give you complete information on the location of files on a disk and it only requires you to input the DRIVE(S) NUMBER(S) that are to be examined. Press <ENTER> and you will given the NAME of each valid file on the disk, whether the file is a SYSTEM file, INVISIBLE, PROTECTED, the location of the File Primary Directory Entry (FPDE), and File Extension Directory Entries (FXDE) if any exist. In addition, you will be told the TRACK and SECTOR NUMBERS where the FPDE resides, PLUS the RELATIVE BYTE NUMBER and DIRECTORY ENTRY CODE (DEC) of the FPDE. Also you are given the End Of File Sector (EOFS), End Of File Byte (EOFB), Logical Record Length (LRL), Encoded Update Password (EUPD), Encoded Access Password (EACC), Number of GRANS in the file, and the location of EXTENTS.

The following will be a LINE BY LINE explanation of each of the information outputs from this utility (see sample printout). The different outputs will be discussed starting from the LEFT of each line and then proceeding to the RIGHT.

LINE 1 - This contains the FILENAME that is being displayed and information on whether the file is <S>YSTEM, <I>NVISIBLE, and the <P>rotection level of the file.

LINE 2 - The first part of the line gives you the LOCATION of the FPDE and tells you the DRIVE NUMBER where it is located. Next, the TRACK and SECTOR NUMBERS where the FPDE resides (directory), the location of the relative BYTE NUMBER of the entry, and the DEC NUMBER. The DEC NUMBER is the RELATIVE BYTE NUMBER LOCATION of the HIT BYTE in the Hash Index Table (HIT) of the file. EXAMPLE: If the DEC NUMBER is 60H, you could display the HIT sector (second sector of the directory), and at RELATIVE BYTE 60H you would see the HIT BYTE NUMBER of the file. The DEC also may be used to figure out the location of the SECTOR NUMBER where the FPDE resides in the following manner. The LAST NUMBER in the DEC PLUS 2 will tell you the sector number on the directory where the FPDE is located. To figure out the RELATIVE BYTE LOCATION of the FPDE on the sector, change the LAST NUMBER to ZERO. EXAMPLE: If the DEC=62H, the FPDE of the file would be on sector 4 (2 PLUS 2) of the directory and it would entered starting at RELATIVE BYTE 60H (change the last number to 0).

LINE 3 - The End Of File Sector (EOFS) will contain the number of the FIRST AVAILABLE SECTOR AFTER THE FILE if the End Of File Byte (EOFB) is 00H. If the EOFB IS NOT 00H, the EOFS will contain a number that is actually 1 less than true and you will have to add 1 to it. NOTE: Don't blame us for this as this is the way TRSDOS was written. The EOFB is simply the FIRST AVAILABLE BYTE AFTER THE FILE. The Logical Record Length (LRL) is figured by multiplying the number by 256. EXAMPLE: If the LRL = 0, then the logical sector contains 256 bytes. If the LRL = 1, the logical sector contains 512 bytes.

LINE 4 - The Encoded Update Password (EUPD) and Encoded Access Password (EACC) hash codes are both printed out in LSB-MSB format. The GRANS NUMBER tells you how many GRANULES are in the entry

. LINE 5 - The EXTENTS tell you how many extents are in the FPDE and the location of the data on the disk. Each EXTENT contains 3 numeric values that tell you the STARTING TRACK number where the data resides, the STARTING SECTOR number, and the NUMBER OF GRANULES in the extent. On SINGLE DENSITY DISKS, each track contains 2 granules and each granule contains 5 sectors. On DOUBLE DENSITY DISKS, each track has 3 granules and

each granule contains 6 sectors. On TRSDOS Mod III, each track has 6 granules of 3 sectors each. EXAMPLE: If line 5 reads "05,0,06", then the data for this EXTENT starts on TRACK #5, SECTOR #0, and would contain a total of 6 GRANULES of data. The TRACK AND GRANULE NUMBERS may be read directly for both single and double density disks, but the SECTOR NUMBER is a little different. On SINGLE DENSITY DISKS, if the sector number is "0", then the data will start on sector "0". If the sector number is "1", then the data will start on sector #5. DOUBLE DENSITY DISKS, the sector number may contain the numbers "0", "1", or "2", (double density disks have 3 granules per track) and these indicate the data starts on sectors "0", "6", or "12" respectively. Mod III TRSDOS can be "1" to "5" representing sectors 1, 4, 7, 10, 13, 16 respectively.

LINE 6 - This line will not normally be displayed unless you have a very large file that requires more than four extents. On the following page is a sample printout of a file with several EXTENSIONS. (* Mod III TRSDOS see footnote 15). When a file with extensions is encountered, line 6 will read as FXDE DEC = ##. Following this, you will see that each extension will be listed in numeric order and will contain the necessary information to determine the location of the file data for each extension. The extension printouts are in the same format as we have been discussing, but WILL NOT contain the information on PASSWORDS, EOFS, or EOFB as this information was already given to you with the FPDE for the file. Also you will notice the the DEC NUMBER in line 6 will be the same as the DEC NUMBER of the FIRST EXTENSION, thereby letting you maintain the BACKWARD CHAIN of the program data.

```
COPY/CMD                IP=6
FPDE - :0, TRACK = 17, SECTOR = 03, BYTE = C0H, DEC = C1H.
EOFS = 00004, EOFB = 4AH, LRL =  00.
EUPD = 8130H, EACC = 9642H, GRANS =  01.
EXTENTS /  06,1, 01 / EOF


SYS0/SYS                SIP=7
FPDE - :0, TRACK = 17, SECTOR = 04, BYTE = 00H, DEC = 02H.
EOFS = 00013, EOFB = 48H, LRL =  00.
EUPD = EB29H, EACC = 210EH, GRANS =  03.
EXTENTS /  00,1, 03 / EOF
```

2.8H DRIVE STATUS:
Selection of this option automatically checks the status of all drives in the system and tells you the individual status of each. If a drive is software disabled, it will be displayed as NOT IN SYSTEM.

2.8I SECTOR ALLOCATION:
This utility will tell you to which file, any sector on the disk has been allocated. Input the DRIVE, TRACK, and SECTOR number that you wish to check and press <ENTER>. You will now be given the NAME of the file that it is assigned to, along with the RELATIVE SECTOR NUMBER that it is in the file. In addition, you will be told the End Of File Sector number (EOFS), End Of File Byte number (EOFB), and the Logical Record Length (LRL) of the file. (* Mod III TRSDOS users see footnote 18).

2.8J BUILD A FILE:
Selection of this option will allow you to PRE-ALLOCATE space on a disk for a file so that when the file is written to the disk at a later time, it will be written in a contiguous manner. You will be asked to supply a FILENAME and after doing so, will be advised of the disks NAME, DATE, NUMBER OF TRACKS, FREE GRANULES AND FILES THAT ARE AVAILABLE. Next, you are asked to input the NUMBER OF GRANULES THAT ARE TO BE ALLOCATED for the file. Press <ENTER> and the file information will be written to the disk and the directory will be updated. (* TRSDOS Mod III users see footnote 16).

2.8K CLEAR A FILE:
This option asks you to input the FILENAME that is to be CLEARED from the disk. Press <ENTER> and you will then be asked "Are you SURE you want to clear it ?". Answer "Y" or "N" and press <ENTER>. When you use this option to clear a file, all of the files data is ERASED from the disk and there is NO CHANCE OF DATA RECOVERY.

2.8L DISK ALLOCATIONS:
This routine requires that you input the DRIVE(S) NUMBER(S). Press <ENTER> and you will observe a full screen display of the ENTIRE DISK ALLOCATIONS. The screen will show the numbers Ø to 95, and each represents a track. To the RIGHT of each number you will see that there is a possibility of THREE ENTRIES for each track (3 for Double Density Disks, and 6 for TRSDOS Mod III). Each entry will be either a PERIOD, GRAPHIC BLOCK, an *, or X and each one represents 1 GRANULE. If the disk being examined is DOUBLE DENSITY, then each of the GRANULE REPRESENTATIONS are valid, but if the disk is SINGLE DENSITY, disregard the RIGHT MOST ENTRY for each TRACK. The STATUS of each GRANULE may be obtained by noting whether the GRANULES are ALLOCATED or not. If a PERIOD is displayed, the granule has already been ALLOCATED. If a GRAPHIC BLOCK is displayed, the granule is STILL AVAILABLE. If an "X" appears, the track is LOCKED OUT and not usable. If an "*" appears, then that track is beyond the physical boundaries of the diskette.

EXAMPLE: On the following sample printout of a SINGLE DENSITY DISK, you will notice that the LEFT MOST granule entry for track contains a period as that granule is ALLOCATED for the SYSTEM BOOT. The granule that is to its RIGHT appears as a GRAPHIC BLOCK and it has NOT been allocated by the system. On all tracks higher than 39 (this printout is a 40 track disk and contains tracks Ø thru 39), you see that each granule entry contains an "*" which tells you that all over these tracks are locked out and not usable.

The reason that we display the 3 GRANULE format to the screen for both SINGLE and DOUBLE DENSITY DISKS is to remain consistant AND it allows us to save a little more memory for other things. This program is OVER 32K long and we have to conserve memory whenever possible by using SCREEN DISPLAY ROUTINES that may be used for more than one subroutine.

2.8M COMPUTE HASH CODE:
This routine will compute the HASH CODE of ANY VALID FILENAME. You will be asked to supply the FILENAME for which you want the HASH CODE, and upon doing so, the program will COMPUTE AND DISPLAY it to the screen.

2.8N COMPUTE PASSWORDS:
Upon choosing this option, you will be asked if you wish to <E>ncode or <D>ecode a password to which you answer <E> or <D>, and press <ENTER>. If you pressed <E>ncode, you will then be required to input the PASSWORD that is to be <E>NCODED, and upon doing so and pressing <ENTER>, the HASH CODE for the PASSWORD will be COMPUTED AND DISPLAYED (* TRSDOS III users see footnote 17).

If you select <D>ecode, you will be prompted to input a FILENAME and upon completion and pressing <ENTER>, it will read the FILENAME from the disk and then COMPUTE a USABLE UPDATE and ACCESS PASSWORD. You may press the <CLEAR> key to halt the password processing at any time.

2.9 DISK CONFIGURATION OPERATIONS MANUAL:

2.9A SYSTEM INFORMATION:

The following will be STEP BY STEP INSTRUCTIONS for persons wishing to CUSTOM CONFIGURE this program to their individual systems. Before I start explaining this procedure, there are a few things that you may have to know in order to insure proper operation of the program.

SUPER-UTILITY PLUS will allow you to configure up to 4 drives, but certain systems require special considerations. Several of the routines in SUPER UTILITY PLUS will TEMPORARILY OVERRIDE the configuration in order to operate and then RETURN back to the ORIGINAL configuration when completed. This AUTOMATIC SWITCHING back has been implemented to allow for much easier operation by the user. So if you happen to look at the CONFIGURE SYSTEM option, it could show something different than you originally set up, but don't worry, it will RETURN to your configuration by itself.

Due to the vast differences of double headed methods that the DOS's are using, SU+'s support is limited to their single sided counterparts.

There are three different ways to CONFIGURE SU+. The first way, by TEMPORARY OVERRIDES, is discussed in the GENERAL INFORMATION section of this manual and will not be discussed in this section. The other two ways are called SOFT CONFIGURING and HARD CONFIGURING.

2.9B SOFT CONFIGURE:

The user may "SOFT CONFIGURE" this program upon initialization or at any time when the program is resident in memory. The SOFT CONFIGURE is only resident for the remainder of time that the program is in memory and if you use this method, you will have to CONFIGURE the system each time you LOAD the program. Select the CONFIGURE SYSTEM option and press <ENTER> at which time you will observe the following screen display appear.

```
   ** CONFIGURATION **                                          #

 =>HISPEED=N, DUAL=N, SAVE=N, ON=3E01D3FE00000, OFF=3E00D3FE0000
   PRINTER GRAPHICS=N, LOCASE=N, MX80=N, PARALLEL=Y, LFEEDS=N

   +:0A= 40, TKS= 40, DIR= 17, STEP=3, DELAY=4/4, WP=N
             DENSITY=S, L-GRANS, TRACK0=S, SS=0, DAM=0
   +:1A= 35, TKS= 35, DIR= 17, STEP=3, DELAY=4/4, WP=N
             DENSITY=S, L-GRANS, TRACK0=S, SS=0, DAM=0
   +:2A= 35, TKS= 35, DIR= 17, STEP=3, DELAY=4/4, WP=N
             DENSITY=S, L-GRANS, TRACK0=S, SS=0, DAM=0
   +:3A= 35, TKS= 35, DIR= 17, STEP=3, DELAY=4/4, WP=N
             DENSITY=S, L-GRANS, TRACK0=S, SS=0, DAM=0

 ? ..........................................................
```


The sixth line of the information display tells you the information for DRIVE :0. Reading the default information across the line from left to right:

+ indicates that the drive is software in the system
  (a - indicates software NOT IN SYSTEM)
:0 indicates the information is for drive 0
A indicates it is currently configured for single density TRSDOS
= 35 indicates the REAL track count for the diskette.
  (* ND80 v2 users see footnote 6)
Tks=35 indicates the RELATIVE track counts for the diskette.
  (this will ONLY be different that REAL tracks when
  specifying ND80 v2 or DOUBLEDOS)
Dir=17 indicates the directory track of the LAST directory
  that was read or written (this changes)
Step=3 indicates the step rate for the specified drive.
  (0=6ms, 1=12ms, 2=20ms, 3=40ms)
Delay=4/4 indicates the time to wait in quarter seconds
  for the drive to come up to speed when it is
  first turned on.
WP=N indicates that the disk is not currently software
  write protected. No disk will be written to with
  this option set to Y.

The information that appears along the second line for each drive is implied information, and cannot be directly changed. Reading from left to right it indicates the following information:

Density=S indicates Single density (else if would be D)
L-Grans indicates LOGICAL grans, that is to say, that REAL tracks
  and RELATIVE tracks are the same. If you are using ND80 v2 Double
  density, or DOUBLEDOS, then you will be using I grans.
Track0=S indicates that track 0 is single density also.
SS=0 indicates that the starting sector for each track is 0.
  (this will only be 1 with Mod III TRSDOS)
DAM=0 indicates the current data address mark that is to be used.
  (this will only be 1 with LDOS)

You may individually change the specifications on any of the drives or perform a wide range of other functions by answering the questions that are displayed on the screen. There will be an arrow along the left side of the screen indicating which drive is currently being operated on.

Across the top data line of the screen you will see 3 yes/no questions, and 2 byte lists. When the arrow is on this line, answer with a Y or N to the corresponding position. If you specify that the high speed clock is to be ON, you may then also specify the sequence of commands that are to be used to activate/deactivate it. Each command may be up to 6 bytes long. The default settings for:

On =3E01D3FE0000 (LD A,1 / OUT FEH,A / NOP / NOP
Off=3E00D3FE0000 (LD A,0 / OUT FEH,A / NOP / NOP

DUAL mode ON will send all bytes that go to the video ALSO to the printer. This will be useful for unattended operation. If you want the program to REMEMBER the configuration during the whole time it is in memory, press "Y" to the SAVE question. If you are planning to examine

several different types of DISK OPERATING SYSTEMS, and intend to use the OVERRIDE COMMANDS, then answer N.

If SAVE = NO, then each time you enter an override command by appending a drive number with a DOS SPECIFIER or a TRACK COUNT, it will be inserted permanently into the tables. The next time you reference the drive, the setting will be the same as it was set prior. Therefore, you will need to enter a DOS type ONLY when you change its setting. Setting it from the configuration menu, or by appending the dos letter to a drive somewhere else will act identically.

If SAVE = YES, then you will only be able to set the default settings from the configuration menu. If you use override commands by appending a dos specifier, it will only be used for the routine in progress. As soon as it is completed, or the <BREAK> key is pressed, it will revert BACK to its setting from configuration menu. In this case, there is a difference between the override commands and the configuration menu.

In general, if you use the same operating system MOST of the time, then turn SAVE ON. This way you will only have to add the dos specifier and track count when it is DIFFERENT from what you have configured. However, you must enter it EVERY time, as the override commands are forgotten. If you change operation on operating systems frequently, it is best to turn SAVE OFF, as the override commands are remembered.

You will then be given an opportunity to INDIVIDUALLY specify the parameters for EACH DISK, one at a time, in your system. When entering the track count information, you will only enter the number following the = sign, the one marked Tks=xxx will be implied from the Dos specifier (* ND8Ø v2 and DOUBLEDOS users see footnote 6).

After inputting the specifications for each drive, you will be asked information about your printer. If your printer is capable of printing GRAPHICS (answer "Y" or "N"), if you have LOWER CASE capabilities ("Y" or "N"), if your printer is a MX-8Ø and requires that the graphic bytes are ADJUSTED ("Y" or "N"), if your printer requires LINEFEEDS ("Y" or "N"), and lastly, if your printer is P>arallel or S>erial ("P" or "S").

2.9C HARD CONFIGURE GENERAL:

HARD CONFIGURING involves ZAPPING the disk and once you have completed this process, the program will be set to the SELECTED specifications each time it is loaded. You may still examine other types of systems simply by using the OVERRIDE COMMANDS, or the program may be RE-CONFIGURED by ZAPPING the disk again for DIFFERENT PARAMETERS, however, it is STRONGLY SUGGESTED!!!!!!!! that you read the section on the MODIFY MODE and familiarize yourself with it prior to attempting ANY zapping on the SU+ disk.

2.9D HARD CONFIGURE TRACK 1, SECTOR #1

To HARD CONFIGURE the program, expose the disk notch, place SU+ in drive #Ø, and then perform the following steps.

1. - Select option #1 (DISK ZAP) and press <ENTER>.
2. - In response to the "DRIVE, TRACK, SECTOR ?" question,

input "ØA,1,1", and press <ENTER>. You will now see the sector displayed.

3. - Look to the left of relative byte #1ØH and make sure that the word "ASC" is there as all modifications on THIS sector are inputted in ASCII. If any other word is there, press "A" and you should see the word ASC appear.
4. - Press the "M" key to enter the MODIFY MODE and you will observe that there are now 2 cursors on the screen. One is on the HEX (left) side and the other is on the ASCII (right) side.
5. - Enter G4Ø to move the cursor to relative byte 4ØH.
6. - Enter the information exactly as you would enter a soft configure command line. The sequence is as follows: Active/Drive #/Dos specifier/track count/directory track step speed/motor on delay/software write protect
7. - Repeat for each of the 4 drives.
8. - When you have CONFIGURED each drive in your system, press <ENTER> and then select the <U>pdate option. The changes will then be written back to the disk.

2.9E HARD CONFIGURE SECTOR #2:

After completing the afore described zapping procedure for SECTOR #1, press the <RIGHT ARROW> one time. You will see additional soft configure information displayed on the screen. Enter ASCII modify and make the changes accordingly. The last question on the screen is the default base. This will be used for any numerical input as the default value. If you like to work mostly in HEX, you may enter 16 in this location. This way, all numerical input will be interpreted in HEX as the default condition. Remember, if you change this, that you must specify D for decimal numbers.

After completing the questions, press <ENTER> and select Update from the choices. Advance to the next sector.

2.9F HARD CONFIGURE SECTOR #3

Here is the remainder of the soft configure information. These are questions involving the operation of your printer, and must be answered with Yes or No in ASCII modify.

2.9G HARD CONFIGURE SECTOR #4

This sector contains information that will be used as the default values in the format utility. The NAME and DATE will be used when you press <ENTER> in response to those questions, and will also be placed in the GAT sector if you do a REPAIR GAT and choose the ENTIRE SECTOR.

This sector also contains the pattern of data that is to be written to the diskette during the formatting process. There is a single and double density pattern, each 16 bytes long. This pattern will be duplicated 16 times (total 256 bytes) in each sector. The default settings are the worst case patterns for each density (E5H for single density, and 6DB6H for double density). You will have a maximum amount of diskette failures using these patterns. This will uncover possible problems during the format

process instead of waiting till you have data on the sectors.

You may, of course, place anything into these strings that you wish. Put your name here, and it will be placed in all the empty sectors during format.

2.9H HARD CONFIGURE SECTOR #5

This sector will allow you to send bytes to ports where you may have special hardware connected. It can turn on your high speed clock, or inverse video, or whatever you have. On the Mod I, SU+ will not send ANY bytes to ANY ports, unless you specify. On the Mod III, only ports affecting the NMI vectors are used.

2.9I PATCH SECTORS:

Sectors 6 thru 9 on track #1 have been left open to allow you to insert another program of your choice WITHIN this program. If you have a special printer driver or other utility program that you want resident at the same time, then you may use these sectors. These sectors will allow TRSDOS STANDARD LOAD FILE FORMAT and the following four code numbers apply.

1. - Ø = TERMINATOR.
2. - 1 = LOAD MARKER.
3. - 2 = ENTRY POINT.
4. - 3-1FH = REMARK.

Additionally, all memory above SU+ may be used for other programs,and you may even use any of the sub-routines in SU+ in YOUR PROGRAMS. SU+ will not require any memory above the label @PGMEND except for buffer areas. If you want to locate and use the subroutines in this program, then you may find the NAMES AND ENTRY POINTS of each by holding down the <RIGHT ARROW> key when SU+ is being loaded, at which time, they will be displayed to the screen. If a HARD COPY is desired, press the <RIGHT> AND <LEFT><ARROWS> together when it is loading and the printer will give you a full copy.

After the program is loaded, you won't be able to get this information, as many changes are made in the program during initialization, but you may RE-BOOT the system at any time to obtain it.

2.9J HARD CONFIGURE TRACK 2, SECTOR 1

On Track 2, Sector 1, is a Model I BOOT that is compatable with TRSDOS, and NEWDOS 2.1. In fact, it will load in ANY program that is in load file format, starts on track Ø, sector 5, and is contiguous. Use REPAIR BOOT to speed up your TRSDOS. If you use other operating systems, you may use COPY DISK SECTORS option from the ZAP menu to copy the boot sector (Track Ø, Sector Ø) to this sector. It will then be used in the repair boot option of the repair utility.

On Track 2, Sector 3, is a Model III BOOT that is compatable with TRSDOS. It will load a program that starts on Track Ø, Sector 2. This boot may also be modified similar to the above sector 1.

3.Ø ENHANCEMENTS AND INCOMPATABILITIES:

3.1 SUPER-UTILITY PLUS ENHANCEMENT DISTRIBUTION:

Breeze/QSD, Inc. will notify all registered owners of SU+, of any enhancements or updates pertaining to this program as they become available. Registered owners will also be placed on our mailing list, so as to be kept informed of any NEW programs as soon as they go on the market.

3.2 REPORTING OF INCAPATABILITIES:

SU+, to the best of our knowledge, contains no "BUGS", but with a program of this size (32K of machine language) it is possible that some minor bugs may be found.

If you find such a problem, please RE-READ the MANUAL concerning that section, and if you still are not able to overcome the problem, send a thorough description of the problem along with the disk that you are having the problem with to:

Breeze/QSD, Inc.
11500 Stemmons Fwy
Suite 125
Dallas, Texas 75229.

We will do our best to assist and support all REGISTERED OWNERS, so be sure and include your SERIAL and PHONE number. Upon receiving your disk, we will attempt to locate and isolate the problem (within reason) to your satisfaction, however, we have found that in MOST cases where problems arose, the user had either NOT READ the section in the manual concerning the problem or had GLOSSED OVER it and had missed the explanation.

*** FOOTNOTES ***

***** 1. LAST is one of the powerful features in Super-Utility Plus. If you are viewing a sector, and accidentally press <BREAK>, you may enter L for the TRACK, SECTOR prompt to return to the last one viewed. Several other routines will also update the LAST sector. When comparing sectors, the last sector that mismatched will update LAST. When performing a string search, the last sector to contain a match will also update LAST. If you are viewing, for example, drive Ø, track 8, sector 3, and wanted to view the same sector on drive 1, you may press <CLEAR>, and enter "1L" for DRIVE, TRACK, SECTOR, and presto, you are right there.

Other routines also have a LAST feature. When in the modify mode, and you use the G command to position the cursor to a relative byte, it also updates its individual LAST. Thus in combination with the above use of LAST, you may examine a specific relative byte on separate drives with a minimum of keystrokes.

When using the L command in the modify mode, press LL to locate the next occurance of the LAST byte looked for. This allows for speedy positioning of the cursor.

***** 2. When several items of data are requested to be entered on the same input line, you may terminate at any item prior to the end, and all entries past the termination will be set to default values. For example, at the "Drive, Track, Sector ?" input, the defaults are Ø,Ø,x (where x is the lowest sector on the track, Ø for everything except TRSDOS III, which would be 1). You may enter the drive only and default to the first sector on that drive. If drive Ø is to be used, the Ø may be omitted if a dos specifier is used. Thus "ad" in response to the above prompt would result in Drive Ø, set for TRSDOS I single density, the directory track, first sector.

***** 3. Normally, the track that you specify to view will be the same as the REAL track on the diskette. ND8Ø v2 and DoubleDOS use a rather unique method of storing information on a diskette. They use a "disk relative sector" technique, and GRANS may span tracks. In order for SU+ to operate on these diskettes, it is necessary to perform the same computation of the "disk relative sector" as they do. Therefore, if you have specified ND8Ø or DoubleDOS, the sector you ask to view, and the one you really are viewing WILL BE DIFFERENT. For example, if you ask to see the directory, which is usually on track 17, sector Ø, you will actually view track 1Ø, sector 8!. If you have specified double density with a single density track Ø, you will never be able to view the REAL track Ø! You will note that the TRUE track is track 1!

As a general rule, if you are working on a ND8Ø or DoubleDOS double density disk, and are NOT working with files or directories (backup, display disk sector, verify sectors, etc.), then DO NOT use the H, I, or J specifiers. Use F, F, or G symbols respectively. This way, you will deal with the REAL sectors on the disk.

***** 4. The floppy disk controller that is used in the Mod I, is capable of reading/writing 4 different DATA ADDRESS MARKS. They are: STANDARD, READ PROTECTED, DELETED DATA, and USER DEFINED. The double density controller, however, is capable of reading/writing ONLY the DELETED DATA, and USER DEFINED marks.

On the Mod I's with double density, TWO controllers are present, and the entire compliment of marks can be addressed. On the Mod III, a single, different disk controller performs both tasks of single and double density. It can only recognize the DELETED DATA and USER DEFINED marks EVEN IN THE SINGLE DENSITY OPERATION.

***** 5. In order to be able to read in extended directories that were created with ND8Ø, SU+ will continue to read sectors until it reaches the first NON-READ-PROTECTED sector. You will note that there is no ND8Ø single density specifier. This is because of the direct compatability between it and TRSDOS. If you have specified a DOS that does NOT have extended directories, only the pertinent sectors will be read. If you have specified a DOS that MAY have an extended directory (A,I,J) the number of directory sectors PLUS ONE will be read into memory. This is normal!

***** 6. The track count of a diskette can, in most cases, be interpreted directly from information in the GAT sector of the directory. This is NOT true, however, with TRSDOS Mod III, and ND8Ø. With all other specifiers, if the directory is to be read in, the only thing that MUST apply is that the CONFIGURED TRACK COUNT must be at least 1 higher that the directory track. Thus if you have an 8Ø track diskette with the directory on track 4Ø, and you ask for a directory but are only configured for 35 tracks, you will get unpredictable results because the directory is beyond the boundaries of the diskette.

When using TRSDOS III and ND8Ø, you MUST specify the EXACT track count of the diskette for correct operation. Follow the same proceedure that you do with ND8Ø when using a double density track Ø. That is, configure for the real track count MINUS ONE. If you have a 4Ø track double density disk, with track Ø as double density, you MUST specify 39 for the track count. Otherwise SU+ may complain about non-existant errors, especially with directory checking.

***** 7. TRSDOS III DOES NOT log the system files in the directory in the normal manner. Therefore, they will not appear as part of a normal directory listing.

***** 8. When using TRSDOS III, you will not be able to use the "S" option in kill by category to remove the system files. Use the "remove system files" option to safely remove them.

***** 9. TRSDOS III logs the system files in the last 32 bytes of the HIT table of the directory. Because of this, when you use the REPAIR HIT on these diskettes, the last 32 bytes will not be disturbed. If these bytes are corrupt also, you must correct them using the ZAP feature. If the bytes are obviously invalid, insert FFH in all 32 spots.

***** 10. When using the REPAIR BOOT sector routine on TRSDOS III, only BootB will be written to sector 1 of the diskette because there is no sector 0.

***** 11. Due to the differences in the READ PROTECT status in the Mod's I and III (see note 4), you will get different results in both machines. If you would like to view a directory on a TRSDOS Mod I formatted disk in the III, you must first "read protect" the directory, as the Mod I marks are not discernable in the III. This will actually write USER DEFINED address marks on the disk, as this is "read protected" in the Mod III. When you transfer the diskettes back to the Mod I, the directory will be unreadable there unless it is correctly read protected.

Therefore, be sure you perform the "read protect" on the correct machine. This will NOT apply to LDOS formatted diskettes, as they use the correct mark that is common to BOTH the Mod's I and III.

***** 12. A DoubleDOS directory CANNOT be moved. Normally the 2'nd or 3'rd byte on a diskette will indicate the track where the directory is located. This is not true for DoubleDOS, and if you do move it, DoubleDOS will not be able to find it!

***** 13. Normally, a GRAN on a diskette is an even subdivision of a track. Thus on a single density disk, it is possible to have two 5 sector grans, or five 2 sector grans. On double density with 18 sectors/track, it is possible to have two 9 sector grans, three 6 sector grans, or six 3 sector grans.

ND80, however, has decided NOT to apply this standard to their system. They have given us a new term, "lumps"?. In order for SU+ to work on ND80 diskettes, the "lump" (GRAN) MUST be one of the above NORMAL conventions. Thus when using ND80, your "lumps" MUST be 5 sectors. (also, the sectors must begin numbering with 0 UNLESS you have configured for TRSDOS III compatability, in which case you would use that specifier).

***** 14. In order to provide SOME support for the Mod III TRSDOS system files, the ability to display their sectors has been provided. This will ONLY work with the display file sectors option in the file utilities.

You may enter the filename starting with an *, and followed by a 2 digit decimal system number. Thus
"Filename ? *05:1a" is valid for SYS5. If the system is INACTIVE, a "file not found" message will be issued. This will ONLY work for TRSDOS Mod III, and ONLY in "display file sectors". You may optionally BUILD a file (see note 16) out of a system file, and then operate on it normally.

***** 15. Mod III TRSDOS CANNOT have extended file directory entries. There are, however, 13 extent elements per file versus 5 for the Mod I TRSDOS.

***** 16. It is possible to log a Mod III TRSDOS system file into the directory. Use the build file option to create a file with 0 grans allocated. Now use the ZAP utility to examine sector 2 of the directory. Starting at byte E0H for 16 bytes, are the system allocations. Each system takes 2 bytes. Thus SYS0 would be bytes E0 and E1, SYS2 would be E4 and E5. When you have the file created, place the first byte into the directory entry relative byte 27H, and the second at relative byte 28H. You you now be able to perform NORMAL file routines. You may copy it to another disk, etc. If you KILL this file though, be sure to MANUALLY put FFFFH in the appropriate slot in the HIT table, if you want to really kill the system. Otherwise KILL will just kill that directory entry.

***** 17. Password encoding is different in TRSDOS Mod III than it is in ALL other systems. Others have remained consistent with the Mod I TRSDOS scheme, but R/S has decided to come up with something new. Therefore, if you use the password ENCODE routine, it will use the method of the LAST DOS SPECIFIER used. Thus if the last thing you did was read a Mod III TRSDOS diskette, then that method will be used.

If you never use TRSDOS III, then you won't have this problem. You can, however, indirectly set the method of your choice (as long as SAVE CONFIG is OFF). From the file utility menu, select "display file sectors". Answer the "filename" question with A:0B, then press <ENTER> followed immediately with <BREAK>. This will set TRSDOS III password encoding. If you want the other scheme, substitue A for the B in the above example.

***** 18. In the "sector allocation" option, if you specify a sector that belongs to a Mod III TRSDOS system file, it will display that the sector is not assigned to any files. This is because their system files are not logged into the directory in the normal manner.

***** 19. When using Mod III TRSDOS, the only way to automatically remove system files is by using the "remove system files" option from the purge utility. If you know what you are doing, you may set the appropriate system bytes in the HIT table, then "repair GAT" table using SU+.

***** 20. When formatting a diskette, an attempt is made at reading the first sector to see if it contains any data. If this is successful, then an attempt is made at reading the directory (provided the ! option is not used from "use configuration"). Since ALL operating systems EXCEPT TRSDOS Mod III start the tracks with sector 0 (TRSIII uses sector 1), you will not get a message that the diskette has data IF it is formatted for TRS III, and you are now formatting it for something different.

***** 21. When SU+ is loading itself into memory, a thorough memory test is done on your machine to insure the integrety of all 48K of available RAM. On the Mod I running at the normal 1.7 meg, the disk drive will turn off during this test (it only stays on for 3 seconds of no access), and then turn back on to read the program into memory. This is normal. The Mod III uses a faster clock speed, and the 3 seconds will not lapse to turn off the disk motor. If you have a high speed clock installed in your Mod I, there should be no motor off phenomenon.

NOTES

NOTES