SUPER UTILITY PLUS VERSION 3.0

by Kim Watt

User's Manual by R. B. Reyes

This product is made available on an AS-IS basis only, and Breeze/QSD shall not be liable for any loss or damages, whether real, alleged or imagined, arising from the use of this software. Breeze/QSD makes no warranties of merchantability or fitness for any particular purpose. Determination of suitability is the sole responsibility of the end user.

A PowerSOFT product from

Breeze/QSD Incorporated
11500 Stemmons Expressway, Suite 125
Dallas, Texas 75229
(214) 484-9428

Second Manual Printing, March 1983

Super Utility 3.0

Table of Contents

PREFACE TO SUPER UTILITY PLUS VERSION 3.0

Super Utility Plus version 3.0 represents an entirely new program. Both program and manual have been rewritten almost from the ground up. The program has beenendowed with new abilities which users have requested over the years that Super Utility Plus was on the market. For example, it will now deal correctly with some double-sided disks, and incorporates support the for the Radio Shack double-density modification for the Model I TRS-80. The new operating systems such as MULTIDOS, TRSDOS 2.7DD and DOSPLUS 3.5 are now fully supported.

The manual has been rewritten to provide new users as well as old ones with more information regarding the operation of the program and the thinking that went behind it. Old users may find this manual a bit redundant in spots, but they should keep in mind that someone who is using this program for the first time may need that redundancy to help him learn how to use the program. Users are cautioned against skipping over the initial chapter. The first chapter of this manual contains information which will be assumed in the later sections of the manual. All users are urged to read the first chapter carefully before proceeding to use the program.

A section has been included giving an overview of diskette structure as it pertains to TRS-80 systems. However, this manual is not intended to be a tutorial on how disk storage systems are organized. If you desire in-depth information on this subject, there are other publications available which may be of use to you. The classic is, of course, William Barden's **TRS-80 Disk Interfacing Guide**. Another useful publication is Paul Wiener's **Inside Super Utility Plus** It contains much useful information which is also relevant to a user of Super Utility Plus 3.0.

Super Utility Plus requires a minimum hardware configuration of 48K of RAM memory and at least one disk drive. The tape copy function will require two tape recorders. The efficiency of the program is considerably enhanced when two or more disk drives are available.

The Super Utility Plus diskette is designed to boot up on either a 35-track, 40-track or 80-track disk drive, on the Model I or Model III TRS-80. This program does not require any external support once it is loaded. Therefore, as soon as the program has been loaded and is running, you are urged to open drive 0, take out the Super Utility Plus diskette, and **PUT IT AWAY**. The less time it stays in the drive, the better is chances are against being damaged or destroyed by accident.

REGISTRATION AND TECHNICAL SUPPORT

Please fill out the registration card which came with your Super Utility Plus and mail it at your earliest opportunity to PowerSOFT, 11500 Stemmons Expressway, Suite 125, Dallas TX 75229. Registered owners will be placed in our database and PowerSOFT will notify all registered owners of any upgrades, enhancements, fixes or new releases for this program. If you have any urgent questions regarding the operation of this program, you may call PowerSOFT Technical Support at 214/484-9428 between the hours of 10 a.m. and 5 p.m. CENTRAL TIME, Mondays through Fridays. If you are going to call, please have your serial number on hand. You will be asked for it, and you must supply it and be verified in the PowerSOFT customer data base before you can receive any technical support. It would also help if you could be at your computer with Super Utility Plus booted up and running when you call.

Super Utility Plus, to the best of our knowledge, contains no major bugs, but it is entirely conceivable that some minor bugs may exist somewhere in the 34K of machine code that make up this program. If you encounter a problem, please re-read the pertinent section of the manual first and determine whether or not it is a real problem. If you are unable to overcome it, please verify the exact circumstances under which it occurs and send a complete description of the problem to PowerSOFT Technical Support at the address above. Include your **SERIAL NUMBER**. We will do our best to assist all registered owners.

This program is normally supplied with a backup disk. If you did not get a backup disk with this package, you may purchase one for $10.00 as long as you are a registered owner. If your disk becomes unbootable and the program itself is not at fault (e.g., your dog chewed it up, your son poured coke over it), you may send the disk back to us for replacement. The replacement cost is $8.00, plus $5.00 if the disk is in unusable condition. These prices are subject to change without notice.

CHAPTER 1 - INTRODUCTION

Super Utility Plus 3.0 is an extremely powerful utility package for use on TRS-80 Model I and Model III disk-based microcomputers. Like its predecessor, Super Utility Plus 2.2z, its many routines allow you, the user, to perform a great variety of tasks, ranging from direct examination and modification of the contents of a diskette to restoring an unreadable diskette to a usable condition. With Super Utility Plus you may also format diskettes in a variety of ways (including a mixed-density track!), backup a diskette to another using a very fast routine, examine a particular file on a diskette, or examine and modify the contents of your TRS-80's memory.

This manual will explain the various utility routines available in Super Utility Plus. Please read it carefully before attempting to use the program for the first time. Due to its power and versatility, Super Utility Plus can do great damage to your diskettes if carelessly used.

It is assumed in this manual that you are familiar with your disk operating systems' features. More information on this may be found in your disk operating system manual.

OVERVIEW OF SUPER UTILITY PLUS

These are the utilities available to you in the program:

```
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. ##  SUPER-UTILITY !  ##  VERSION 3.1A  ##  BY: KIM WATT  ##  .
.       (C)(P) 1983 BREEZE/QSD, INC.  --  DALLAS, TEXAS        .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. __ UTILITY FEATURES __                                                .
.                                                                       .
.  1 - ZAP UTILITIES            6 - TAPE UTILITIES                     .
.  2 - PURGE UTILITIES          7 - MEMORY UTILITIES                   .
.  3 - FORMAT UTILITIES         8 - FILE UTILITIES                     .
.  4 - BACKUP UTILITIES         9 - CONFIGURATION                      .
.  5 - REPAIR UTILITIES        10 - EXIT PROGRAM                       .
.                                                                       .
.   CHOICE? #__                                                         .
.                                                                       .
.                                                                       .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
```

Each selection is preceded by a number and represents a group of utilities. By entering the appropriate number on your keyboard, you will be taken into a further menu which will allow you to choose from the utilities available in that group.

ZAP Utilities are routines which permit you to directly examine, modify and copy the contents of a diskette. The routines in this group will also permit you to search a disk for a particular occurrence of bytes or characters.

PURGE Utilities will allow you to quickly and easily kill or recover files from a TRSDOS, LDOS or data diskette and clean up your directory in the process. You may optionally remove all traces of a file from a diskette, change the diskette name, and view the disk directory before and after making changes.

DISK FORMAT Utilities are routines for formatting a diskette in a variety of ways. You may construct format tracks to your own specifications, even tracks with mixed single and double density sectors. You may even reformat a diskette without losing any data previously written on it! There is even a routine which removes all traces of data from a diskette, in effect bulk-erasing it.

The **BACKUP Utilities** will perform standard or special backups of one disk onto another. The destination diskette may optionally be formatted before backup begins. These backup routines are very fast and very intelligent.

The **REPAIR Utilities** will let you restore an unreadable disk directory or damaged boot sector to a usable condition, if at all possible. You may also recover files which were killed by Super Utility Plus, and check the disk directory for any errors which may result in the destruction of files later on.

TAPE Utilities will allow you to perform a variety of tape-to-tape or tape-to-memory tasks. 500 baud tape rates are supported by these routines.

MEMORY Utilities will perform a variety of functions of the RAM memory of your TRS-80. You may examine the contents of memory, move the contents of a segment of memory to another location, search memory for the occurrence of a particular string, read a port, send a byte to a port, and transfer memory to disk and vice-versa. Additionally, you may also transfer to your own machine-language subroutine in memory.

FILE Utilities are similar to the ZAP utilities, except that they are file-oriented. You may use them to view the contents of a file on a diskette without knowing exactly where on the diskette the file resides. You may compare two files for differences, copy files from one diskette to another, display the free space on a disk, display the locations of files on a disk, create new files, and check the status of your disk drives.

The **CONFIGURATION** system allows you to tailor Super Utility Plus to your own system. You can tell Super Utility Plus the characteristics of your lineprinter, if you have one, how many disk drives you have, and what kind of disks you expect to be using in each drive. You can also optionally software write protect a particular disk drive so that no data on it can be inadvertently changed. The configuration parameters may be saved on disk for automatic loading when the program is rebooted.

EXECUTING SUPER UTILITY PLUS

The Super Utility Plus program is supplied on a special "self-booting" diskette. Insert the Super Utility Plus diskette into your drive 0 and press the RESET button. The Super Utility Plus logo will appear on the screen while your disk drive continues to run. After a few seconds the disk drive will stop, and the logo will animate. Press any key and the program will bring up the main menu.

You should be aware that during the entire loading process, Super Utility Plus is performing a memory test on your computer's RAM to ensure correct operation of the program. If the message "memory error" appears on the screen, you should get your computer's memory tested and replaced if necessary.

Former versions of Super Utility Plus would display the program labels if you held down the left or right arrow keys during the bootup process. This version no longer has that feature. All the program labels and register conditions are documented in the Super Utility Plus 3.0 Technical Manual.

Super Utility Plus does not require the presence of a DOS system disk in drive 0 at all. Nor does it require the presence of its own disk in drive 0. Once you have successfully brought up the main Super Utility Plus menu, remove the Super Utility Plus disk from the drive and put it in a safe place. You should never keep the Super Utility Plus diskette in the drive any longer than absolutely necessary.

Super Utility Plus is menu-driven, that is, its various functions are accessed through a series of menus which appear on your screen. By keying in the appropriate number for the routine you want, you will activate that particular function. Simply pressing **ENTER** will default to the first selection on the displayed list. To exit a function at any time, press the **BREAK** key. To return to the main menu at any time, hold down the **SHIFT** key and press **BREAK**.

If you are prompted for additional input after selecting a routine, you may enter requested numeric information in either decimal (the default base), hexadecimal (by appending **H** after the number), octal (by appending the letter **O** or the letter **Q** to the number), or binary (by appending **B** to the number). String input may be typed in directly, and lower case may be used at any time. **SHIFT-0** ("shift-zero") acts as a reverse case toggle. Pressing it once will lock you into reverse case (normal upper case, shifted keys lower case), pressing it a second time will restore normal upper/lower case.

When several inputs are requested, you may enter them separated from each other by commas, as you would when answering a BASIC prompt for multiple numbers. Alternatively, the inputs can be separated from each other by spaces.

In addition to the various keyboard functions, Super Utility Plus has a powerful screen printer built right into the program. Pressing **SHIFT-CLEAR** at any time after the main menu has been brought up will cause whatever is displayed on your screen to be reproduced on your lineprinter. Pressing **Shift-@** key alone will empty the printer buffer. If your lineprinter is

capable of producing TRS-80 block graphics, you can configure Super Utility Plus to do so; otherwise it will change graphics symbols to "#" signs before printing.

When a display is scrolling on the screen, as when you select "File Allocations" from the Utilities menu, the display may temporarily be frozen by pressing the spacebar. Pressing ENTER will cause the program to resume.

In addition to the various key controls, three control combinations are active at any point of the program. These are:

Clear-A - Turns the moving graphics characters at the corners of the screen ON or OFF.

Clear-V - Displays the Super Utility Plus version number and the assembly date of your copy. This will help you in deciding whether or not to send your copy in for an update.

Clear-I - Toggles an INKEY function. When turned ON, single-key responses to prompts will be acted upon immediately. Otherwise, the program will wait for the ENTER key to be pressed.

These control combinations require that you hold the **CLEAR** key down and press the second key.

The main menu presents you with the various groups of utilities available in the program:

```
1 - ZAP Utilities        6 - TAPE Utilities
2 - PURGE Utilities      7 - MEMORY Utilities
3 - FORMAT Utilities     8 - FILE Utilities
4 - BACKUP Utilities     9 - CONFIGURATION
5 - REPAIR Utilities    10 - EXIT Program
```

By entering the number to the left of each selection and pressing ENTER, you will be taken to a further menu which will present you with the various routines available under that group. Pressing ENTER alone will always take you to the first selection on the displayed menu.

The following chapters will deal with each group of utilities in detail. Please read each chapter carefully and keep the manual handy while using Super Utility Plus. We cannot emphasize enough the dangers of using this program in a careless fashion. If you are not sure of something, check the manual.

CONFIGURING SUPER UTILITY PLUS

Super Utility Plus may be configured to your system very easily. In the main menu, you will see a selection that reads, "**CONFIGURATION**." If you press the number **9** on your keyboard, you will see a display that looks like this:

```
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. ##  SUPER-UTILITY !  ##  VERSION 3.1A  ##  BY: KIM WATT  ##  .
.       (C)(P) 1983 BREEZE/QSD, INC.  --  DALLAS, TEXAS        .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. __ CONFIGURATION __                                                  .
=>DUAL=N GRAPHICS=N LOCASE=N LINEFEEDS=N DOUBLER=R SPEED=N     .
. +:0 T3D'  PTKS= 40 RTKS= 40 DIR= 17 STP=3 RDLY=4 WDLY=4 WP=N .
.         D0=D DD=D LS0=1 HS0=18 LSD=1 HSD=18 S/G=3 G/T=6 DD=I .
. +:1 T3D'  PTKS= 40 RTKS= 40 DIR= 17 STP=3 RDLY=4 WDLY=4 WP=N .
.         D0=D DD=D LS0=1 HS0=18 LSD=1 HSD=18 S/G=3 G/T=6 DD=I .
. +:2 T1S'  PTKS= 35 RTKS= 35 DIR= 17 STP=3 RDLY=4 WDLY=4 WP=N .
.         D0=S DD=S LS0=0 HS0=09 LSD=0 HSD=09 S/G=5 G/T=2 DD=S .
. +:3 T1S'  PTKS= 35 RTKS= 35 DIR= 17 STP=3 RDLY=4 WDLY=4 WP=N .
.         D0=S DD=S LS0=0 HS0=09 LSD=0 HSD=09 S/G=5 G/T=2 DD=S .
. ? #_____________________________________________________    .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
```

There are several lines to the display, and you will see a little arrow to the left of the top line:

```
=> Dual=N Graphics=N Locase=N Linefeeds=N Doubler=R Speed=N
```

The arrow indicates which line is being worked on, and you can bring it down by simply pressing the ENTER key.

The first four items on this line pertain to your system's lineprinter and its characteristics. **DUAL** refers to whether or not you want dual output on or off; if you turn it on, everything that appears on your screen will be sent to the lineprinter simultaneously.

GRAPHICS refers to whether or not your lineprinter is capable of printing the TRS-80 block graphics symbols. If this is turned on, then Super Utility Plus will send the graphics codes out to your lineprinter. Otherwise, it will replace all graphics symbols with pound signs ("#").

Some printers can generate the TRS-80 block graphics, but do not use the same ASCII codes as the computer uses. An example of this is the popular EPSON MX-series printers. If you are using an EPSON MX printer, you may force the correct offset by entering **GRAPHICS=M** (For "MX"). This will cause Super Utility Plus to output the correct graphics codes. Other such printers, however, must be considered as incapable of generating the graphics codes, since Super Utility Plus does not possess any ability to offset the computer's internal graphics codes to the codes that the printer uses.

LOCASE simply refers to whether or not your lineprinter can print lowercase letters or not. If turned on, Super Utility Plus will send all lowercase codes as is; otherwise, lowercase codes will be translated to uppercase before sending to the printer.

LINEFEEDS tells Super Utility Plus if your lineprinter requires linefeeds after a carriage return to move to the next line. Most Radio Shack lineprinters do this automatically, and normally this option would be left off.

To turn the options on or off, simply enter Y (yes) for on, or N (no) for off, for each option. For example, if you wanted to set your options as follows: Dual mode on, no graphics (your printer can't generate them), lowercase on, linefeeds off, you would enter on the command line:

? Y,N,Y,N

and press the ENTER key. Note that you only have to enter the answers, but they must be in the correct relative position. As soon as you hit enter, you will see the top line change to reflect your answers, and the arrow (**=>**) will move down to the next line. Invalid answers will cause the line to be prompted for again; no change will take place in the configuration tables beyond the incorrect entry.

The next item on this line is relevant to Model I users only, and allows you to indicate to Super Utility Plus whether or not you have a Doubler (a double-density adaptor) installed in your machine. It is normally set to **DOUBLER=R**, indicating the presence of a Radio Shack doubler. If you do not have a doubler installed, you should change this to indicate N. This will prevent Super Utility Plus from making any attempt to read a double-density disk. If you have a double density board installed that is not a Radio Shack kit, enter X (for "Brand X"). In any case, Super Utility Plus can usually determine which type you have, and will alter this setting accordingly.

The last item on this line is **SPEED** and is used to indicate the presence of a CPU speed-up modification. If set to Y, Super Utility Plus will assume that your CPU is operating at high speed and adjust accordingly. Following the speed parameter, you may insert the code necessary to turn your high speed modification ON and OFF. There are no prompts on the configuration line for this, but you may enter up to 8 hexadecimal bytes each for the ON code (preceded by **O=**) and OFF code (preceded by **F=**). If you enter less than 8 hexadecimal bytes, the string will be padded with NOPs by Super Utility Plus. For example, if your high speed modification was turned on by an OUT (0FEH),1 instruction and off by an OUT (0FEH),0 instruction, you could enter D3FE01 for the on code and D3FE00 for the off code, as follows:

... (preceding input) Y,O=D3FE01,F=D3FE00

The next lines refer to the disk drives in your system. Each drive is described by two lines, but only the options on the first line may be changed; the data on the second line for each drive is implied from the first.

Let's take a look at the entry for drive 0:

```
=>+:0 T3D' PTKS=40 RTKS=40 Dir=17 Stp=3 Rdly=4 Wdly=4 Wp=N
   D0=D Dd=D LS0=1 HS0=18 LSd=1 HSd=18 S/G=3 G/T=6 DD=I
```

To the left of the drive number is a plus sign. This means that Super Utility Plus will recognize that drive as being in the system. If it were a minus, then Super Utility Plus will assume that it is not in your system and will refuse to do any I/O to it. You can change this by typing a minus sign as the first character in the prompt line.

If you had only two drives, 0 and 1, for example, you might want to remove drives 2 and 3 from the system. In this way, if you should inadvertently reference drive 2 or 3, you will be informed that the drives are not available in the system.

You may also specify "=", indicating that Super Utility Plus should operate in "**SKIP**" mode for this drive. This is used only when trying to read a 35 or 40 track diskette in an 80-track disk drive (**NOTE:** NEVER write to a 35 or 40 track diskette while it is in an 80-track drive. This disk may not be readable in a 35/40 track drive afterward.)

T3D' is a **DOS SPECIFIER**. This tells Super Utility Plus what Disk Operating System (DOS) it can expect to find in that drive. The DOS specifiers recognized by Super Utility Plus are given below. Note that whenever more than one item appears in the Model I or Model III column for a particular DOS type, these items are equivalent in meaning and may be used interchangeably.

DOS	**Model I**	**Model III**
TRSDOS		
Sgl. Den.	T, T1, TS, T1S	* Invalid *
Dbl. Den.	T1D	T3, TD, T3D
LDOS		
Sgl. Den.	L, L1, LS, L1S	L, LS, L3S
Dbl. Den.	L1D (SOLE disk) (note 1)	L3, LD, L3D
DOS PLUS		
Sgl. Den.	D, D1, DS, D1S	D, DS, D3S
Dbl. Den.	D1D (system disk) (note 1)	D3, DD, D3D
MULTIDOS		
Sgl. Den.	M, M1, MS, M1S	M, MS, M3S
Dbl. Den.	M1D (system disk)	M3, MD, M3D (note 1,2)
NEWDOS80 V2		
Sgl. Den.	N, N1, NS, N1S	N, NS, N3S
Dbl. Den.	N1D (Tk 0 rev.)	N3, ND, N3D (note 1,2,3)

DBLDOS

Dbl. Den.	B, B1, BD, B1D	* Invalid *

Note 1: For model I data disks which do not have the single density track 0, use one of the Mod III codes.

Note 2: Relative sectoring as required by NEWDOS80 V2 double density disks is specified by appending the letter R to the specifier, e.g., N1DR. This is mandatory for correctly handling ND80 V2 and DBLDOS double-density systems in file and directory-oriented operations. The R modifier is allowed only in MultiDOS Model I double-density to produce the "P-density" disks.

Note 3: For NEWDOS80 V2 diskettes with a normal Track 0, use one of the corresponding Mod III specifiers.

As you can see from the table above, the system of DOS specifiers used in Version 3.0 of Super Utility Plus is considerably changed from earlier versions. Now you have a choice of which specifier to use, and you can select whichever is most meaningful for you.

The "**R**" modifier allows Super Utility Plus to correctly handle disks which use a relative track/sector formatting scheme, such as NEWDOS/80 Version 2 double density disks. This modifier is mandatory for any file-oriented operation, or any operation which affects the diskette directory. However, it may be left off for other operations such as disk sector modifications, standard backups and formats (including format without erase). If you run into problems with this modifier, try repeating the operation -- but leave the "R" off the DOS specifier.

Double sided media support: Limited support for double-sided media is now available in this new release of Super Utility Plus. Only LDOS, DOSPLUS and MULTIDOS are supported. A double-sided disk is specified by appending a double-quote to the DOS specifier, e.g., **L3D"** or **DD"**. Conversely, a single-quote appended to the DOS specifier indicates a single-sided disk. Most functions (with the exception of COPY FILES) can be performed on double-sided disks.

The DOS specifiers may also be used within the other routines of Super Utility Plus to override the configuration table settings without returning to the configuration routine. To override the current setting, append the required DOS specifier to the drive number on a prompt line. For example, when prompted for drive, track and sector, you might enter something like this:

0M3D'=40,21,5

By appending M3D to the drive number, you are telling Super Utility Plus that the disk in that drive is now a Model III double-density MULTIDOS disk. **=40** indicates that is it formatted for 40 tracks. Similarly, when asked for a filespec, the following form can be used:

MYFILE/BAS:3LD"=40

to indicate that the disk containing MYFILE/BAS in drive 3 is a double-sided double density LDOS disk. The use of the override system avoids the necessity of having to return to the configuration routine each time you wish to scan a different disk type.

PTKS=40 is the number of formatted physical tracks that Super Utility Plus expects to find on the diskette in drive 0. If you are going to put in a diskette with a different track count in this drive, you should change this figure.

RTKS=40 refers to the number of relative tracks on the diskette. For systems that use a relative track scheme, like NEWDOS80 Version 2, this figure will differ from that of PTKS. Note: this item is not user controllable. It is calculated from the PTKS value.

DIR=17 tells you which track the diskette directory is located in. If a relative track scheme is being used, this value should be the relative and not the physical track location of the directory.

STEP=3 refers to the head stepping rate of your disk drive. This is a coded value, and the corresponding rates are as follows:

0 5/6 milliseconds
1 10/12 milliseconds
2 20 milliseconds
3 40 milliseconds

A standard Radio Shack Model I disk drive is normally capable of stepping its read/write head from one track to the next at 20 milliseconds, although some may be slower (older drives cannot step faster than 40 milliseconds). Model III Radio Shack drives are for the most part capable of stepping at the fastest rate, 6 milliseconds. You should consult the drive's specifications for the correct rate. Do not specify a step rate faster than the drive is capable of, or you will produce I/O errors which can ruin your disk. If in doubt, set the step rate to 3 (almost all disk drives are capable of this speed).

RDLY=4 refers to the delay (in quarter-seconds) from the time a drive's motor comes on to the time when Super Utility Plus first attempts to read from the disk. If set to "4", there will be a one-second delay from motor-on to the first read attempt. If you find that Super Utility Plus is having trouble reading your disks for the first time, you may need to set this delay factor to "4".

WDLY=4, conversely, controls the amount of delay from the time the disk motor comes on to the time Super Utility Plus first attempts to write to the disk. If set to "4", Super Utility Plus will delay 1 second before attempting a write operation. If set to "2", it will wait only one-half second. Since writes are more critical than reads, this delay factor is controlled separately to ensure reliable writes to the disk. Again, if disk I/O errors occur as a result of bad writes to the disk, you may wish to make sure this item is set to "4".

Both RDLY and WDLY can take values from 1 to 4. A value of 0 (theoretically, no delay) is invalid.

WP=N is a "switch" which tells Super Utility Plus whether or not to "write-protect" your drive. If you specify "Y" to this, Super Utility Plus will read any diskette in that drive, but will not write to it. This is functionally equivalent to putting a write-protect tab on your diskette.

The second line contains information about the disk's formatting.

D0=D indicates the density of track 0. Some disks have track 0 formatted in a different density from the rest of the disk. One example of this is a double-density Model I LDOS system disk to which the "SOLE" modification (a trademark of MISOSYS, Alexandria, VA) has been applied to permit it to boot up. Such a disk has a single-density track 0. This item will indicate S for single density or D for double density, and pertains to track 0 of the disk only.

Dd=D indicates the density of the rest of the disk. Dd stands for "Density of disk." This will be S for single and D for double.

LS0=1 indicates the sector number of the lowest sector ON Track 0. TRSDOS 1.3 numbers sectors from starting from 1. All other systems number their sectors from 0.

HS0=18 tells you the sector number of the highest sector on Track 0. For a single-density disk, this will be 9. For TRSDOS 1.3 this will be 18. Double density Model I disks which are designed to boot up on that computer will show this value as 9.

LSd=1 indicates the sector number of the lowest sector on the rest of the diskette's tracks. This will be 1 for TRSDOS 1.3 and TRSDOS 2.7DD, 0 for all others.

HSd=18 is the number of the highest sector on the rest of the diskette's tracks. This value will be 18 for TRSDOS 1.3 and TRSDOS 2.7DD, 9 for other single-density non-RS systems and 17 for double density systems.

S/G indicates the number of sectors per granule. Since the granule is an arbitrary unit, its size can, and indeed does, vary. A single density disk will have 5 sectors per granule. A double-density TRSDOS 1.3 and 2.7DD disk will have 3 sectors per granule. Double-density LDOS, DOSPLUS and MultiDOS disks use 6 sectors per granule.

NEWDOS80 V2 uses "lumps" instead of granules, and the size of a lump can vary. If you are going to work on a NEWDOS80 V2 disk with Super Utility Plus, your disk should be configured as closely as possible to the standard "granule" sizes as defined in the preceding paragraph. Super Utility Plus cannot correctly handle lump sizes that are widely different. The configuration table will always show 5 sectors per gran on relative sectoring.

G/T stands for granules per track. This value will obviously vary according to the way a granule's size is defined. For TRSDOS 1.3 and 2.7DD, this will be 6. For single density disks

(TRSDOS and other non-RS systems) it will be 2, and for double density non-RS systems it will be 3 (NEWDOS80 disks will show 2 grans per track on relative-sectored diskettes).

DD indicates the type of data address marks used by the disk. Each disk writes one type of data address mark for the data tracks and another type for the directory track. For all operating systems except TRSDOS 1.3, this will read S meaning "standard convention." For TRSDOS 1.3, it will read I for "inverted."

Note that the information on the second line of each drive configuration is IMPLIED from the first line, and therefore cannot be changed directly. Only the items on the first line can be changed, with the sole exception of RTKS.

To alter the settings, you must enter a series of answers to the prompt which describe your disk drive. The settings in the example above would have been given by this string:

?+,T3,40,17,1,4,4,N

The + is optional and indicates that the drive should be active in the system; the rest of it indicates that Super Utility Plus should expect a TRSDOS 1.3 (T3) disk with 40 formatted tracks, a directory on track 17, in a drive capable of stepping at 12 milliseconds (1 – see table above) requiring motor-on delays for both reads and writes, and that the drive should not be write protected. Note also that the side specifier is not entered, and will default to ' (single sided disk).

If we were to change the specifications to a single-density model I disk, we might enter,

?+,T1S,35

T1S indicates that the disk in the drive will be single density TRSDOS 2.3, and will have 35 tracks. The rest of the information is the same and does not have to be entered. In most cases, it is also unnecessary to pre-set the directory track; Super Utility Plus will find it.

If you wish to take a drive out of the system, it is only necessary to enter the - to disable that drive -- the other specifications will obviously not matter.

In general, the only things that really need to be configured are the lineprinter specifications, and for each drive the DOS type, step rate, motor-on delay, the software write-protect switch, and whether or not a particular drive is to be actively in the system. The track count and directory location of the disks are determined when Super Utility Plus goes to read a diskette in the specific drive, and will change accordingly. Sometimes, if you swap a disk of a different density into that drive, read it, and then view the configuration table, you will see that the table has changed to reflect the density of that disk.

If you are configuring your disk drives in an identical manner, (e.g., all of one DOS type), you may simply configure the first drive of that type. Then, with the pointer on the next drive, enter on the prompt line:

<d

where "d" is the drive number whose configuration characteristics you want copied into the current drive. For example, if you had configured drive 0 to DD",40,20, etc., and you want the exact same configuration on drive 1, you would move the pointer down to the drive 1 configuration listing and enter on the prompt line, **<0**. The configuration characteristics of drive 0 will be copied into drive 1's control table.

Conversely, if you want to copy the configuration data of one drive to another drive, position the => pointer to the source drive (the drive you want the configuration data copied from) and enter a >d at the prompt line. So if you wanted to copy the configuration of drive 0 into drive 3, for example, you would position the => at drive 0, and then type **>3**. The configuration information of drive 0 will be copied to drive 3.

Other modifiers may follow the ">d" and "<d" specifications. For example, you might enter on the prompt line,

<0,80,40

This would FIRST copy the drive configuration of drive 0 into the current drive. THEN it would alter the current drive's specifications to indicate an 80-track drive with a directory on track 40.

Note that if you are going to use these specifications, "<" or ">" must be the first character on the line. If it is not the first character on the line, the whole line will likely be rejected.

After you have configured Super Utility Plus' disk drive settings to your specifications, press ENTER once more. You will now be asked whether you wish to save the configuration or not. If you reply "Y", you will be asked to mount the Super Utility Plus diskette in drive 0. Make sure there is no write-protect tab on the disk. Press ENTER. Your configuration will be written out to the disk and will automatically be established the next time you boot Super Utility Plus up.

OVERVIEW OF DISKETTE DATA STRUCTURE

When you format a diskette using the FORMAT program from your system disk, information is written to the diskette in concentric rings called tracks. On a Model I TRS-80, each diskette is divided, or "formatted," into 35 tracks, or cylinders (the newer model I drives can be formatted to 40 tracks, but this depends on the operating system. TRSDOS 2.3 normally recognizes only 35). On a Model III, a diskette is formatted into 40 tracks.

Each track in turn is divided into sectors. The number of sectors on a track depends on the diskette's density. Model I TRS-80's normally format diskettes in single density unless they are equipped with a double density modification, in which case they can format either single or double density. Model III TRS-80s format them in double density. A single density track contains 10 sectors, while a double-density track contains 18 sectors. Note that these are the actual, physical numbers of sectors on a track. What the operating system CALLS them can be different. But whatever the density of the diskette, a sector will always contain 256 bytes (unless a special formatting scheme was used) of data plus some additional bytes of information which identify the sector to the computer in terms of its track and sector location.

Each sector has two header fields, the ID field and the data field. The ID header is a 6-byte field which consists of an address mark, a track designator, a head designator, a sector number designator, a length byte, and a two-byte CRC (see below). There is no separate field for a track number; this is specified in the as part of the header data for every sector on the track. The head designator is used to identify which side of a double-sided diskette the sector is on.

Following the sector number is a length byte, which is an encoded value of the number of data bytes in the sector. TRS-80 diskettes normally use IBM conventions, in which a 0 length byte corresponds to 128 bytes of data, 1 corresponds to 256 bytes, and so on. See the next chapter for an explanation of non-IBM conventions. In double density diskettes, only IBM length values from 00 to 03 are valid, corresponding to lengths of 128, 256, 512 and 1024 bytes per sector respectively.

The header field is placed on the diskette at format time, and is never altered unless the disk is reformatted. The second field, however, is the data field, and that is changed each time the sector is written to. This field contains the actual data for the sector along with 2 CRC bytes.

Each field (ID header and data) is preceded by a gap consisting of 12 bytes of FFH and 6 bytes of 00H. These bytes are placed there to separate the sectors. Note that depending on the formatting and density, the gap bytes may be some value other than FF or 00. There is no firm convention established as to what these bytes should be.

The ID and data fields on each sector are each preceded by an address mark which is used by the FDC to detect the start of a block of data. The address marks are also used in the TRS-80 disk operating systems to identify a track as being a part of a directory track, or part of a data track. This is discussed below.

The sectors on each track are grouped together into units called granules, or "grans." A gran is the smallest unit TRSDOS (or a TRSDOS-compatible system) will allocate to a diskette file. Machine-language programs, BASIC programs, and data files are all "files" to the system and are assigned granules of space on the diskette as needed. As a file grows (for example, in a mailing list file which is being added to) the DOS will assign more grans to it to hold the additional data. The use of this grouping prevents excessive thrashing around of the disk drive's read/write head and also allows for faster file accesses. A single-density diskette, such as that produced by Model I systems without the Radio Shack double-density adapter, will have two grans of 5 sectors on each track. A Model III TRSDOS double- density diskette will have six grans of three sectors each per track. Most double-density non-RS diskettes will have three granules of six sectors each per track.

When a file is saved to disk, it may occupy several grans of space. It is the job of the disk operating system to keep track of where the file is located on the disk, and how much space it takes up. Sometimes a file must be broken up into several segments, or extents, in order to make maximum use of the available space on a diskette. The disk operating system keeps track of all this and maintains information on each file in a special place on the disk called the directory track.

When you request a file, such as what happens when you issue the RUN "filename/ext" command from Disk BASIC, the DOS first goes to the diskette's directory track and looks for the file you specified. If it finds the file, the system then examines the information associated with it to find out where on the disk the file resides, and how long it is. Armed with this information, it then goes to the spot on the disk where the file begins, and proceeds to load it into the computer's memory.

The directory track is "marked" off from the rest of the tracks on the diskette by the use of a special data address mark, or DAM. This is a piece of information written onto the disk by the DOS during the format process, and permits it to locate the directory track quickly, since it is different from the address marks used on the other tracks. However, the type of DAM written to the directory is dependent on the hardware in the machine, specifically, the floppy disk controller chip. The Model I uses a different controller chip than the Model III. This poses certain problems, the major one being the fact that you cannot read a Model I single-density TRSDOS disk's directory on a Model III without changing the ID marks to ones the Model III can read. Reading Model III diskettes on a Model I, however, poses no problems if the Model I is equipped with a double-density adapter. Other operating systems avoid this problem by using a data address mark for the directory track that both Model I and Model III controller chips can read and write.

The directory track, then, is one of the most important parts of a diskette. If the directory track is damaged in some manner so that the operating system cannot correctly read the information on it, the disk essentially becomes unusable. The file itself is still on the disk, but the system no longer has any way of finding it.

There is one other place on the disk which is vital to the operation of a system disk. This is the very first sector on the first, or outermost, track of a diskette. This sector is called the boot

sector. On a Model I TRSDOS single-density diskette, and on all other non-RS systems' diskettes, this will be Sector 0 of Track 0; on a Model III TRSDOS diskette it will be Sector 1 of Track 0. When a system disk is placed in Drive 0 and the TRS-80's RESET button is pushed, special code in the ROMs orders the disk drive to move its read/write head to track 0 and read the boot sector into memory (starting at 4200H for the Model I, 4300H for the Model III). Once the boot sector is in memory, the computer jumps to its starting address. This sector contains information which then permits the computer to read the rest of the operating system into memory. If the diskette is not a system diskette, the boot sector produces the "NO SYSTEM" notice on your screen.

If the boot sector on a system disk is damaged, that disk also becomes unusable as a system disk, although it may still be possible to use it as a data disk. Super Utility Plus has the capability of restoring to a usable condition disks which have sustained damage to the directory track or boot sector, if the damage is not too extensive. Sometimes the damage is so widespread that no recovery is possible, but in many cases, a disk can be restored to working condition at least long enough for you to copy important files over onto another diskette.

Many users who find that one of their system disks will not boot automatically assume that the boot sector has been damaged. This may not necessarily be the case. The process of bringing a DOS up to the point where it is ready to accept user input consists of several steps, and a failure at any one of these steps may give the impression that the boot sector is bad. However, it may be that the resident module (SYS0) has been damaged, is missing or is in the wrong place; or that the command interpreter (SYS1) is damaged or missing. Users who are unaware of this process will simply assume that the boot sector need to be repaired, and will proceed to do so. But when the system still fails to come up then they may become confused and think that somehow Super Utility Plus has developed a bug or malfunction, when in fact it may have done exactly what they wanted. Exactly, but no more.

When this happens to you, it pays to explore other possibilities before assuming the something is wrong with the program. Try copying a good SYS0 or SYS1 module from an undamaged disk onto the bad one. That may be all you need to get the system up and running again.

CHAPTER 2 - ZAP UTILITIES

Bring up the main Super Utility Plus menu as described above, and press 1. You will then be presented with the ZAP Utilities menu.

```
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. ##  SUPER-UTILITY +  ##  VERSION 3.1A  ##  BY: KIM WATT  ##  .
.       (C)(P) 1983 BREEZE/QSD, INC.  --  DALLAS, TEXAS        .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. __ ZAP UTILITIES __                                                  .
.                                                                      .
.  1 - DISPLAY SECTORS           7 - REVERSE SECTOR DATA               .
.  2 - VERIFY SECTORS            8 - EXCHANGE SECTORS                  .
.  3 - COMPARE SECTORS           9 - STRING SEARCH                     .
.  4 - COPY SECTORS             10 - SECTOR SEARCH                     .
.  5 - COPY SECTOR DATA         11 - READ ID MARKS                     .
.  6 - ZERO SECTORS             12 - ALTER DATA ADDRESS MARKS          .
.                                                                      .
. CHOICE? #__                                                          .
.                                                                      .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
```

Select the routine you want by entering its number and pressing the <ENTER> key (see previous chapter for enabling INKEY-style entry).

I. DISPLAY SECTORS

When this option is selected, you will be presented with an additional prompt:

Drive, Track, Sector ?

Enter the number of the disk drive containing the disk you want to examine, then the track number and sector number you wish to display. Separate them from each other with a comma, for example,

Drive, Track, Sector ? 1,0,1

and press ENTER. If you wanted to read a diskette with a differing track count or density on the same drive, an alternative form would be 1T1S=40,0,1 where T1S indicates that the diskette is single density TRSDOS, and =40 means it is formatted for 40 tracks. See the previous chapter's discussion on configuring the system for more details. If you omit the track and sector, you will be shown the lowest sector on the lowest formatted track on the disk. If you enter D for track, Super Utility Plus will go to the configured directory location. If you enter an up-arrow for the track, then you will be shown the highest configured track on that diskette.

On this particular option only, you can also use two special symbols in front of the drive number. The first special symbol is a pound sign, or "#". This will cause Super Utility Plus to identify the diskette's density automatically, in case you are unsure of its density. This will work only with Display Sectors. This will also allow you to display non-standard sectors, however it will not scan the back side of a two-sided disk.

The second special symbol is an exclamation mark, or "**!**." This switches in automatic DOS recognition. Super Utility Plus will examine the disk and determine which DOS formatted it. This process takes a few seconds. This option is also available at any other routine which reads a diskette's directory. Due to the similarity of many DOS'es, this routine is not totally reliable. It will serve to identify whether a disk is TRSDOS-formatted or LDOS compatible. Note that DOSPLUS and MULTIDOS disks can be mistaken for LDOS disks at times. The main purpose of this routine is to render a disk readable to Super Utility Plus, and it is up to the user to decided whether Super Utility Plus' identification is sufficient or whether he should go in further.

The particular sector you requested will be displayed on your screen. It will look something like this:

```
#    00|F5DB ED2F 2173 4477 2CA6 2808 2C1F 380D|###/!SDW,*(.,.8.
HEX  10|2CB7 20F8 CDBE 4D20 18F1 E1FB C9F5 C5D5|,# ###M .#######
DRV  20|E5DD E511 D244 D55E 2C56 EBE9 DDE1 E1D1|###.#D#^,V######
 0   30|C1F1 18DC 3007 C5CD 5047 C118 DC3A 2844|##.#0.##PG#.#:+D
TRK  40|CB67 20D5 2105 443E C396 20CD 772A 0644|#G #!.D>## #W#.D
 00  50|F1E3 FBC9 CDF8 01C3 3040 5F455F45 5F45|######.#0@_E_E_E
TRU  60|5F45 5F45 5F45 5F45 5F45 5F45 5F45 5F45|_E_E_E_E_E_E_E_E
 00  70|5F45 1129 35D5 3E08 CD37 453E 09CD 3745|_E.)5#>.#7E>.#7E
SEC  80|3E0A CD37 453E DBCD 3745 2188 4234 7EE6|>.#7E>.#7E!#B4~#
 09  90|0707 6F26 4522 6245 5E2C 5605 DDE1 EB5E|..O&E"BE^,V####^
STD  A0|2356 EBE9 D13A 6245 0F11 5F45 FE0C D007|#V###:BE.._E#.#.
0DD  B0|6F26 45F3 732C 72FB C95E 4521 0000 5E23|O&E#S,R##^E!..^#
     C0|56EB D118 EEFD 7E03 EE40 FD77 0301 0924|V##.##~.#@#W...$
     D0|CB77 2803 0111 45FD 7107 FD70 0818 1878|#W(...E#Q.#P...X
     E0|A7C8 FE07 CA05 4602 1C46 FE06 2853 3D28|###.##F#.F#.(S=(
+00  F0|11FD 3405 FE04 0658 286B FD36 0500 0601|.#4.#..X(K#6....
```

This display contains a wealth of information. The leftmost column contains information about the display. HEX refers to the current modification mode base (see below). DRV and the number directly beneath it refer to the drive number just accessed. TRK and the number beneath it refer to the current track number, TRU and the number beneath it refer to the actual track number written on the disk, and SEC and the number beneath it is the sector being displayed on the screen.

Below the sector number is a three-letter code which identifies the particular data address mark written on the disk when the it was formatted. Super Utility Plus will identify four types of data address marks: STD, or standard, DDT, or deleted data, RPT or "read-protected", and UDF, or user-defined. The terms are those used by the manufacturer of the floppy disk controller chip and are not necessarily meaningful except insofar as they differentiate one type of address mark from another. TRSDOS uses a different type of data address mark for the directory track than

for all the other tracks on a diskette and this is the major difference you need to be concerned about.

The floppy disk controller chip used by the Model III can only recognize two of the four types of DAMs: STD and RPT. The table below indicates the equivalent types for Mod I and Mod III as used in Super Utility Plus:

Model I STD & RPT = Model III STD
Model I DDT & UDF = Model III RPT
Model III STD = Model I STD
Model III RPT = Model I UDF

Below the data address mark identifier the density of the diskette being examined will be displayed. This will normally be 0SD for single-density Model I diskettes and 0DD for double density diskettes. Super Utility Plus has the capability to recognize the density of a diskette it is reading and switch between single and double density if you specify the special "#" symbol in front of the drive number. The first character of the density ID string is "0" or "1" depending on which side of the disk is being read.

At the very bottom of the leftmost column of information you will see +00. This indicator is used by a special feature of Super Utility Plus called decryption which will be explained in a later section.

The next column gives you the relative byte numbers (in hexadecimal) of the data immediately to the right of the graphics border. "Relative byte" position simply means the position of a particular byte with respect to the first position, designated 00. Each row of the display shows 16 bytes with their values in hexadecimal format. Each group of four hexadecimal digits represents two bytes.

The data read from the disk sector is displayed between the two graphics borders, and to the right of each row is the ASCII representation of these bytes. Values which do not represent displayable ASCII characters (below ASCII 32) are displayed as periods. Also, if you are using a machine not equipped with lower case, any lower case alphabetic character will be shown in upper case; however the HEX value will always be accurate.

I.1 Paging

You can use the arrow keys to page across sectors. The right-arrow key will page forward one sector, the left-arrow key will page back one sector. Pressing the up-arrow key will take you to the same sector on the next higher-numbered TRACK, while pressing the down-arrow key will take you to the same sector on the next lower-numbered track (unless you are already at the lowest). In addition, there are a number of other paging controls, given in the table below.

TABLE 2-1 - PAGING CONTROLS

KEY	ACTION
Right arrow	pages one sector higher (or to the lowest sector of the next track if the current one is the last for this track)
Left arrow	pages one sector lower (or to the highest sector of the preceding track if the current one is the lowest for this track)
Up arrow	pages one track higher, same sector
Down arrow	pages one track lower, same sector
SHIFT-Right arrow	pages one sector higher but will not leave current track (see note)
SHIFT-left arrow	pages one sector lower but will not leave current track (see note)
SHIFT-)	displays highest sector on current track
SHIFT-(	displays lowest sector on current track
SHIFT-Up arrow	pages to the same sector on the highest track (defined in the configuration tables)
SHIFT-Down arrow	pages to the same sector on the lowest track
R	displays Track 0, Sector 0 (Sector 1 on TRSDOS 1.3 and 2.7DD.
T	displays prompt for new Track, Sector
S	displays prompt for new sector
CLEAR	displays prompt for new Drive, Track, Sector
. or >	pages to the next higher VALID sector
, or <	pages to the next lower VALID sector
number keys **0-9**	displays the correspondingly numbered sector on the current track
BREAK	returns you to the ZAP Utilities menu
SHIFT-BREAK	returns you to the main menu
@	enables DECRYPT mode (see below)
H,D,B,O,Q,A	Sets modification mode BASE to hexadecimal, decimal, binary, octal (O and Q are equivalent) or ASCII.
M	Enters modification mode.

NOTE 1: Shift-right arrow and shift-left arrow may occasionally display sector numbers that are beyond the expected range. This will happen when Super Utility tries to read non-standard sectors and is not a cause for concern.

NOTE 2: Depending on the case setting, you may need to use either @ or shift-@ to enter decryption mode.

As you can see, the ZAP utility provides you with tremendous flexibility in searching through a disk. You can view any sector on the disk with relative ease.

One of the most powerful commands in Super Utility Plus is the L or LAST command. When you are viewing a sector and exit to the menu, either deliberately or accidentally, you can always return to the sector you were viewing by calling up the drive, track and sector prompt line

and simply entering d L where "d" is the drive number and "L" takes the place of the track and sector values (the space is mandatory). You will immediately be returned to the last sector you were viewing.

The value of LAST is updated by a number of routines. When performing a sector comparison, the last sector in which a mismatch was found will update LAST. Thus if you were comparing two disks and wanted to scan the mismatch, you would merely press CLEAR to stop the routine, request DISPLAY DISK SECTORS, and when the prompt for drive track and sector comes up, enter the drive number followed by L and you will be taken to the sector where the mismatch occurred.

When performing a string search (see below), the last sector in which a string match was found will also update LAST.

I.2 Modifying the contents of a disk sector

The ZAP utility gives you the powerful ability to directly modify the contents of a disk sector. This should only be undertaken with great care, since careless modification can make a file or a disk totally useless for other purposes.

Display the sector you wish to modify using the instructions given above. Then look at the leftmost column on your screen. On the second row you will see HEX displayed. Super Utility Plus' ZAP utility gives you the ability to modify the contents of a disk sector in either Hexadecimal, Decimal, Octal, Binary, or ASCII form. The default is hexadecimal, and this is what the HEX means. Press D on your keyboard, and you will see it change to DEC. Now press A, and ASC will be displayed, meaning that ASCII modification is enabled. The keys for switching the modification case are H for hexadecimal, O or Q for octal, D for decimal, B for binary, and A for ASCII.

Press H again. Now press M. You will see a pair of flashing cursors appear in the data portion of the display, one in the hexadecimal portion, and another in the corresponding position in the ASCII display to the right.

The cursor may be positioned anywhere in the data portion using the arrow keys. The right arrow moves the cursor one byte to the right, the left arrow moves it one byte to the left. The up and down arrows move the cursor up and down the rows. By holding down the SHIFT key and pressing one of the arrow keys, you can position to the far ends of the display. For example, pressing SHIFT and the right arrow key will position the cursor to the rightmost byte on that row. Pressing SHIFT and the down-arrow key will move the cursors to the bottom row of the display. The current position of the cursor, in relative byte format, will be displayed at the upper left of the screen. Pressing the CLEAR key will return the cursor to the upper left position (byte 00). The other positioning keys available in MODIFICATION MODE are given in the table below.

TABLE 2-2 - Modification Controls

Key	Action
H,D,B,O,Q,A	selects modification mode base (Hexadecimal, Decimal, Binary, Octal or ASCII). This should be done before entering Modification mode.
SHIFT-ENTER	reset modification mode (permits reselection of H, D, O, Q, B, or A).
Right Arrow	moves cursor one byte to the right.
Left Arrow	moves cursor one byte to the left.
Up Arrow	moves cursor up one row.
Down Arrow	moves cursor down one row.
SHIFT-Right Arrow	moves cursor to last byte on row.
SHIFT-Left Arrow	moves cursor to leftmost byte on row.
SHIFT-Up Arrow	moves cursor to top row.
SHIFT-Down Arrow	moves cursor to bottom row.
CLEAR or S	returns cursor to relative byte 00 (leftmost byte of top row).
Q	moves cursor to the last byte of the bottom row.
G + relative byte no.	moves cursor to the specified relative byte location (not active if in ASCII modification mode).
L + numeric input	moves cursor to the next occurrence of specified numeric input, i.e., "L3F" will place cursor on the next occurrence of 3F in that sector (not active in ASCII modification mode). The numeric input MUST be in the current modification base.
+ and numeric input	moves the cursor that many bytes forward from its current position.
- and numeric input	moves the cursor back that many bytes from its current position.
>	insert data at current cursor position and move the rest of the data one byte to the right.
<	delete the data beneath the cursor and shift the rest of the data one byte to the left.
P + numeric input	copy the byte beneath the cursor the specified number of times to the right. The numeric input must be in the current modification base.
Z	zeroes out display and holding buffer from the current cursor position on.
ENTER	terminate modification mode.
BREAK	abort modification mode, return to ZAP utilities menu.
SHIFT-BREAK	abort modification mode, return to main Super Utility Plus menu.

Note that the keys being used in modification mode are the same as those used in PAGING mode, but have entirely different actions.

Extreme care should be taken when using the modification mode controls, as the result may not be what you expect.

By typing a valid key (that is, a key valid for the current modification mode you are in), you will enter that value into the display at the current cursor location. For example,

```
     00 FE11 3ED0 D3F0 2102 0022 EA43 AF32
HEX 10 CD3E 43FE 0128 0CFE 0220 E7CD 3E43
```

In this case the cursor is under F0 on the first row. If you now enter a valid numeric digit, say 7, the first row would look like this:

```
00 FE11 3ED0 D3__ 2102 0022 EA43 AF32
```

Note that the cursor has changed, and the byte you are modifying has temporarily disappeared. Super Utility Plus is reminding you that you need to enter another hex digit to complete the modification of that byte. Now type D. The row now looks like this:

```
00 FE11 3ED0 D37D2102 0022 EA43 AF32
```

F0 has been replaced with 7D, and the cursor has moved one byte over.

The cursor will remain in its changed state until you have entered the necessary number of digits to enter a new value. This will vary depending on the modification base you are using. For example, if you are in BINARY modification, you will normally have to type in 8 binary digits before the new value appears on the display. If you wish to terminate the input early before entering all the digits, simply press ENTER. The binary value will be padded on the left with the requisite number of zeroes so as to evaluate properly. To abort an entry simply type in an invalid digit and press ENTER.

No matter what modification base you use, the display will always appear in hexadecimal, with the corresponding ASCII equivalents on the right.

In ASCII modification, all keys on the keyboard except for the arrows, the greater-than and less-than symbols (> and <), BREAK, CLEAR and ENTER are valid for input.

Pressing ENTER terminates modification mode. You will now be presented with the prompt:

U>pdate, R>eturn to modify, or C>ancel ?

The modified sector will be written out to disk when you press ENTER or explicitly choose the Update option (type U and press ENTER). The sector data will be written back out to disk, and then re-displayed. Remember, UPDATE is the default, and hitting ENTER alone will select it.

If you enter R, you will be returned to the sector display, and the disk will not be updated. You can continue making modifications. The display will contain the modifications you have made up to that point.

Pressing C cancels the modification session. All your changes will be canceled, and the sector will be re-displayed with its original data intact.

I.2.1 Using the special modification controls

The greater-than key (>) permits you to insert data into the display without having to retype what is already there. All the other bytes will be shifted one byte to the right, with the last one being lost. Conversely, the less-than key (<) will delete the byte beneath the cursor and shift all other bytes one place to the left to fill in the gap. The last byte, which was shifted OUT when you pressed ">", will NOT be brought back in.

While in a numeric (not ASCII) modification mode, you can use a number of other keys to make changes to the displayed sector data with ease.

The G key allows you to move directly to any position in the displayed sector without having to use the arrow keys. Simply type G followed by the relative byte position you want to go to, and the cursor will be placed on that position. The number following G will be interpreted in the current modification base (except when the modification base is ASCII, of course). For example,

```
    00 00FE 11CD C901 FD21 0000 3A02 4257 1EE4
HEX 10 0651 CDAB 4220 5801 E650 FE50 2056 2A16
                    .
                    .
                    .
    60 1823 10F6 18CA CD7E 426F CD7E 4267 C921
```

The cursor is positioned at relative byte 5, which holds the byte 01. We want to position the cursor to relative byte 67 hex, which is 7E. To do this, just type G67. The cursor has now been positioned where we want it:

```
    00 00FE 11CD C901 FD21 0000 3A02 4257 1EE4
HEX 10 0651 CDAB 4220 5801 E650 FE50 2056 2A16
                    .
                    .
                    .
    60 1823 10F6 18CA CD7E 426F CD7E 4267 C921
```

The LAST variable is also updated by this command, so that if you reposition the cursor and later wish to return to the original byte, simply typing GL will get you there. The use of the L will allow you to reposition the cursor with a minimum of keystrokes.

The + and - keys operate in a similar fashion. Typing a + followed by a number in the current modification base will move the cursor forward that many bytes from its current position. If the number entered was such that it would cause the cursor to leave the display, the cursor will be positioned at the last byte of the display. Typing a - followed by a number will move the cursor back that many bytes from its current position.

Another handy control is L. This permits you to locate a particular byte on the display and position the cursor over it. In the example above, suppose the cursor was at relative byte 00,

and you wanted to locate the byte whose value was 21 hex. Type L21. The cursor will immediately be positioned over relative byte 7. Now press L21 again. The cursor is now positioned at relative byte 6F hex, which is also 21H! Remember however, that like the G command, the number which follows L is also interpreted in the current modification base.

Instead of repeatedly pressing L21, however, you could also enter LL. The second L refers to the "last value entered" and would produce the same results. This is also available in the G command.

For quick positioning to the start and end of the display, the S and Q keys are available. The S key will move the cursor back to the first byte on the display, while the Q will move it to the last byte on the display.

Sometimes you want to duplicate a certain byte a specific number of times. You can do this very easily with the P key. In the example above, suppose you wanted to replace the entire first row with 00. Position the cursor over the first byte, which is already 00, press P, and 0F. The rest of the row, 15 bytes all in all, have been replaced with 00 (0F hex =15 decimal). As with the G and L commands, if your modification base was decimal, you would have had to enter P15, since the number would be taken in the current base.

The S, Q, G, L, +, - and P commands are obviously not usable when you are in ASCII modify mode, because in this mode all keys except for the four arrow keys, >, <, BREAK and ENTER are valid input.

For zeroing out the display and the contents of the holding buffer, you may press Z. This will immediately remove all data from the holding buffer and replace them with 00 bytes starting at the current cursor position.

I.3 Bit-shift operations

While you are in paging mode, you may perform various bit-shift and bit-rotate operations on the displayed data. These operations are analogous to those performed by the Z-80 assembly language instructions RLCA, RRCA, SLA, and SRL. They involve shifting the bits of an 8-bit byte a specified number of times to the left or right to form a new value. In shift operations, the bits at the far end are usually lost, while in rotate instructions, the bits at the far end are "rotated" into the opposite end. For example, take the binary representation of the value 85 (decimal):

0 1 0 1 0 1 0 1

To shift this byte 1 place to the right would yield the value

0 0 1 0 1 0 1 0

or 42 decimal. The trailing 1 bit has been lost. Shifting the original number left one place would give you

1 0 1 0 1 0 1 0

or 180 decimal. Shifting it left two places yields the binary value

0 1 0 1 0 1 0 0

which is 94 decimal.

Rotate operations involve moving the bit that would normally be lost at the far end into the opposite end of the bit string. Using our example, 0 1 0 1 0 1 0 1, If we rotate this value right one bit, the result would be

1 0 1 0 1 0 1 0

or 180 decimal. Rotating it left one bit yields

1 0 1 0 1 0 1 0

also 180 decimal! Rotating it left two bits gives us

0 1 0 1 0 1 0 1

bringing us right back where we started, at 85 decimal. Other numbers would give other results, of course.

Shift and rotate instructions can be carried out for any number of places you wish, up to 7 (the width of one byte). In Super Utility Plus, if you wish to execute a shift or rotate instruction, you must first be in the paging (not modification) mode. The shift or rotate operations are carried out on ALL the displayed data bytes simultaneously. To see how this works, first display a sector and make sure you are in the paging mode. Now press @. Toward the bottom of the display at the left side, you will see the prompt

DCR

with a line beneath it (DCR stands for "decryption" -- this routine can be used to investigate sectors which may have been encrypted either by bit shift, logical or increment/decrement operations).

To execute a Rotate Right operation, press RR followed by the number of places you want to rotate each byte on the display. You will see the bytes change as soon as you hit ENTER. To execute a Rotate Left operation, type RL followed by the number of places to rotate and press ENTER. SR followed by a number executes a shift right operation, and SL executes a shift left operation.

In addition, this particular routine is capable of carrying out a variety of logical operations on the displayed data. The commands are listed below. Each command must be

followed by a numeric input (the base must be properly identified by appending H, B, O, or Q – decimal is the default -- to the number) and ENTER:

A	AND the displayed bytes with given input
O	OR the displayed bytes with given input
X	XOR displayed bytes with given input
+	ADD the given input to each byte and display the result, modulo 256
-	SUBTRACT the given input from each byte and display the result modulo 256

You can also command the computer to increment or decrement the display automatically any given number of times. This may be useful if you wish to see if any ASCII words in the displayed sector have been encrypted by adding or subtracting a constant value to its ASCII value. It may be necessary to type "@" again to re-enter decryption mode as some of the commands automatically take you back to normal paging mode.

While executing this operation, you can further tell the computer whether to update the ASCII side of the display only (the default), by entering "*****" as the first character in response to the DCR prompt, or both the HEX and ASCII sides by entering "**:**".

You can increment or decrement the display one bit at a time in decryption mode also. You would enter decryption mode as usual, and select whether you want the HEX side only modified, the ASCII side only modified, or both. Then you would enter an up-arrow if you wanted an increment operation, or a down-arrow if you wanted a decrement operation (you cannot do both simultaneously). Finally you can control the speed at which the computer updates the display.

Suppose you wanted to increment the ASCII side of the display and view the results. In response to the DCR prompt, you would enter ***** followed by an up-arrow (for incrementing) and finally a decimal number from 1 to 255 for speed (1 is the fastest, 255 is the slowest). As soon as you press ENTER you will see the ASCII side of the display begin to change.

You may stop this automatic operation anytime by pressing @, or you can pause it momentarily by pressing the spacebar. The display will stop updating when all the bytes reach the maximum value, FFH or 255 decimal. Entering a "**!**" makes the changes permanent and must be the first character of a new DCR prompt line.

This operation can be used to encrypt your own messages very easily. Suppose you had a sector of code which, among other things, contained the ASCII string "MYPROGRAM" and you wanted to hide the string by XOR'ing each byte with 88H. Enter decryption mode and make note of where the string MYPROGRAM starts in the sector. Then XOR the entire sector with 88H. Now enter modification mode and place the cursor at the location where the string starts, and type in the string MYPROGRAM normally. Exit modification mode and go into decryption mode again. Now once again, XOR the entire sector with 88H. The rest of the sector will now be back to normal, but your string "MYPROGRAM" will now become encrypted! Finally, use the "**!**" symbol to make the changes permanent and write the sector out to disk.

The +00 on the bottom left of the display is updated to show the amount the data bits are shifted, rotated, logically operated on, or incremented or decremented by a DCR operation. The symbol preceding the two numbers will change to reflect the current operation, in accordance with the table above. Pressing ENTER in response to the DCR prompt will reset this number to 00.

I.4 Error Recovery

Should any I/O errors occur while Super Utility Plus is attempting to read or write to a disk, a message describing the type of error encountered will be displayed, for example,

Sector NOT FOUND!

or

DATA CRC ERROR!

and you will be given an option to retry the I/O operation with the prompt,

R>etry, S>kip, C>ontinuous, N>onstop or Q>uit ?

for as many times as necessary. You may press R (or just ENTER) to retry the I/O operation once. If the error occurred as a result of some momentary condition, this is usually sufficient to correct the situation. However, if the error continues to appear, you may want to type C for continuous retries. This will force Super Utility Plus to retry the I/O operation until it gets it right, or CLEAR is pressed. Pressing CLEAR will restore the retry prompt. Also, break and shift-BREAK may be pressed from the keyboard to abort the operation and return you back to a Super Utility Plus menu. Pressing N will produce the same results as Continuous, but in addition will automatically place Super Utility in continuous mode on subsequent errors.

Pressing S will immediately bring up the sector display. Depending on the error encountered, this may or may not contain any data. If the sector could not be read at all, the display will contain all 00 bytes (the buffer is zeroed out before a sector read is performed). If Super Utility Plus was successful in partially reading the display, some data will be present in the display. However, you should not assume that what is shown on the display is an accurate representation of the data. If the error was simply a data CRC error, then the data may be intact. However, some bytes of data could also be garbled. You should examine the display closely and come to your own determination of whether the data displayed is intact or at least repairable. If you decide the data is intact, or have carried out the necessary repairs, then writing the sector back out to disk will automatically correct any CRC errors which may exist as long as the media is physically intact.

Pressing Q will abort the operation in progress and return you immediately to the ZAP Utilities menu.

II. VERIFY SECTORS

This option allows you to scan all or part of a disk for conditions which would produce I/O errors. This routine does not check the data, but rather whether or not sectors are readable. You will be prompted for the drive, track and sector to verify, and the number of sectors to verify. The program will then proceed to read these sectors and report any errors which it encounters.

When errors are encountered, the program will display an error message followed by the prompt to R>etry, S>kip, C>ontinuous, N>onstop, or or Q>uit. If you press R, the program will attempt to read that sector again. If it succeeds, it will continue. If it does not succeed, it will re-display the error message.

Pressing C or N for continuous re-try will force the program to re-read the bad sector UNTIL it gets it right. If it cannot read the bad sector at all, you may exit by pressing CLEAR, BREAK, or SHIFT-BREAK depending on where you wanted to go next.

Skip tells the program to skip the bad sector and continue. The program will keep an internal count of the bad sectors encountered and will report the total number of bad sectors at the end of the operation.

Quit simply brings the verification routine to an end at that point. You will be brought back to the ZAP utilities menu.

III. COMPARE SECTORS

This utility permits you to compare the contents of two different sectors and is useful if you want to verify that a backup operation made an exact copy. This routine will also check for data address mark mismatches.

When this option is selected, you will be prompted for the drive, track and starting sector number of the source disk. The program will then ask you for the number of sectors to compare (default is to the end of disk). When you have entered this number, you will then be prompted for the drive, track and starting sector location on the destination disk for the comparison operation.

After this, you will be asked, "Prompt for disk mounts?" If you answer Y (yes), you will be told when to swap disks. This is especially useful when you have only one drive in which to do the comparison. The number of disk swaps will be determined by the amount of memory available to Super Utility Plus for buffer space. On a two-drive comparison, of course, no disk swaps are necessary. However, you can use the disk mount prompts on a two drive system to check the readability of a disk on two different drives, should you suspect that the drive hardware is at fault.

IV. COPY SECTORS

This option will allow you to copy sectors from one disk to another, or from one location on the disk to another. Only full sectors are copied. When this option is selected, you will be prompted for the drive, track and starting sector of the source disk, and the number of sectors to be copied. Then you will be asked for the drive, track and starting sector number of the destination disk. As with the COMPARE SECTORS option, you will then be asked if you want to be prompted for disk mounts. If you reply Yes, you will be told when to swap disks. If your source and destination drives are the same, the disks you swap must be the same disk type.

The data will then be copied over to the new locations, leaving the original locations intact. The track and sector id fields will not be copied, but the correct DAMs and the actual contents of the sectors will be transferred.

Note that this operation is not reflected in the diskette's directory. You may use this routine to copy a file from one location to the other, but your directory will not show the file in its new location.

V. COPY SECTOR DATA

This option will allow you to copy partial sector data onto a new sector. You will be prompted for the drive, track and sector of the source disk, the relative byte number within that sector where the copy is to begin, and the number of bytes you wish copied. Then you will be prompted for the drive, track and sector of the destination disk, along with the starting byte position for the copy, and whether or not you want disk mount prompts. The copy will then proceed. You will be advised upon completion of the routine whether any disk I/O errors occurred during the copy.

This routine will allow you to copy from 1 to 65535 bytes of information to a new location of your choice.

VI. ZERO SECTORS

This option will totally remove the data from the specified sectors, setting the entire contents of the sectors to 00 and resetting the data address marks to STD (RPT on Model III TRSDOS). You will be prompted for the drive, track and starting sector number for the operation, along with the number of sectors to zero. BE CAREFUL WITH THIS ROUTINE! Upon pressing the ENTER key the operation will immediately be carried out, and there is ABSOLUTELY NO CHANCE of recovering the data once a sector hasbeen zeroed out!

Do not use this routine to zero out directory sectors, as the directory sectors will be written out with the wrong DAMs. Use the "Zero Unused Entries" routine in the PURGE UTILITIES menu.

VII. REVERSE SECTOR DATA

This routine simply takes the data of a specified sector and reverses it, so that the byte that as in relative position 00 is now in relative position FFH and so on. This routine may be useful in creating disk protection schemes for machine language programs and data. If multiple sectors are specified then each sector will be worked on independently. That is, the reversal of sector data takes place entirely within each individual sector.

VIII. EXCHANGE SECTORS

This routine will exchange the data contained in one or more sectors with the data in another set of sectors. You will be prompted for the source drive, track and starting sector number, along with the number of sectors to exchange. Then you will be prompted for the destination drive, track and starting sector number and whether or not you want to be prompted for disk mounts. Upon pressing the ENTER key the data in each sector on the source disk will be exchanged with the corresponding sector on the destination disk. Upon completion of the operation you will be advised of any disk I/O errors that may have occurred.

IX. STRING SEARCH

Up to an entire disk may be searched for a given ASCII string, BYTE list or WORD list (a word equals two bytes, a total of 16 bits) using this routine. Additionally you may optionally specify a replacement string which the routine will insert in place of the target string whenever the target string is found.

You will be prompted for the drive, track and starting sector for the search, and the number of sectors to search. Next, you will be prompted to enter the search string, and lastly, the replacement string (if you do not wish to perform any replacement, merely hit ENTER in response to this prompt). If the length of the replacement string is shorter than that of the search string, only that portion of the search string which corresponds to the length of the replacement string will be replaced. This will permit you, for example, to search for all occurrences of "John Brown" and change them to "Carl Brown" by simply giving "Carl" for a replacement string.

If the replacement string is longer than the search string, it will be truncated to the length of the search string. The disk will then be searched, and upon completion, the location of each match and the total number of matches found will be displayed.

The search string must be contained wholly within a sector; that is, part of it cannot reside in one sector with the rest in the next sector. If this is the case, the routine will not find it. If you suspect that this may be the case, try performing a search on only a portion of the string.

To search for an ASCII string, simply enter it when prompted. For example, if you wanted to search for an occurrence of the name "Jim", you would enter,

"Jim"

in response to the prompt. Enclosing the word in double quotes means that you want the search to match upper and lower case EXACTLY. If you wanted to find occurrences of "Jim" and "JIM" you would enclose the search string in single quotes. Single quotes tells Super Utility Plus to search for the string in a case-independent fashion, that is, upper and lower case differences will not matter.

If you wanted to find all occurrences of "Jim" and "Tim" on the disk, you could enter,

"?im"

in response to the prompt. The "?" is a wild-card character, meaning, in effect, "I don't care what's in this position, match it with whatever's there and show it to me anyway."

To search out BYTE values, when you are prompted for the search string, enter a series of values in the range 0 to 255 in any valid numeric base separated by commas or spaces. For example,

10H,13H,CDH

which in decimal would be

16,19,205

Note that each element of the reply is separated from the rest by a comma (they may also be separated by spaces).

The word search is slightly different. This is normally used to locate sixteen-bit address references in a machine language program. The Z-80 stores such addresses in reverse order, that is, the least significant byte first, followed by the most significant byte. Thus, the address reference 7F42H would be stored in a machine language program as 42 7F.

The word search routine assumes this to be the case, so that when you enter a two-byte value, it will automatically reverse them before starting the search. Thus, if you wanted to find the two bytes 7F 42 in that exact order, you must enter them as 427FH or, alternatively, use the byte search mode to look for the byte pair 7FH,42H. Note, however, that this will not work when you are looking for byte pairs that end in 00, for example 3300H. The reason for this is that when Super Utility Plus evaluates your ASCII input into binary, it will skip over leading zeroes. Thus if you were to enter 0033H, it would be evaluated as 33H. For these cases you must use a BYTE search.

The word search is specified by entering 16-bit values on the prompt line in any valid numeric base, although due to the reversal which takes place it is probably easier to use hexadecimal (easier for YOU, not the computer). The values should be separated from each other by commas or spaces. For example,

7F42H,CD30H,402DH

Notice how the bytes are grouped in pairs and each pair is separated by commas. This will result in a search for the bytes (in hexadecimal) 42 7F 30 CD 2D 40 in that order.

If you wish to specify a replacement string, you must make sure that the replacement string is the same length as or shorter than the search string, or the replacement string will be altered (truncated to the right length). You could specify an ASCII search string and then replace it with a BYTE or word string, but it is usually easier to enter the search and replacement strings in the same format.

However, it is also possible to mix different modes in a single search or replacement string. That is, ASCII strings, byte, and words can all be placed on the same line. For example,

"Test",42H,79E0H,'disk'

is a perfectly valid string search specification.

Finally, note that if "@" is the first character of a search string, then it will be assumed that the target string has been encrypted according to the current setting of the decryption routine. The search string will be encrypted similarly before the search is conducted. If a byte string is given along with "@" then the byte string will be created first, then the string encrypted.

X. SECTOR SEARCH

This option is useful in searching out duplicate sectors on one or more disks. You will first be prompted for the drive, track and sector number of the sector that is to form the search template. Then you will be asked for the drive, track, and starting sector number for the search, followed by the number of sectors to search. When the routine completes, it will display the total number of matches that it found, and their locations on the target disk.

XI. READ ID ADDRESS MARKS

This routine will examine a disk and identify the track and sector data written on the target disk. This will identify any false or non-standard sectors on the disk and help you determine just how a disk was formatted. You will be prompted for the target drive to examine, and the program will start reading track 0 and present the information on the display. The information will scroll by very rapidly, but can be stopped at any time by pressing the spacebar and resumed by pressing ENTER. It may also be exited by pressing the BREAK key. It will advance up one track whenever you press the UP-ARROW and down one track when you press the DOWN-ARROW. Additionally, you may proceed directly to the highest formatted track on the disk by pressing SHIFT-Up arrow, or to the lowest track on the disk by pressing SHIFT-Down arrow.

You may also press S to force the routine to read only single density sectors, or D to make it read only double-density sectors. In addition, when dealing with double-sided diskettes, pressing F or ' (apostrophe) will force a read of the front side, while pressing B or " (double quote) will cause the back side of the disk to be scanned. These keys are active while the display is scrolling. Note that these keys will alter the configuration settings for the affected drives.

If the routine encounters an unformatted track, it will display the error message "ID ADDRESS READ ERROR" or "TRACK NOT FORMATTED."

Initially, the routine will display seven columns of information, as follows:

```
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. ##  SUPER-UTILITY +  ##  VERSION 3.1A  ##  BY: KIM WATT  ##  .
.       (C)(P) 1983 BREEZE/QSD, INC.  --  DALLAS, TEXAS        .
# SOURCE  TRACK HEAD SECTOR LENGTH CRC1  CRC2  CKCRC IBM DATA  #
. :0D'= 01  01    00   06     01    15H   2FH                   .
. :0D'= 01  01    00   02     01    D9H   EBH                   .
. :0D'= 01  01    00   04     01    73H   4DH                   .
. :0D'= 01  01    00   17     01    8FH   CBH                   .
. :0D'= 01  01    00   07     01    26H   1EH                   .
. :0D'= 01  01    00   03     01    EAH   DAH                   .
. :0D'= 01  01    00   16     01    BCH   FAH    YY   Y   STD   .
. :0D'= 01  01    00   12     01    FAH   E4H    YY   Y   STD   .
. :0D'= 01  01    00   14     01    9CH   86H    YY   Y   STD   .
. :0D'= 01  01    00   16     01    BCH   FAH    YY   Y   STD   .
. :0D'= 01  01    00   01    01    BCH   B8H    YY   Y   STD   .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
```

The leftmost column is labeled SOURCE. This gives you information as to which drive you are reading, and the density of the disk in that drive. For example, :1D= 01 indicates that the disk in drive 1 is double density, and the read/write head is positioned over physical track 1.

The next column is labeled TRACK, and this is the track number which is actually recorded on the disk. Some protected disks use non-standard track numbers, and if the disk you are reading is one of these, the track column would not necessarily agree with the actual track number to its left.

The third column, HEAD, is the head number actually recorded on the disk. In double-sided disk drives, the head number could indicate which side of the disk is being read.

The fourth column is SECTOR. These are the sector numbers as recorded on the disk, and will not appear in any special order. Sectors which are physically located side by side on a disk will not be numbered consecutively. Also, some protected disks will assign false sector numbers to prevent them from being read by standard operating systems.

The next column, labeled LENGTH, is a coded indication of the amount of data contained in the sector. In the IBM convention which TRSDOS and most TRSDOS-compatible systems use, a length of 00 means 128 bytes per sector, and a length of 01 means 256 bytes per sector. If you

are scanning a standard disk, this should display 01. In the Model III and double-density Model I disks, the only acceptable values for LENGTH are 00 through 03, corresponding to lengths of 128, 256, 512, and 1024 bytes respectively.

If the disk is not formatted with the IBM conventions, then a 00 length byte indicates 1024 bytes per sector, and a non-zero length byte multiplied by 16 will yield the actual number of bytes in the sector. By this convention, a non-IBM length byte of 10H (16 decimal) would be equivalent to an IBM length byte of 01. Non-IBM conventions are possible on Model I single density disks only, as the double-density controller used by the Model III does not use the IBM/NBM indicator.

Note that the length code indicates the number of bytes as recorded on the disk's data fields; it does not necessarily mean that there are that many bytes actually present in the sector (again, this may be used by disk protection schemes to lay a false trail).

The next two columns are labeled CRC1 and CRC2. CRC stands for "Cyclic Redundancy Check," and the two bytes are the result of a lightning-fast calculation made by the floppy disk controller chip on the sector data. When a sector is read back in by the computer, the disk controller chip recalculates the CRC's and compares them with those recorded on the disk. If the bytes recorded on the disk fail to match those recalculated by the FDC, it signals an error condition, which normally appears on your DOS display as "CRC error" or "Parity error during read."

Planting false CRC bytes is another favorite sport of disk protection experts. They will override the FDC's calculation and insert their own bytes in place of the true CRC's, thus forcing standard operating systems to ALWAYS signal an error when attempts are made to read their disks.

If you press the X key while the display is scrolling, you will produce an additional three columns of information on the screen. The first, immediately to the right of the CRC2 column, is labeled CKCRC. This is the result of Super Utility Plus's reading of the sector, which forces the FDC to report back the CRC bytes for the sector. The result is indicated by two letters. The letter to the left pertains to the ID field's CRC, and the letter to the right pertains to the data field CRC. A "Y" indicates that that CRC byte on the disk is correct, and an "N" means that there is a discrepancy between the recorded CRC byte and the actual byte arrived at by the FDC on the basis of the sector data. If the ID field CRC is bad, then the data field CRC check will appear as "*" since it is impossible to check the data.

The next column, labeled "IBM," is an indicator of whether or not the track was formatted using IBM conventions, and will help you in determining the meaning of the LENGTH code. Once again, this is relevant only for Model I single density disks, as the Model III FDC does not use the IBM/NBM byte.

Finally, the DATA column tells you what data address mark was used in formatting. This will display STD for standard, DDT for deleted data, RPT for "read protected," and UDF for

user-defined. As we said before, the names do not really mean anything and are used simply to differentiate one type of address mark from another.

The Model I floppy disk controller chip is capable of reading and writing all four types of data address marks, or DAM's. However the double-density controller of the Model III can only produce and recognize two types of data address marks: STD and RPT, even if it is reading or writing a single-density disk.

NOTE: when X is pressed to bring up the additional three columns of display, the keyboard will tend to become sluggish as it will not be scanned as often as when the normal display is scrolling past. Thus if you wish to use the arrow keys to move to another track, for example, it is necessary to hold these keys down for a couple of seconds before the desired effect is achieved.

XII. ALTER DATA ADDRESS MARKS

This routine will allow you to alter the data address marks on a diskette to something other than the standard marks used by your disk operating systems, and may be a good way to produce your own "protected" diskettes. You will be prompted for the drive, track and starting sector for the alteration, and the number of sectors to alter. Finally, the program will prompt you for the type of data address mark you wish to use. Answer S for standard, R for read-protected, D for deleted data, and U for user-defined. If you are using a Model III or a Model I in double density, please remember that you can only use the S and R types.

Again, care should be taken when using this option, as you can wind up with a disk that no operating system can read correctly.

CHAPTER 3 - PURGE UTILITIES

Bring up the main menu and press 2. This will display the Purge Utilities menu:

```
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. ##  SUPER-UTILITY +  ##  VERSION 3.1A  ##  BY: KIM WATT  ##  .
.       (C)(P) 1983 BREEZE/QSD, INC.  --  DALLAS, TEXAS        .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. __ PURGE UTILITIES __                                                .
.                                                                      .
.  1 - KILL SELECTED FILES       6 - ZERO UNUSED ENTRIES               .
.  2 - KILL BY CATEGORY          7 - ZERO UNUSED GRANS                 .
.  3 - REMOVE SYSTEM FILES       8 - CHANGE DISK NAME                  .
.  4 - REMOVE PASSWORD           9 - CHANGE FILE PARAMS                .
.  5 - DISK DIRECTORY           10 - CHECK DIRECTORY                   .
.                                                                      .
. CHOICE? #__                                                          .
.                                                                      .
.                                                                      .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
```

The Purge Utilities operate mainly on the directory of your system or data diskettes. They provide a very fast and very convenient way of killing (and recovering!) files from a disk, and even include facilities for checking your disk directories for errors, and removing all traces of a file from a disk by zeroing out the sectors which it occupied. This last is very useful in maintaining security.

I. KILL SELECTED FILES

The first choice on the purge menu is KILL SELECTED FILES. This will permit you to scan the directory of a diskette and "tag" certain files for removal. The files are shown on the screen, along with a cursor. Active files are shown surrounded by left and right arrows (left and right square brackets on the Model III) and the non-active file entries (killed files) are shown surrounded by graphics blocks.

Note that the KILL command of TRSDOS removes all traces of a file from the directory, so that you may not see any non-active file entries on the display. Super Utility Plus's purge routines do NOT remove all traces of a killed file from the directory in order to leave open the possibility of recovering killed files if needed. Similarly, other non-RS systems merely tag killed files as inactive rather than wiping all traces of them completely from a diskette directory.

You may position the cursor at any filename by using the arrow keys. By pressing K while the cursor is positioned at a particular file, you "tag" that file for killing. You can hit BREAK at any time to abort this operation without doing any damage, since the purge operation does not actually take place until you give the command to write the updated directory back onto the disk.

If you press C instead of K, you will not only kill the selected file, but also physically remove all traces of its entry from the directory. Pressing the CLEAR key at any time will zero out ALL unused directory entries.

Pressing N will advance the cursor to the next filename in the list and is analogous to using the right-arrow to position the cursor.

By pressing R while the cursor is positioned at a non-active file, you can restore that file to the directory as an active file. However, you should exercise caution when recovering killed files, as other files may have overwritten parts of it. There is no guarantee that a recovered file is still intact, unless you recover it before any other write operations to the disk have taken place.

If there are more files than can be shown on one screen display, you may advance to the next "page" by pressing shift up-arrow, or return to a previous "page" by pressing shift down-arrow.

The changes to the directory are maintained in a buffer in memory until you press W (for "Write to disk"). When you press "W", you will be asked to confirm your decision, i.e., whether you really want to write the changes back to the disk or not. This will give you a chance to change your mind. If you reply "Y", the updated directory is written back to the disk. If you reply "N", then all changes are cancelled, and the original directory on the disk is kept intact.

You may also input more than one drive number, separated by commas or spaces, when initially asked for the drive. Pressing A will read in the directory of the next drive. If you have made alterations to the current directory, make sure that you have written it back to the disk before pressing "A" or all changes will be lost.

Throughout this procedure, BREAK and SHIFT- BREAK will remain active and will bring you to the PURGE Utilities menu or the main menu, respectively, if pressed.

II. KILL FILES BY CATEGORY

This option will permit you to kill certain classes of files from a disk with ease. You will be prompted for a drive, then for the common category. You may enter a filename extension, such as /BAS to kill all files with the /BAS extension, or /CMD to kill all files with the /CMD extension. Do not forget the initial slash ("/") if you are specifying an extension.

WARNING: On a Model I TRSDOS 2.3 disk, and on all non-RS system disks, if you specify /SYS, you will kill not only all system files, but also the BOOT/SYS and DIR/SYS entries in the directory as well! This is a very easy way to make a disk TOTALLY unusable! If you wish to remove the system files, use the wild-card characters option, described in the next paragraph, or the "Remove System Files" option.

You may also use a wild-card option to define a common category of files. Entering a letter or a group of letters alone will cause all files which start with that letter to be killed from the disk. For example, if you were to enter B, you would kill the files BASIC/CMD, BACKUP/CMD, BULLDOG/BAS, BZ/TXT, etc. If you entered BA, however, you would only kill the files BASIC/CMD and BACKUP /CMD.

Finally, you may kill classes of files based on their attributes. You may kill all INVISIBLE files, all VISIBLE files, all SYSTEM files, or files with a protection level other than 0 (no protection). However, see the caution regarding killing system files, above.

To kill a class of files based on their attributes, you must enter a SPACE as the first character when prompted for the common category, followed by I for invisible files, V for visible files, S for system files, or P for files with a protection level other than 0. You may enter more than one attribute, separated by commas or spaces.

TRSDOS 1.3 and TRSDOS 2.7DD system files cannot be killed with this option, since TRSDOS does not log them in the directory in the same fashion as normal files. See the next option.

III. REMOVE SYSTEM FILES

This option is used to safely kill the system files on a diskette. The only input required here is the drive number (you may specify more than one, separated by spaces or commas). All system files except for BOOT/SYS and DIR/SYS will be killed.

Since the Super Utility Plus purge routines do not zero out the directory entries of killed files, these files may be reinstated as active files using the Restore command of KILL SELECTED FILES or the disk repair utility (see below). However, this is safe ONLY as long as no normal disk writes have been done to the disk (EXCEPTION: TRSDOS 1.3 and TRSDOS 2.7DD system files cannot be restored).

IV. REMOVE ALL PASSWORDS

This routine will enable you to strip the passwords from all files on a disk, including system files which are logged in the directory and invisible files. You do not have to specify any previously-set password. When this routine is selected, you will be prompted for the drive numbers containing the diskettes from which passwords are to be stripped. The routine will then read into memory the directory track of each disk in turn, strip the passwords from each file, and change the protection level of each file to 0 (full access). Other file attributes will not be touched. The directory track will then be written back out to the disk, and the routine will proceed with the next one, if any.

This routine will set the DISK master password to "PASSWORD." File passwords (both access and update) will be removed.

V. DISK DIRECTORY

This option will, upon input of the drive number(s), supply you with the name and date on the disk, the number of tracks it was formatted for, the number of free granules remaining, and the number of free file entries available in the directory. In addition, it will display all valid files on the specified diskette(s). The display will include protection level (if any), and the file attributes (whether it is a SYSTEM file, an INVISIBLE file, or a visible file).

This option can work even if you do not know what DOS formatted the disk you wish to look at, by using a "!" in front of the appropriate drive number to activate Super Utility Plus's special DOS detect routines. Note that this process may take a minute or two, and it should not be used on disks which have relative sectoring formats (e.g., ND 80 V2 double density) or disks which were created by operating systems not recognized by Super Utility Plus.

For example:

SYS0/SYS SIP=7

means that the file SYS0/SYS is a system file (S), has the invisible attribute (I), and has a protection level of 7 (no access). The protection levels and their meanings are as follows:

level	protection
0	FULL access
1	KILL + 2,4,5,6,7
2	RENAME + 4,5,6,7
3	Unused
4	WRITE + 5,6,7
5	READ + 6,7
6	EXECUTE + 7
7	NO Access

You will note that each protection level implies all the ones below it. You will note also that protection level 3 is unused in the standard operating systems. You should not assign a protection level of 3 to any files because the results would be unpredictable. Protection level 7 is assigned to system files and means that the user has no access to those files.

If the disk directory has more entries than can be displayed on a single screen page, Super Utility Plus will pause at the end of the first screen page. Pressing any key will display the next page. If more than one drive was specified at the initial prompt, the next drive's directory will be displayed upon completion of the previous one.

VI. ZERO UNUSED ENTRIES

This option will zero out the directory entries for KILLed files. Under most operating systems, a killed file is not removed from the disk directory; its entry is left in place, but flagged as "inactive." This procedure will completely remove the entry of killed files from the directory. It requires only the input of the disk drives containing disks whose directories are to be worked on.

Since TRSDOS automatically zeroes out a directory entry when the KILL command is issued, the usefulness of this utility is in cleaning up other non-RS DOS directories and directories on which Super Utility Plus was used to kill off certain files.

VII. ZERO UNUSED GRANULES

This option may be used to remove ALL traces of killed files from a disk. This may be desirable when security must be maintained at a high level. Input the drive number(s), and the routine will zero out all unassigned sectors on the target disks. Once this utility has been run, no data recovery of any kind is possible for killed or inactive files.

This facility will not touch the diskette directory. To remove inactive file entries, you must use the ZERO UNUSED ENTRIES option. In most cases (with the exception of TRSDOS) when a file is killed the data remains on the disk, and so does the file directory entry; all that happens is that the file directory entry is flagged as inactive. To remove the file directory entry completely, it will be necessary to use the ZERO UNUSED ENTRIES procedure. To remove all traces of a file from a disk, use ZERO UNUSED GRANULES, along with ZERO UNUSED ENTRIES.

VIII. CHANGE DISK NAME

Some operating systems do not supply an easy facility for changing the names or attributes of diskettes. This option in Super Utility Plus will permit you to do all that. You will be prompted for the drive number(s). The directory will be read, and the DISK NAME will be displayed and you will be prompted for another one. If you merely press ENTER at this point, the old name will be retained.

Next, the creation DATE of the disk will be displayed. You may supply a new date, or press ENTER to retain the existing one. You will then be prompted for a new disk master password. If you hit ENTER at this point, the password will be changed to "PASSWORD."

Finally, if you are reading a system disk, you will be told if an AUTO command is active. If so, you may alter the auto command so that a different program is executed on bootup. Pressing ENTER will disable the AUTO command on that disk.

IX. CHANGE FILE PARAMETERS

When you select this option, you will be prompted for a filename. After you supply one, the directory of the specified disk is searched for that file. If it is found, you will then be prompted for the new filename. Press ENTER to leave it unchanged. Next, you will be asked for the new ACCESS password. Pressing ENTER here will set it to blanks. You will then be asked for the new UPDATE password, and you can either supply one or also just press ENTER to set it to blanks.

The routine will prompt you for the protection level next. You must enter a number from 0 to 7 (see the table above. Note that 3 is an unused level and should not be used).

Finally, you will be asked to supply the file attributes. Enter V if you want the file visible, I if you want it invisible, or S to declare it an invisible SYSTEM file. Note that any file can be given the SYSTEM attribute. At this time the updated information will be written back out onto the diskette.

When copying files to or from a TRSDOS 1.3 diskette to any other diskette, this routine will come in handy for removing the passwords from a specific file. TRSDOS 1.3 uses a different password encoding algorithm from all other systems, so a file which may not have passwords on one system is sure to have them when moved to the other! This routine will permit the removal of passwords from specific files only, if you do not wish to use the Remove All Passwords utility.

X. CHECK DIRECTORY

This option scans the directories on the specified drives for errors. You will be advised of the disk name, date, number of formatted tracks, number of free granules and number of free directory slots. In addition, you will be advised of any errors in the directory. Such errors may include granules allocated to nonexistent files, HIT table entries which do not have corresponding file entries, or vice versa, or improperly linked extended directory entries. If any such errors occur, you may repair the disk by selecting the disk repair utilities of Super Utility Plus (see below).

CHAPTER 4 - DISK FORMAT UTILITIES

Bring up the main menu by pressing SHIFT-BREAK. Now press 3 and ENTER. You will be presented with the DISK FORMAT Utilities menu:

```
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. ##  SUPER-UTILITY +  ##  VERSION 3.1A  ##  BY: KIM WATT  ##  .
.       (C)(P) 1983 BREEZE/QSD, INC.  --  DALLAS, TEXAS        .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. __ FORMAT UTILITIES __                                                .
.                                                                       .
.  1 - STANDARD FORMAT           4 - BUILD FORMAT TRACK                 .
.  2 - SPECIAL FORMAT            5 - WRITE FORMAT TRACK                 .
.  3 - FORMAT WITHOUT ERASE      6 - SOFTWARE BULK ERASE                .
.                                                                       .
. CHOICE? #__                                                           .
.                                                                       .
.                                                                       .
.                                                                       .
.                                                                       .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
```

This group of routines permits you to format your diskettes in a variety of ways. All of the utilities will produce formatted tracks that are readable by your particular operating system, except for option number 5, SOFTWARE BULK ERASE, which completely removes all data from a disk.

I. STANDARD FORMAT

This option will format your target disk using the standard format for your operating system. You will be prompted for the drive number(s), the NAME for the disk, the DATE, and the MASTER PASSWORD to be used on the disk. If you press ENTER to the prompt for the name and date, Super Utility Plus will automatically supply the name "* Data ** Disk *" overlapping in the two (name and date) fields. Pressing ENTER in response to the PASSWORD prompt will cause "PASSWORD" to be used.

Finally you will be asked if you wish to use the CONFIGURATION for which the system has been set. A full discussion of the configuration tables is in Chapter 1. Essentially, you may configure Super Utility Plus to recognize that certain types of diskettes will be found in each drive. For example, you may set the configuration table to recognize that drive 0 will always contain a MODEL III TRSDOS disk, formatted for 40 tracks, while drive 1 will always contain a single-density MODEL I DOSPLUS disk formatted for 35 tracks, and so on.

If you answer "Y" to the "Use Configuration ?" prompt, Super Utility Plus will scan the data for the select the disk in that drive should be formatted. If you answer "N", you will get additional prompts asking you to define the format further.

There is a very easy way to add tracks to a diskette using the standard format. For example, if you wanted to turn a 35 track disk into a 40-track disk, without losing any data that is already on it, just place the disk in a disk drive and select standard format. Go through all the prompts until you arrive at the Use Configuration ? prompt. Now reply "N" and press ENTER.

A further prompt will appear, as follows:

:d Type,Tks, Dir, St Tk ?

where :d is the target drive. Answer the prompts. Make sure you give it the proper DOS type. Give "40" for the track count, "17" for the directory track and "35" for the St Tk (Start track).

Super Utility Plus will then check the disk for the presence of valid data, and ask you whether you wish to continue or quit. If you elect to continue with the format at this point, Super Utility Plus will proceed to format the disk, starting at track 35 and moving up to track 39. You will then be prompted whether you want to write the directory track and boot sector. This prompt will appear only if the specified starting track number was other than 0. Reply "N."

DO NOT REPLY "Y" or a blank directory will be written to your disk, rendering previous files on the disk inaccessible!!

To complete the process, check the configuration tables and make sure that the configuration for this drive correctly reflects the number of formatted tracks on the disk. Then go into the REPAIR Utilities menu and select the REPAIR GAT TABLE option. This will open up the extra tracks on your disk, and your 35-track disk is now a functional 40 track disk with 5 extra tracks of space.

This presupposes, of course, that your disk drive was capable of reading and writing 40 tracks in the first place. Also, in order to make use of the added space, your DOS must be capable of recognizing that it is there and is able to make use of it. Adding new tracks to a 35 track with TRSDOS 2.3 will not help you a whit, since TRSDOS 2.3 cannot recognize more than 35 tracks on a disk.

During the format, Super Utility Plus will first scan the disk in the target drive. If it sees something other than a blank disk, it will immediately display the message,

DRIVE n has DATA!

where n is the drive you specified for the Format operation. If possible, it will also display the disk's name and date. You will then be prompted whether you want to continue or abort the operation. If you elect to continue, the data on the disk will be overwritten. A request to abort will put you back in the DISK FORMAT Utilities menu.

There are two special override commands which you can use when the prompt, "Use Configuration?" is displayed. The first one is an asterisk (*). Typing * will force the routine to format but NOT verify the disk. Typing ! will force an immediate format; that is, Super Utility

Plus will check Track 0 for data, but will not check the directory if it does find data, proceeding with the format instead.

When formatting is completed, you will be asked, "Repeat?" You may now take the disk from the target drive and insert a fresh one. Then, typing "Y" will cause the format operation to be repeated. The same information as to DOS type, number of tracks, etc. will be used. If you wish to format a disk differently, you must hit BREAK and re-select standard format in order to specify the new configuration. Typing "N," however, will return you to the DISK FORMAT menu.

If during the format process bad granules are encountered, these will be mapped into the track lockout table in the GAT sector of the directory. However, if you are formatting a disk which uses a relative sector format scheme (i.e, NEWDOS80 V2 double density or MULTIDOS P-Density) there will not be room for mapping bad granules in the GAT table. In fact, these type of disks do not use a lockout table in the same way as regular disks do. What Super Utility Plus will do in such a case is allocate the bad granules to prevent their use by the operating system. The user should note that this procedure of allocating bad sectors may result in subsequent Directory check errors. They should NOT be fixed!!

II. SPECIAL FORMAT

This routine allows you to construct one or more tracks sector by sector, with a variety of formatting options. You can control the contents of the identification fields for each sector, the data address mark for each sector, and even the placement of false CRC bytes. In addition, you can construct a track with a mixture of single and double density sectors (providing, of course, that your computer is equipped with double-density). This routine may be used for constructing specially-formatted "protected diskettes."

When you enter this routine, you will first be asked how many tracks you want to create. After you have entered a number, you will be shown a line that looks like this:

TRACK 00 :SINGLE 00 DOUBLE 00

with a prompt line beneath it. This line indicates which physical track and sector you are working on, and whether the sector is single or double density. The position of the colon (:) indicates which it is. If you are going to mix densities within a track, you must construct all your single-density sectors first before going on to the double-density sectors. Once you start on a double-density sector there is no going back to single till the next track.

To construct an individual sector, you must supply the routine with the necessary information for that sector. These are:

Tnn	**TRACK number**
Hnn	**HEAD number**
Snn	**SECTOR number**

Lnn	**SECTOR length**
I or N	**IBM or Non-IBM length convention**
Ax	**DATA ADDRESS MARK type**
Cy	**CRC type (true or false)**

T, H, S, and L may be followed by a number, or the letter R. That is, you may specify which track, head, and sector number, and sector length, is to be associated with this physical sector. If you specify R instead, you will generate random numbers for the particular parameter you give R to. Any parameter not specified will default to certain values. Track number will default to the physical track number. Sector will default to 0. Head number defaults to 0, length defaults to 1. Remember that the track and sector numbers do not have to correspond with the physical track and sector numbers! In fact, each sector on a track may have a different track and sector number associated with it! HEAD refers to a byte which indicates which side of the disk this sector is on (Head 0 indicates the first side, Head 1 indicates the second side. You can use anything.).

The L parameter is a coded value which indicates the number of data bytes in this sector. How it is evaluated depends on whether you are writing a sector using the IBM convention or not, which is specified by the parameter I or N. IBM is the default. Using IBM conventions, 0 would mean a 128-byte sector, 1 indicates 256 bytes, 2 indicates 512 bytes, and so on. On a Model I, single or double-density sector you may specify IBM or non-IBM (NBM) conventions. The default is IBM in all cases. When creating NBM sectors on a Model I, when you switch from single to double density, the IBM/NBM bit is RESET to IBM, so it will be necessary to re-enter N to indicate Non-IBM for the following sectors. The bit is not reset when going from double density to single density (i.e, when moving on to the next track).

The IBM/NBM bit is not used on the Model III or double-density Model I.

The type of data address mark to be used for this sector only may be specified with the letter A followed by one of four identifiers: S for standard DAMs, R for read-protected, D for deleted data, and U for user defined. For example, a read-protected data address marks would be specified by AR. Note that on double-density Model I TRSDOS and Model III TRSDOS, only the S and R types are available.

The condition of the CRC bytes can also be controlled by entering the letter C followed by a number from 0 to 3. The numbers correspond to certain conditions, given below:

0	**Normal ID and DATA CRCs (default)**
1	**Correct ID field CRC, error on data CRC**
2	**Correct ID field CRC, no data CRC or sector header written**
3	**CRC error on ID field, no sector header or data CRC written**

Thus, to create one sector, a typical command line might be:

T21 H0 SE0H L1 AR C0

This line creates a single sector with the following information: The sector has a track number of 21, has a head ID of 0, a sector number of E0H, an IBM-type length byte of 1, read-protected data address marks, and normal CRCs. The next sector might be created using TFFH H1 S8FH L0 AD C1 which creates a sector with a track number of FFH, and has a head id of 1, a sector number of 8FH, "deleted data" address marks, a correct ID field CRC but false data CRCs.

To switch from single to double density, enter a D on the prompt line. This indicates to the routine that all following sectors created are to be in double-density. Remember that you cannot return to producing single-density sectors once you have moved to double-density until you move to the next track.

Pressing ENTER on a blank prompt line will terminate all processing for the current track. Your special format track will now be residing in memory. You will now be presented with a new prompt asking if you wish to examine the track before writing it out to disk. If you reply "N", then the tracks will be written out to the destination disk.

If you elect to examine the track, you will be placed in the DISPLAY MEMORY mode and the track buffer will be displayed. You can now load up its sectors with data, and make any alterations you wish. Pressing the spacebar will write one track to the disk and return to displaying the NEXT track in the buffer. Thus each press of the spacebar will write one track out. Pressing CLEAR will write all the sectors out to the destination disk without returning to the display mode. The track will be written to the disk, and the program will advance you to the next physical track. As usual, pressing BREAK and shift-BREAK will terminate the procedure and return you to the FORMAT and main menus, respectively.

While in the track display mode, all DISPLAY MEMORY paging and modification controls are active as usual (except for the CLEAR key). See the MEMORY UTILITIES chapter for more information.

III. FORMAT WITHOUT ERASE

This routine will permit you to reformat a disk without losing any readable data that was already on it, and is very useful for revitalizing disks which have been lying around for a long period of time. It is also a good way to repair "Sector not found" and "Data CRC" errors. You will be prompted for the drive number(s). If you specify a "!" at the initial prompt for the drive numbers Super Utility Plus will scan the disk to see how it is formatted, then proceed to reformat it without destroying any data.

When you press ENTER to start the routine, Super Utility Plus will scan each track, reading each sector into a buffer in memory. It will then reformat that track, and write the data back out to it.

If during this process Super Utility Plus encounters a bad sector, it will present you with the R>etry, S>kip, C>ontinuous, N>onstop or Q>uit ? menu. If you are trying to recover a bad disk through this method, use R, C or N as heavily as possible. Don't skip a bad sector at once. If

Super Utility Plus can read the sector at all, it will rewrite it back out with the correct format, thereby repairing the error. So give the program a chance. It may take several minutes, but if the sector can be read at all, Super Utility Plus will do it – and recover your data.

IV. BUILD FORMAT TRACK/WRITE FORMAT TRACK

These two routines will permit you to create specific tracks and write them out to disk at specific locations. This may be useful for saving a glitched diskette, by replacing the damaged tracks with new ones. The data on the damaged track(s) will be lost, of course, but the rest of the disk will be intact.

The BUILD FORMAT TRACK procedure differs from the SPECIAL FORMAT routine in that it only constructs standard format tracks and requires no input from the user other than the DOS type and track number. When you select the BUILD FORMAT TRACK option, you will first be asked to specify the DOS type. Reply with one of the valid DOS type identifiers to create the appropriate track. You will then be asked what track number to assign to it. This should be the track number where you intend to write it out to the disk.

Super Utility Plus will then create the track in memory, and display the message, "Press <ENTER> to view the track at nnnnH". By pressing ENTER, you will be able to view the track buffer in memory. Press BREAK to return to the DISK FORMAT Utilities menu.

Once you have built a format track in memory, you may write it out to disk by selecting WRITE FORMAT TRACK and specifying the track number. The track in the memory buffer will be placed on the disk. Note that you can build a track in memory with one track number, but write it out to another location on the disk. This is entirely possible, but not recommended.

V. SOFTWARE BULK ERASE

This utility will completely remove all data from a target disk and can be used to clean up a disk in place of a bulk eraser or magnet. If you have problems with the DOS rejecting a backup or a format due to "different pack IDs," this routine will solve that problem. You will be asked which drives are to be bulk-erased. Make sure they contain the proper disks! When you press ENTER, the disk will be overwritten completely with 00 bytes, removing all traces of formatting and data. Take as much care in using this routine as you would in using a bulk eraser, since there is no recovery possible.

CHAPTER 5 - BACKUP UTILITIES

There are two routines to this group: Standard Backup and Special Backup. Standard Backup is used on those disks which can be identified using one of the valid DOS specifiers. Special Backup is used for all others.

I. STANDARD BACKUP

Super Utility Plus has a very fast backup routine which can be used in place of the standard BACKUP/CMD of your operating system(s). It will work on a one-drive system as well as on a multi-drive system. Additionally, more than one destination drive can be specified, in which case, the source disk will be backed up to the other drives one at a time.

When selecting this procedure, you will first be asked for the source drive. Enter the drive number which contains your source disk. Do not enter more than one source drive. You will then be asked for the destination drives. You may enter more than one, separated by commas, or spaces.

You will next be asked if you wish to format the destination drives first. You may reply "Y" to format, "N" if the disk is already formatted, "!" to force a format even if the disk has data, or "*" to format and skip the verification cycle. If you type "Y", "!", or "*", each destination drive will be formatted. To avoid any problems, make sure you have configured both the source drive and the destination drive(s) with the same DOS specifier. While it is possible to back up one disk onto another with a different DOS specifier, this is not recommended.

Immediately after formatting, a full disk backup will be performed. Unlike the backup procedure of some systems, this backup will not skip any empty tracks. Nonetheless it will still be quite fast.

If you do not wish to reformat the destination disks, merely reply "N" and the backup will proceed. The destination disks must of course have previously been formatted.

After the source disk has been backed up to all the destination drives, you will be asked "Repeat?" Reply "Y" if you wish to do more backups, or "N" to return to the menu.

If you backup one disk to another with a higher track count, as for example a 35-track disk to a 40-track disk, the backup disk will still reflect the availability of 40 tracks rather than 35 as would result from using other Backup programs.

When using the backup procedure to copy disks, the source and destination diskettes must have a compatible configuration. For example, backing up a single-sided disk onto a drive configured as double sided is illegal, and will result in an error.

Also, the drives must be of compatible type. Attempting to back up a double-sided disk placed in a single-sided drive will result in severe errors.

The BACKUP utility is the easiest way to recover the data from a disk with bad sectors. As each bad sector is encountered, you will be presented with the R>etry, S>kip, C>ontinuous, N>onstop or Q>uit prompt. Use R, N or C whenever this happens. If the sector can be read at all, it will be copied over intact to the backup disk, thereby recovering the data. Even if not all of the sectors can be copied, you will still wind up with a usable (that is, totally readable) disk from which you can copy off as much useful data as you can. In addition, the original disk, bad sectors and all, will still be available for another try.

II. SPECIAL BACKUP

The special backup routine is intended for disks that do not conform to any of the valid DOS specifiers. This routine requires the input of the number of physical tracks on the target disk (or a reasonable estimate thereof). When ENTER is pressed, the program will begin scanning the source disk and building a table in memory of its formatting characteristics. If the routine is able to successfully scan the entire disk, it will then ask you whether you want to view the format track buffer in memory. At this point, Super Utility Plus has constructed an image of the source disk's formatted tracks in memory, and you can reply "Y" to display them (in DISPLAY MEMORY mode) to examine their structure. Note that these are bare formatted tracks, with no data in the sectors. Pressing the spacebar will write the current track to the destination disk and return to displaying the NEXT track. Pressing CLEAR will write the buffer to the destination disk without returning to the viewing mode. While in the viewing mode, all DISPLAY MEMORY paging and modification controls (except for CLEAR, which has a different function) are active. See the MEMORY UTILITIES chapter for more information.

If you do not wish to view the formatted tracks in memory, simply reply "N" in response to the prompt, and the tracks will be written out to the destination disk. After the destination disk has been formatted, then the data will be copied from the source disk onto the destination disk.

This routine is intended for your own personal use only in making backups of disks which you own. There is no guarantee made that the Special Backup routine will be able to duplicate every disk ever produced with a special format. In fact, there exist particular formats which by their very nature cannot be easily duplicated, or duplicated only partially, without introducing unwanted changes into their structure. There are also certain types of formats which, while they can be duplicated, cannot have any data written to them without destroying their special characteristics. If you encounter a disk which cannot be copied with this routine, please contact your dealer or supplier to obtain a copy. Most dealers who sell protected disks have a reasonable backup policy and will allow you to purchase a copy at a nominal fee. Please do not call us for updates to the SPECIAL BACKUP routine.

CHAPTER 6 - REPAIR UTILITIES

Selection of this group from the main menu will produce the following menu display:

```
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. ##  SUPER-UTILITY +  ##  VERSION 3.1A  ##  BY: KIM WATT  ##  .
.       (C)(P) 1983 BREEZE/QSD, INC.  --  DALLAS, TEXAS        .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. __ REPAIR UTILITIES __                                                .
.                                                                       .
.  1 - REPAIR GAT SECTOR          6 - RECOVER KILLED FILES       .
.  2 - REPAIR HIT SECTOR          7 - MOVE DIRECTORY             .
.  3 - REPAIR BOOT SECTOR         8 - DISPLAY DIRECTORY          .
.  4 - READ-PROTECT DIRECTORY     9 - CHECK DIRECTORY            .
.  5 - UN-READ PROTECT DIR       10 - CLEAR UNUSED ENTRIES       .
.                                                                       .
. CHOICE? #__                                                           .
.                                                                       .
.                                                                       .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
```

These utilities are designed to restore disks which might be unusable or contain garbled data to a usable condition, if it is at all possible. Many times a disk becomes unreadable because of some damage to the directory. If the damage is not too extensive, the REPAIR utilities will be able to fix it. However, the disk itself must be readable, i.e., it must not contain any format errors or CRC errors which will prevent Super Utility Plus from reading it. If necessary, use FORMAT WITHOUT ERASE on the disk to make sure it is readable.

In addition, the REPAIR utilities contain routines which will allow you to check on the condition of a directory, recover any files killed by the PURGE utility of Super Utility Plus, move the directory track from one location to another on the disk, or clear the unused directory entries by setting them to zero.

I. REPAIR GAT SECTOR

The GAT (for Granule Allocation Table) sector of a diskette directory is where the operating system keeps track of which granules have been assigned to files, which are available for use, and which have been locked out for one reason or another. An error in the GAT can result in files being allocated disk space which already belong to other files, thus resulting in DATA loss. This is the kind of creeping error which can eventually render an entire disk worthless.

The GAT REPAIR option will permit you to repair a bad GAT table. Upon entering this option, you will be asked for the drive number containing the disk to be repaired. You may enter more than one drive number if you are repairing several disks. You will then be asked whether you want the GAT Table only (the first 96 bytes), or the entire sector, to be repaired.

If the disk name, and date are unreadable, select the ALL option to rebuild it. Otherwise, if the GAT table alone is damaged, select GAT Table repair only. This will allow you to retain the original disk name, date, password, and any active auto command. Super Utility Plus will then proceed to rebuild the granule allocation table of the target disk(s).

II. REPAIR HIT SECTOR

Each filespec in a directory is "hashed" by the operating system into a one-byte code and entered into the HASH INDEX TABLE sector, in a position which defines the file's actual position in following directory sectors. This permits the operating system to find filespecs in the directory very quickly. Errors which may occur here are HIT codes assigned to nonexistent files, invalid HIT codes, etc. Or the sector itself may have been damaged so that part or all of it may have been turned into garbage.

If a directory check reveals errors in the HIT sector, you may use this option to repair it. The only required input is the drive number containing the damaged disk (you may enter more than one drive number). When you press ENTER, Super Utility Plus will proceed to reconstruct the entire HIT table.

TRSDOS 1.3 codes the SYSTEM files in the last 32 bytes of the HIT table, and TRSDOS 2.7DD in the last 48 bytes. Because of the fact that they do not have any standard directory entries, these bytes will be left untouched by Super Utility Plus during a HIT repair, since without standard entries, it is not possible to reconstruct a HIT entry. If these bytes have been corrupted, you must use the ZAP utility to effect repairs. If the bytes look obviously invalid to you, enter FF in all 32 slots (48 for TRSDOS 2.7DD).

III. REPAIR BOOT SECTOR

If a system disk will not boot, the chances are that the boot sector has been damaged. This sector contains code which loads the rest of the operating system into memory, so if this sector is corrupt, the rest of the DOS will not load. If this is the case, you can effect a repair of the boot sector by selecting this option. You will be prompted for the drive number containing the disk with the damaged boot. When you hit ENTER, Super Utility Plus will write the new boot routine into this sector.

Super Utility Plus contains three different boot routines, all of which are intended for use on TRSDOS disks ONLY, which it can use to replace a damaged boot sector. These boots are for a single density disk, a double-density disk with a single density Track 0, and for a pure double-density disk. The one which Super Utility Plus writes to your disk will depend on the DOS type. The boots are not the same as the standard boots, but very fast routines which check each byte of the the resident operating system module (SYS0) as it loads it into memory to ensure that the load is a good one.

This repair option does not log the boot in the directory (the operating system will already have it logged in anyway). All it does is write the new routine into the sector formerly occupied by the damaged one.

To repair the boot sector on a non-TRSDOS disk, first make sure that the boot track is readable (does not contain any CRC errors, etc.). Then take a GOOD disk and copy the boot sectors from it onto the bad disk using the COPY SECTORS procedure in the ZAP Utilities.

IV. READ-PROTECT DIRECTORY

If you try to boot a system diskette, and the disk drive seems to have difficulty in finding a file, or if you try to read a file on a data diskette and the drive seems to be hunting all over the diskette, or if you try to display a directory and the disk drives grind along for several minutes, the chances are that part or all of the directory was re-written using the wrong DATA ADDRESS MARK (Note: this can also happen if you try to read a single-density Model I TRSDOS diskette on a Model III machine). The directory track of a diskette has a data address mark (see chapter 2) which is different from all the other tracks, and is used by the operating system to identify its location.

Whenever the DOS tries to read a directory and finds that some or all of the directory sectors do not have the correct data address mark, it will hunt all over the disk for a track that has the proper DAMs. It will not find it, and so your drive sounds like its head keeps moving back and forth (which is exactly what is happening).

READ-PROTECTING a directory is a misnomer and refers to the process of writing a directory track with the correct data address mark which can be recognized by the operating system. It does NOT mean changing the directory so that it cannot be read at all. Unfortunately, the term stuck.

When this option is selected, you will be prompted for the drive number(s) containing the target disks. When you press ENTER Super Utility Plus will read the directory track, then write it back out with the correct data address marks. If it cannot find the directory you will be asked to specify its track location and length (i.e., number of sectors in the track). You may enter the track number where the directory is located, or press ENTER to default to the directory track number and track length in the configuration table.

If you recall the discussion of data address marks in chapter 2, you will realize that if you try to read a MODEL I single-density disk in a Model III machine, your MODEL III will not be able to find the directory track. The reason is that the data address mark used by Model I TRSDOS (and other similar systems) is one that the MODEL III hardware cannot recognize. If this problem does not occur with another operating system, it is most likely because that system uses a data address mark for the directory track that both the Model I and Model III can recognize.

This has other implications. Super Utility Plus will adapt itself to whatever machine it is running on. If you attempt to read-protect a TRSDOS 2.3 disk directory on a Model III machine, you will write the WRONG data address marks on the Model I disk! So even after you thought you had fixed the problem, now your TRSDOS 2.3 will be unable to locate the directory track!

If you are going to use this option, therefore, make sure that you are using it on the RIGHT machine. Do not attempt to read-protect a TRSDOS 2.3 directory track on a Model III computer, and vice versa. This restriction may not apply to other operating systems, however.

V. UN-READ PROTECT DIRECTORY

This option takes a directory track and rewrites it using standard data address marks (or RPT data address marks, if the disk is Model III TRSDOS). The only input required is the drive number. You may want to use this routine if you are developing a protected disk with a directory that you do not want any standard operating system to be able to read.

VI. RECOVER KILLED FILES

This routine will allow you to recover files previously killed using Super Utility Plus's PURGE utilities. You cannot use it to recover files killed with the standard TRSDOS "KILL" command, as the entire directory entry is zeroed out when a file is "KILL"ed (you may, however, use it to recover a file KILLed under other non-RS operating systems).

Remember that recovering a killed file is a chancy affair IF you have performed any disk writes since the time you killed the file. In the first place, you may introduce errors into the directory's GAT (granule allocation table). Secondly, the file you recover may have had parts of it overwritten by subsequent files. So be careful when using this routine.

You will first be prompted for the drive number or numbers. Upon hitting ENTER, Super Utility Plus will display the disk name, date, number of formatted tracks, and number of free granules and free directory slots on the disk. Press ENTER again. You will now be shown a list of all VALID and NON-VALID (i.e., killed) files in the directory of the target disk. All valid files will be surrounded by right and left arrows (right and left square brackets on the Model III). Non-valid files will be surrounded by solid graphics blocks.

To recover a killed file, move the cursor to the file that you wish to restore, using the arrow keys. Press R and you will see the graphics blocks replaced by right and left arrows (or brackets), indicating that the file is now a valid directory entry.

When you have recovered all the files you want to, press W to write the updated directory back onto the disk. You will be asked to confirm your decision. Press "Y" to confirm and write the directory, or press "N" to abort the procedure. Pressing A will abort the process and proceed to the next drive if more than one was specified, so if you have made any changes, make sure you write the directory back out to disk before using A to proceed to the next target drive.

If there are more files than can be displayed on the screen at one time, you can press SHIFT and UP ARROW to go the next display page, or SHIFT and DOWN ARROW to move to the previous page.

VII. MOVE DIRECTORY

This option will move the directory track to any other track on the target disk as long as that track is not already allocated to another file. You will be prompted for the drive number(s) containing the target disks, and the track number where you want the directory moved. If the track you selected is already allocated (either fully or partially), Super Utility Plus will display a message to that effect and ask you to select another track. Upon entering the track number and pressing ENTER the directory track will be moved, and the boot sector changed so that it now points to the new updated directory (the position of the directory is coded in the second or third byte of the boot sector depending on the operating system involved). Also, on TRSDOS 2.3 and all other non-RS disks, the DIR/SYS entry in the directory will be updated to reflect the directory's new location.

If you wish to abort this routine, BREAK or SHIFT-BREAK will always take you back to the menus.

VIII. DISPLAY DIRECTORY

This option requires only the drive number or numbers containing the disks whose directories you wish to view. It will give you a full screen display of all active files on the disk(s), along with the attributes and protection levels of each file (protection level 0 is not displayed). In addition, the disk name, date, number of free granules and free space in K (kilobytes, units of 1024 bytes), and number of free directory slots are also displayed.

If you view a TRSDOS 1.3 or 2.7DD directory, you will not see any system file names displayed. As was discussed above, TRSDOS 1.3 and 2.7DD does not code the system files in the directory in a normal manner. They will therefore be displayed as numbers, e.g.

00-01-02-03-04-05-06-07-08-09-10-11-12-13-14

indicating which of the relative system files are currently active (not killed) in the directory.

IX. CHECK DIRECTORY

This option is designed to perform a very thorough check of the directory on the target disk. You will be prompted for the drive number. When you hit ENTER, the routine will scan the directory. It will then display the disk name, date, the number of free granules and the number of free directory slots. Any errors encountered will then be displayed.

If the routine reports any GAT or HIT errors, you may use the GAT REPAIR or HIT REPAIR options to automatically repair the directory.

X. CLEAR UNUSED ENTRIES

This option will permit you to clean up a diskette directory by completely erasing the entries of any non-active files. Normally the entries of non-active files are left in the directory by systems other than TRSDOS 1.3 and 2.7DD and the Super Utility Plus purge routines to leave available the possibility of future recovery. If you use this routine, of course, no future recovery will be possible.

You will normally not need to use this directory to clean up files where TRSDOS 1.3 or 2.7DD did the killing, since these TRSDOS systems remove the entries of non-active files anyway.

This option requires only the input of the drive number(s) containing the target disk(s). You may enter more than one drive number, and the directories will be scanned and cleaned one by one.

CHAPTER 7 - TAPE UTILITIES

The TAPE UTILITIES of Super Utility Plus give you the following procedures:

```
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. ##  SUPER-UTILITY +  ##  VERSION 3.1A  ##  BY: KIM WATT  ##  .
.       (C)(P) 1983 BREEZE/QSD, INC.  --  DALLAS, TEXAS        .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. __ TAPE UTILITIES __                                          .
.                                                               .
.                                                               .
.  1 - READ TAPE                3 - VERIFY TAPE                .
.  2 - WRITE TAPE               4 - COPY TAPE                  .
.                                                               .
. CHOICE? #__                                                   .
.                                                               .
.                                                               .
.                                                               .
.                                                               .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
```

It should be noted at the outset that these utilities do not provide any disk-to-tape or tape-to-disk interface. The TAPE UTILITIES are intended for tape-to-tape procedures. Secondly, the user should remember that only 500 baud cassette operations are supported. The higher 1500 baud tape rate of the Model III is not supported by Super Utility Plus.

I. READ TAPE

If you are using a Model I this routine will prompt you for the deck number to read from and input the data from the specified deck, or from the regular tape port on a Model III. When this option is selected, you will see the message:

<KEY> to begin:

Press the PLAY button on your cassette recorder, then press ENTER. The routine will begin reading the tape, and the message "Looking for sync byte" will appear. You may abort the operation at any time by pressing the CLEAR key. Once the routine picks up the sync byte it will start reading the data into a memory buffer. The data will be displayed on the screen. The actual ASCII character will appear on the top row, and below it, reading vertically, the ASCII HEX value of that character.

The read operation will continue until the CLEAR key is pressed, or the buffer fills. When the read is completed, you will be told how many bytes were read in from the tape and where they are stored. You will also be given the checksum calculated from the data. Pressing ENTER at this point will take you into MEMORY DISPLAY mode, and you will be able to view the data that was read in. See the next chapter for details on the memory display routine.

The tape read operation will abort when the buffer fills up. You will be told how many bytes were read in up to the time of the abort, and then given a chance to display the data and examine it.

II. WRITE TAPE

This procedure will allow you to write a tape using the data that is in the holding buffer. If there is no data in the buffer, you will be informed. Then you will be asked for a starting and ending address. Note that you may specify an address anywhere in memory, the ROM space included, and write that out to tape. If you are using a Model I, you will be prompted to specify which tape deck (1 or 2) to use. When you press ENTER to begin the write operation, the routine will write the leader and sync byte, and then the data. The data is displayed on the screen as it is written. Again, you may abort the operation at any time by pressing CLEAR.

Note that this routine will write out the data exactly as it finds it. No conversion to any particular format will be done; the routine assumes that the data is already in the proper format.

At the completion of the tape write the routine will display the number of bytes written, the starting and ending address, and a two-byte checksum of the data. Pressing ENTER at this point will bring you back to the TAPE UTILITIES menu.

III. VERIFY TAPE

Selection of this procedure assumes that the data you want to verify has previously been read into the holding buffer with READ TAPE. Reposition the tape to the start of the file, press the PLAY key on the appropriate recorder, and press the ENTER key on the keyboard. The program will begin reading the tape again and will compare the data with whatever is in the holding buffer. The data will be displayed as it is read in, and discrepancies will be indicated with an asterisk above the offending byte. At the end of the verification process, you will be told how many errors were encountered during verification.

Note that the routine has no way of knowing WHICH of the byte pairs compared was in error -- the byte read into the holding buffer the first time, or the byte read in during the verification. It will simply report the discrepancies.

IV. TAPE COPY

This routine will allow you to back up a tape. You will need two tape recorders. The cable coming from the cassette port of the computer terminates in three jacks. Insert the black jack into the EAR plug of the source recorder. Insert the grey miniature jack into the MIC input of the destination recorder, and the grey subminiature jack into the motor control of the destination recorder also. You may need to use patch cords if the wires are not long enough.

Select the copy tape routine. Specify the SOURCE deck. Press ENTER. Press the PLAY button of the source recorder, and the PLAY and REC button of the destination recorder. A graphic block will appear on the screen. As soon as valid data is picked up, this block will begin flashing.

The routine will read each bit from the source recorder, clean it up, and write it immediately to the destination recorder without storing it in memory. This should produce a byte-for-byte copy of the original tape. The routine will have to be manually terminated by pressing CLEAR.

CHAPTER 8 - MEMORY UTILITIES

The MEMORY Utilities of Super Utility Plus permit you to view and manipulate memory in a variety of ways. Many of these routines are very similar to the ZAP utilities, as can be seen from the menu below:

```
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. ##  SUPER-UTILITY +  ##  VERSION 3.1A  ##  BY: KIM WATT  ##  .
.       (C)(P) 1983 BREEZE/QSD, INC.  --  DALLAS, TEXAS        .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. __ MEMORY UTILITIES __                                          .
.  1 - DISPLAY MEMORY            9 - STRING SEARCH               .
.  2 - MOVE MEMORY              10 - INPUT BYTE FROM PORT        .
.  3 - EXCHANGE MEMORY          11 - OUTPUT BYTE TO PORT         .
.  4 - COMPARE MEMORY           12 - MEMORY TO SECTORS           .
.  5 - FILL MEMORY              13 - SECTORS TO MEMORY           .
.  6 - REVERSE MEMORY           14 - MEMORY TO TRACK             .
.  7 - TEST MEMORY              15 - TRACK TO MEMORY             .
.  8 - JUMP TO MEMORY                                           .
.                                                               .
. CHOICE? #__                                                   .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
```

In addition, the memory utilities permit you to read and write to any I/O port from 0 to 255. You may also perform a direct memory-to-disk or disk-to-memory transfer.

Great care should be taken when modifying memory in any of the ways made available to you from this menu. Unlike the modification routines in the ZAP utilities, changes to memory are not buffered and take effect immediately on entry.

I. DISPLAY MEMORY

This option requires input of the starting memory address that you want displayed. You may enter the address in any of the four number bases that Super Utility Plus will accept (Hexadecimal, Decimal, Octal, or Binary). The default base is Decimal. To enter a hexadecimal number, you must append "H" to the number; to enter an octal value, append O (the letter "O" -- not zero!!!) or Q and to enter a binary number, append B to the number.

The display is very similar to the ZAP display, with the exception that the first column of information is empty, and the actual addresses (in hex) are displayed in the second column, adjacent to the actual data. The data is displayed in blocks of 256 bytes each. You can look on this display as a window into the TRS-80's memory which can be moved around so as to present different views of the RAM or ROM "landscape."

Using the arrow keys, you can scroll through memory. By pressing the right or left arrows, you can move the display window one by in either direction. Pressing SHIFT-right arrow or SHIFT-left arrow will advance or decrease the display 256 bytes at a time.

Pressing up-arrow or down arrow will also move the display window 256 bytes at a time in either direction. However, pressing SHIFT-up arrow has a different effect; this will cause the highest 256-byte block of memory available in the computer to be displayed. Similarly, pressing SHIFT-down arrow will display the lowest 256-byte block of memory.

Pressing the CLEAR key will cause Super Utility Plus to prompt for a new memory address to display.

Pressing M will put you in Modification Mode. As in the ZAP utility's modification mode, you can select your input base by pressing H (hexadecimal), O or Q (octal), D (decimal), B (binary) or A (ASCII) before pressing M. The cursor may be moved with the arrow keys, and the action of the modification control keys are the same as in the ZAP utilities. See table 2-2 for the various keys and their actions.

In memory modification mode, however, if you attempt to move the cursor past the display window, more data will simply be brought in, one byte at a time. If you attempt to move the cursor past the highest address in the computer, FFFFH, the display will wrap around to display 0000H. Pressing ENTER in modify returns to paging mode.

II. MOVE MEMORY

This routine enables you to move a block of memory from one location to another. You will be prompted for the starting and ending address of the block to be moved, and the starting address where it is to be moved to. The prompt will look like this:

Start, End, Start ?

If you wanted to move the bytes located at 7000H through 70EBH inclusive to a new location starting at AB44H, you would then reply as follows:

Start, End, Start ?7000H,70EBH,AB44H

and press ENTER. The routine will then execute the memory move, and you will be advised of the number of bytes moved.

Be careful when moving things around in the lower 32K of memory, which is occupied by Super Utility Plus itself! If you move a block of memory into a location being used by the program, you may destroy vital parts of it and find yourself facing a disastrous system crash

III. EXCHANGE MEMORY

This routine is similar to "Move memory," except that it actually exchanges the contents of the origin and destination blocks of memory instead of merely copying the contents of the origin block over what was in the destination block.

The prompts will be identical to Move Memory, and will request the starting and ending locations of the origin memory block, and the starting location of the destination memory block where the exchange is to take place. For example, if you wanted to exchange the contents of the block of memory in 8000H through 87FFH with the contents of memory at C000H, your reply would be

Start, End, Start? 8000H,87FFH,C000H

and when you press ENTER, the two blocks of memory will be exchanged. The bytes which resided at 8000H through 87FFH now reside at C000H through C7FFH, and vice versa.

IV. COMPARE MEMORY

This option permits you to compare one block of memory with another. Suppose you had executed a memory move using option 1 and wished to verify if the move was properly executed. You would then enter the starting and ending addresses of the original block and the starting address of the comparison block.

The program will do a byte-by-byte comparison of the two blocks of memory and advise you of any mismatches that it finds. If there are many such mismatches, they will scroll past on the screen at a high rate of speed, but the display may be paused by holding down the spacebar.

V. FILL MEMORY

Memory may be filled with a single byte value using this routine. You will be p prompted for the starting and ending addresses of the block to be filled, and the byte value to use as the filler. For example, if you wanted to fill the memory locations from DA00H through DFFFH with the byte 0AH, you would enter, DA00H,DFFFH,0AH.

All of the address entries have default values. The starting address location defaults to the first byte beyond the end of the Super Utility Plus program itself, and the end byte defaults to the top of memory. The fill byte defaults to 00. Thus, pressing ENTER at this prompt will zero all memory not used by Super Utility Plus and can be used to "clean up" memory for other procedures which may require use of this space (such as "Track to Memory" -- see below).

Be careful when using this routine in the lower 32K of memory as you could wipe out a critical part of Super Utility Plus and cause the program to crash. You can easily determine the first free memory address by selecting Display Memory and hitting ENTER in response to the

address prompt. Super Utility Plus will display the first 256 bytes of FREE memory. You should not do any FILL MEMORY function below the address on the upper left hand corner of the display.

VI. REVERSE MEMORY

This option performs the same function as the "Reverse Sectors" option of the ZAP Utilities, except that it works on a block of memory rather than on a disk sector, and the area to be reversed is not limited to 256 bytes. You only need to enter the starting and ending addresses of the memory block to be reversed, and the routine will execute immediately. On completion of the routine you will be advised of how many bytes had been reversed. Note that unlike Reverse Sector Data, which is sector oriented, this routine is block oriented and will perform the reversal from the starting address to the ending address inclusive.

VII. TEST MEMORY

This option will perform a test of the region of memory that you specify. The test is a very complete one, but will not disturb the previous contents of the tested addresses! After each byte is tested, its original contents are restored. Thus you could test the region of memory occupied by Super Utility Plus without destroying the program in memory.

There is, however, an exception to this. You cannot test the actual locations from which the routine executes, or it may malfunction and behave unpredictably. Thus, when you select this option, you will see the message,

DO NOT test between xxxxH and yyyyH !!
Start, End ?

xxxxH and yyyyH are the locations where the test routine executes from, and altering them at the wrong time will definitely cause unwanted results.

To use this routine, merely enter the starting and ending addresses of the block of memory you wish to test and press ENTER. The memory test may be stopped at any time by pressing the CLEAR key.

If you attempt to test the addresses occupied by the BASIC ROMs (read only memories), you will get a display indicating that every address is bad. This is because ROMs cannot be written into.

Also, if you test the video RAM on a Model I without a lower-case modification installed, those addresses (3C00H through 3FFFH) will also be reported back as being bad. However, this is normal (the video memory on an upper-case -only Model I only has 7 bits per byte instead of 8) and does not indicate that something is wrong with your video memory.

Any errors encountered will be displayed by this routine, and you will be given a bit-by-bit breakdown of the problem addresses.

VIII. JUMP TO MEMORY

This option will allow you to execute a jump, or transfer of control, to any memory address you select, and you need only enter that address. This would normally be done if you wanted to jump to your own machine-language subroutine in high memory, for example. This routine is written in such a way that a simple RET instruction (C9H) will bring you back into Super Utility Plus. However, DO NOT try jumping into random locations blindly, especially into the Super Utility Plus program! Know what you are doing before trying a jump to memory!

IX. STRING SEARCH

This routine will allow you to search memory for the occurrence of a particular ASCII string, byte string, or two-byte word string, and replace them with another string of your choosing. The routine works in the same fashion as the STRING SEARCH routine of the ZAP Utilities, except that this one works on memory, not disk. You must enter the starting and ending locations of the memory block to be searched. Next you will be asked to enter the string to be searched for.

Enter an ASCII string directly. Surround the string with double quotes if you want an exact case match on the search, or with single quotes if you want a case independent search to be made. To enter a byte string, enter a series of values in the range 0 - 255 decimal and separate them with spaces or commas. The values may be entered in any of the acceptable numeric bases recognized by Super Utility Plus. To search for two-byte WORD strings, enter two-byte values separated by spaces or commas. The two-byte values should not exceed the range 0 - 65535 decimal. Remember that the routine will reverse the order of two-byte words in order to correctly search out address references, which are kept in LSB-MSB (least significant byte - most significant byte) order. Also, note that leading zeroes may cause the program to evaluate a two-byte value into one byte. For example, 0033H would evaluate to 33H instead of 3300H.

As with the search string function of the ZAP utilities, you may enter a search string of intermixed ASCII, one-byte values, and two-byte values.

If you do not wish to replace the original string, merely press ENTER when prompted for the replacement string, and the program will display the matches as it finds them, along with their locations. If there are many of them, they will scroll past at a high rate of speed. Hold down the spacebar to temporarily stop the display. If the string was to be replaced, they will be replaced at this time. Remember that if the replacement string is longer than the search string, it will be truncated to the length to the search string. However, if the replacement string is shorter than the search string, only the corresponding number of bytes in the search string will be replaced.

X. INPUT BYTE FROM PORT

This option will allow you to input and display a byte from a hardware port. You need only supply the port number from which you wish to input. For example, if you wished to read the modem status register on your machine (port E8H) you would reply E8H (or 232 in the default decimal base) to the prompt, "PORT ?" and Super Utility Plus would immediately return the value in that port.

XI. OUTPUT BYTE TO PORT

Conversely, this option permits you to send a particular byte out to a hardware port of your choice. You will be asked for the PORT number and the byte you wish output to it. Upon hitting ENTER, Super Utility Plus will immediately send that byte out to the specified port. There will be no acknowledgement. To see the effect of this option, hook up your cassette recorder to your TRS-80, press the PLAY button, and select this option. In response to the PORT,BYTE? prompt, enter FFH,4 (ECH,2 for the Model III). You will see that the cassette recorder's motor has been turned on. To turn it off, enter FFH,0 (ECH,0 for Model III).

XII. MEMORY TO SECTORS

Selection of this option will allow you to write out to disk a block of memory. It requires that you input the starting byte of the block of memory you wish to move to disk, and whether the sector to be written is to be in IBM format or not (see the discussion of data address marks in chapter 2). You will then be asked to supply the drive number, the track number and starting sector to be written on your disk. Finally you will be asked how many sectors are to be written.

If you specified IBM format, then sectors consisting of 256-byte memory blocks will be written to disk, starting at the address which you specified.

Memory saved to the disk this way is NOT noted in the disk's directory, so it will remain invisible to the DOS unless you make an entry in the diskette's directory for it. However, assuming that you remember the locations on the disk where you wrote it out, you may reload the sectors back into memory using Super Utility Plus (see the next option).

XII. SECTORS TO MEMORY

This option permits you to read in any sectors from disk into a specified region of memory in your TRS-80. You must know the locations on the disk which you want to load, and you must also know where in memory you want to load the data. There are three prompts under this option. The first is

Drive, Track, Sector ?

Enter the drive number, track number, and starting sector number that you wish to load into memory. You will then be asked,

Sector count ?

Enter the number of sectors you wish to read into memory. If you attempt to load more sectors than your memory can hold, Super Utility Plus will detect this and load only what it can. You will be informed of how much was actually read into free RAM.

Finally, you are asked, ADDRESS ? Here you must enter the starting memory address that you wish to load the sectors into. If you wished to load the sectors into memory starting at address A000H, you would reply A000H and press ENTER. This should normally be defaulted by pressing ENTER, however, since the default value will point to a safe area in memory away from any region used by Super Utility Plus.

Upon pressing ENTER to the ADDRESS prompt, the disk sectors you specified will be read into memory.You will then be informed whether or not any sectors could not be loaded (this would happen if there was a flaw on the disk). Then you will be given the message, PRESS <ENTER> TO DISPLAY.

Pressing ENTER will immediately display the region of memory that was just loaded from disk, in the DISPLAY MEMORY format. In effect you will be in DISPLAY MEMORY mode, and you may carry out any operations that are available to you in that mode. You can exit back to the menu by pressing BREAK, or to the main menu with SHIFT-BREAK.

XIII. MEMORY TO TRACK

This option will permit you to write a section of memory out to the disk as ONE track. You are asked to input the starting address in memory for the write operation, the drive number, and the track number to be written. A track's worth of data starting from the address which you specified will be written out to disk.

BE CAREFUL WHEN USING THIS OPTION!!! This procedure assumes that you have constructed in memory an image of a formatted disk track, which not only consists of the data to be written, but also the actual formatting information that the DOS (and Super Utility Plus itself!) needs to read it. This includes not just the actual sector data but also the track ID fields, the sector ID fields, the CRC bytes and the inter-sector gap bytes. If any of these are not present or are incorrectly positioned, you will have constructed a totally unreadable track on your disk with NO way to read it back in. It is strongly suggested that if you do not have any experience in constructing format tracks, that you do not attempt to use this option unless there is valid track data in memory already, created by the "Build Format Track" utility.

XIV. TRACK TO MEMORY

This option will read a full track of data into memory, including all formatting information such as sync fields, data address marks, gaps, etc. You will be asked to input the drive number and track number where you want to start examining. Then you will be asked whether you want the floppy disk controller to synchronize on the ID address marks during the read or not. This means that the controller chip will start its accumulation of data at the data address marks rather than elsewhere. If you want this option, reply "Y" otherwise reply "N".

An entire track of information will be transferred from the disk into a holding buffer in memory, and you will be informed where it is located. You may view the data by pressing ENTER. This will also put you in DISPLAY MEMORY mode, with all the operations described above available.

Note that the TRACK TO MEMORY operation is not reliable when double-density tracks are involved. The reason is that when reading a double-density track, the floppy disk controller chip automatically synchronizes to ID marks regardless of what you told Super Utility Plus. Certain bit patterns can fool the FDC into thinking that it is looking at an ID mark and cause it to re-synchronize. The net effect is lost data. An example of this is a track with the word "TRSDOS" somewhere in it. The FDC will see the bit pattern of the "T" as an ID mark and try to re-synchronize at that point, resulting in a bad read. But if the capital "T" is changed to a lower-case "t", the FDC will NOT attempt to re-synchronize.

Experienced programmers can use this option to create protected tracks on a disk, for example, by changing the CRC bytes on each sector so that a standard DOS will always signal an error, or changing the actual track numbers or sector numbers so that the DOS will not be able to read the track at all. The altered track can be written back out to disk with the MEMORY TO TRACK option, above. However, this exercise should not be undertaken lightly by people who have little experience with disk formatting. You could create an unreadable disk otherwise.

To see the difference between this procedure and the DISPLAY DISK SECTORS procedure in the ZAP utilities, use this option to read in the directory track of a TRSDOS disk and examine it. Then examine the same track using the DISPLAY DISK SECTORS option and you will see the difference. The DISPLAY DISK SECTORS procedure does not show you the formatting information written on the disk.

CHAPTER 9 - FILE UTILITIES

The final group of routines available in Super Utility Plus are the FILE Utilities, selected by pressing 8 at the main menu. These routines operate very much like the ZAP utilities, except that they are file oriented rather than disk-oriented. This means that if you display a file using the FILE Utilities, you will be able to view all of that file even though it may be in two or more segments located at different places on the disk. The FILE Utilities use the file information maintained in the diskette's directory to know where to find the file. Of course, this also means that you cannot use the FILE Utilities to examine a file on a disk with a non-standard or missing directory.

The available utilities in this group are as shown on the menu:

```
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. ##  SUPER-UTILITY +  ##  VERSION 3.1A  ##  BY: KIM WATT  ##  .
.       (C)(P) 1983 BREEZE/QSD, INC.  --  DALLAS, TEXAS        .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
. __ FILE UTILITIES __                                           .
.                                                                .
.  1 - DISPLAY FILE SECTORS      8 - DRIVE STATUS                .
.  2 - COMPARE FILES             9 - SECTOR ALLOCATION           .
.  3 - COPY FILES               10 - BUILD FILE                  .
.  4 - DISK DIRECTORY           11 - CLEAR FILE                  .
.  5 - FREE SPACE               12 - DISK ALLOCATION             .
.  6 - OFFSET FILE              13 - COMPUTE HASH CODE           .
.  7 - FILE LOCATIONS           14 - COMPUTE PASSWORDS           .
.                                                                .
. CHOICE? #__                                                    .
#  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  .  #
```

I. DISPLAY FILE SECTORS

This option requires the input of the filename that you wish to see. Do not forget extensions and drive numbers when entering the specification. Super Utility Plus will search its directory when you press ENTER. If it finds the file, it will display the following information:

END OF ALLOCATION SECTOR NUMBER (EOA) - Since a standard DOS allocates disk space in granules, it often happens that a file will not completely fill up the last granule allocated, so that one or two sectors are allocated to it but are actually empty. The EOA number displayed by Super Utility Plus refers to the last sector of the last granule allocated to that file by the operating system.

END OF FILE SECTOR NUMBER (EOF) - This is the LAST sector of the file which actually contains file data. It may or may not agree with the EOA number, but will normally be within the last allocated granule.

Super Utility Plus will then prompt you for your.

CHOICE ?

This refers to your CHOICE of which sector in the file to view. You need to enter your choice in relative sector form, that is, the first sector of the file is Sector 0, and so on. If you wish to begin viewing the file from the first sector, merely press ENTER; otherwise enter the sector number you wish to see.

Once the display is on the screen, then you may move through the file by using other keys. The arrow keys will page through the file as usual. In addition, you may go to the EOF sector by pressing "E" or to the EOA sector by pressing "A". Pressing the CLEAR key will allow you to select another sector to view.

TABLE 8-1 - FILE UTILITIES PAGING CONTROLS

Key	Action
Right or Up arrow	pages to the next higher sector of the file
Left or Down arrow	pages to the next lower sector of the file
Sh-right/Sh-up arrow	displays last allocated sector of the file
Sh-left/Sh-down arrow	displays initial sector of the file
CLEAR	requests new relative sector to display
E	displays end-of-file sector
BREAK	Returns to File Utilities menu
Shift-BREAK	Returns to main menu
@	Enables DECRYPT mode (see chapter 2). You may need to press SHIFT-@ depending on the CASE setting of the keyboard (toggled with shift-0).
H,D,B,O,Q,A	Sets modification mode base to hexadecimal, decimal, binary, octal (O and Q are the same) or ASCII respectively.

The arrow keys are used to page through the file. The right and left arrows or the up and down arrows may be used (in this routine the up and down arrows perform the same function as the right and left arrows) to go through the file one sector at a time. If you attempt to page beyond the limits of the file, you will be given an error message.

At any time, you may enter Modification Mode by selecting your input base (H for hexadecimal, D for decimal, B for binary, O or Q for Octal, or A for ASCII) and then pressing M. All the modification controls available in the DISPLAY DISK SECTORS routine of the ZAP utilities are available to you. See Table 2-2.

The screen display of DISPLAY FILE SECTORS is very similar to that of the DISPLAY DISK SECTORS; however, the leftmost column of information is different, as you can see below:

```
    L00|2700 EDB0 2100 0022 8E40 2193 4E11 0040|'.##!.."#@!#n..@
HEX B10|0E0C EDB0 2A11 FFED 587D 53FE 2A20 04DF|..##*.##D[)#* .#
DRV A20|3801 EB22 A040 F921 2853 363A 2370 2336|8.#"#@#!(S6:#P#6
 0  S30|2C23 22A7 4011 5241 2100 4E01 9300 EDB0|,#"#@.RA!.N.#.##
TRK I40|2187 5236 00CD 8F1B 3A1F 44FE 5130 0821|!#R6.##.:.D#Q0.!
 26 C50|5152 CDA7 2818 3B3E 00FE 2A20 3E3A 8E40|QR##(.;>.#* >:*@
TRU /60|CDF4 5321 0000 22A0 4021 0000 22A4 40E0|##S!.."#@!..*#@#
 26 C70|5BB1 402A 1144 DF30 0022 B140 11CE FF19|[#@*.D#0."#@.##.
SEC M80|22A0 40CD 501B 0118 1AC3 AE19 2179 52CD|"#@#]....##.!YR#
 08 D90|8A42 37C3 C657 2169 5218 F411 C362 21A9|#B7##W!IR.#.#B!#
STD  A0|6406 0A73 2372 2310 FA21 0000 22BD 6421|D..S#R#.#!.."#D!
0DD  B0|0000 1188 52CD 5444 20DC 22C9 5311 0000|...#R#TD #"#S...
FPDE C0|7A2F 3211 4211 FFFF 7A32 8F40 1103 007A|Z/2.B.##Z2#@...Z
RSEC D0|A720 C37B FE10 30BE 328E 403C 21BF 6411|# #{#.0#2#@<!#D.
0002 E0|8764 0122 01F5 3A01 0200 54BF 40A7 28D1|#D.".#:...T#@#(.
+00 -F0|04F1 22A7 6436 00EB 7323 7223 EB09 3D20|.#"#D6.#S#R##.=
```

The leftmost column of information first gives you the current modification mode base. Then it displays the drive number, the track number, and the sector number being displayed. Then it displays the data address mark type that it found on the disk (see chapter 2 for details) along with the density of the disk -- 0SD for single-density, 0DD for double-density. The first character will be either 0 or 1, indicating which side of a disk you are viewing. At the bottom, you will see three lines that look like this:

```
FPDE
RSEC
0000
```

FPDE stands for "File Primary Directory Entry." Here it means that the sector you are viewing is allocated in the file's primary directory entry. If it said FXDE, then the sector you are viewing is allocated in the one of the file's extended directory entries. TRSDOS 1.3 does not use FXDE's, so this message should never appear if you are scanning a disk formatted by this particular system.

RSEC stands for Relative Sector. The value then displayed on the third line is the relative sector being displayed as taken from the directory information for that file.

Next to this column is a single vertical column which displays the NAME of the file being viewed, for example,

```
B
A
S
I
C
/
C
M
D
```

Toward the bottom of this column, you will see one of three symbols: -, +, or @. The minus sign ("-") indicates that you have not yet reached the EOF sector of the file; The plus sign ("+") indicates that you have already passed the EOF sector and are viewing a sector that was allocated to the file but not used by it. The @ symbol indicates that you are viewing the EOF sector itself. When this symbol appears, there will be a hex number above it, viewed vertically. This is the first available byte after the end of the file itself. The last byte of the file would be the one preceding this byte.

If this number is 00, it means that the last byte of the file was at relative byte FFH of the preceding sector.

TRSDOS 1.3 and 2.7DD system files need to be handled slightly differently in order to be viewed. This is due to the lack of a standard directory entry for the system files. If you want to view a system file on either of these two systems, for example SYS5, and the TRSDOS disk is on drive 1, reply to the prompt as follows:

FILENAME ? *05:1

If the file is inactive in the directory (that is, it has been killed), a "File not found" message will be issued. This format will work only for DISPLAY FILE SECTORS and only for TRSDOS 1.3 and 2.7DD system disks.

If a disk error is encountered, Super Utility Plus will display a message describing the error and give you the option to retry the I/O operation with the prompt,

R>etry, S>kip, C>ontinuous, N>onstop or Q>uit ?

Pressing R will cause the program to retry the I/O operation. If the error was due to a momentary condition, this is usually sufficient to correct the situation. If the error appears again, you may select the Continuous or Nonstop options, which will force Super Utility Plus to keep trying to read the bad sector until it gets it right or you stop the process. The only way to escape this is a successful I/O operation or by pressing CLEAR, BREAK or SHIFT-BREAK.

If you press S, for Skip, Super Utility Plus will go immediately to the sector display routine, with whatever it was able to read before the error forced it to stop. If it was unable to read anything, the display will show all 00's. If it was successful in doing a partial read of the sector, then whatever it was able to read will be displayed. Note that this may not be the same as what is actually on the disk. An error of any kind should always alert you to the fact that the data in Super Utility Plus' buffers may be unreliable.

Pressing Q for Quit will abort the entire operation and return you directly to the File Utilities menu.

II. COMPARE FILES

This routine will allow you to compare two files byte by byte and see if any mismatches exist. You may find this routine useful if, for example, you have doubts about the integrity of a particular file and wish to check it against another copy. You will be asked to enter the source filename (don't forget the drive number!) and the compare filename. Super Utility Plus will scan the two files and report any mismatches on your screen:

MISMATCH, RELATIVE SECTOR 0000H, BYTE CFH

At the end of the scan, you will be told how many disk errors (due to CRC errors, etc) were encountered, if any, and the total number of sectors in which mismatches were found between the two files.

III. COPY FILES

This routine is a very fast multiple-file copy utility which will let you move any number of files between two disks, the only restraint being the available space on the destination disk. You are asked to supply the source drive number. Super Utility Plus will then read the directory of the diskette in the source drive and display a directory listing on the screen. At the bottom of the screen you will see a filename, followed by its length in grans and sectors, followed by the query, COPY ?

If you press "Y", the filename will scroll upward and you will be presented with another filename. If you press "N" then the filename will be replaced with another one without scrolling. If you hold down either key, it will repeat until all the files have been "tagged" for copy or not.

When the last file has been "tagged," you will be asked for the destination drive. Super Utility Plus will scan that drive and display the number of sectors to be copied (the total number of sectors occupied by all the files tagged for copy) and the number of sectors available on the destination disk. It will then proceed to copy the files over one by one. You may specify the same disk drive for destination as for source, and you will be prompted for disk swaps as needed. However, make sure that both source and destination disks are the same DOS type. This is mandatory for single-drive copy operations.

If a file already exists on the destination disk, Super Utility Plus will display a line of asterisks " ***** " beside the filename as it copies to indicate that that file is being overwritten. If there is not enough space on the destination disk to hold all the files, the routine will display a message to that effect before starting the copy.

After the last file has been copied over, the directory of the destination disk is updated and written back out to the disk.

This routine will place the files starting at the lowest available track/sector on the destination disk and building upward from there, without leaving any gaps between files. It is a

very good routine to use if you wish to "pack" a disk. Where possible, the file copy routine will keep the file in just one extent rather than splitting it up.

If you are copying a file from a TRSDOS 1.3 or 2.7DD formatted disk to a formatted disk created by another system or vice versa, Super Utility Plus will deliberately strip any passwords which may have been set for the source file. The reason for this is that Model III TRSDOS computes passwords differently from other operating systems and if the encoded passwords were copied over, even if you gave the correct password, it would not be recognized. If you have any password-protected files on your source disk, you must reassign the passwords using the ATTRIB command of the operating system on the destination disk after the copy routine is completed, or use Super Utility Plus' CHANGE FILE PARAMETERS procedure.

If you press BREAK during the copy process, you will be returned to the File Utilities menu, and the destination directory will not be updated.

IV. DISK DIRECTORY

This routine will display the directory of the disk(s) specified. You will be asked for the drive number. You may specify more than one, separating each number with commas or spaces. Super Utility Plus will scan the disks in order and display for each one the disk name, date, number of formatted tracks, number of free granules, and number of free directory slots. It will then display all the valid files on the disk, along with commas or spaces. Super Utility Plus will scan the disks in order and display for each one the disk name, date, number of formatted tracks, number of free granules, and number of free directory slots. It will then display all the valid files on the disk, along with their file attributes and protection levels. For example, SIP=7 means a file is a system file, is invisible, and has a protection level of 7.

V. FREE SPACE

Selection of this option will cause Super Utility Plus to scan all mounted disks and display the disk name, date, number of formatted tracks, number of free granules, free space in Kilobytes and number of free directory slots for each disk. The mounted disks must all be readable, that is, they must have been formatted by a TRSDOS or TRSDOS-compatible system (Model I or Model III) and must contain recognizable directory tracks. If a non-standard disk is on any drive, you will be presented with an error message.

VI. OFFSET FILE

This routine will allow you to load a file into memory at one location and cause it to relocate and execute from another location. You may even set up the file so that the TRS-80 interrupts are disabled prior to execution. The file must be in load file format. BASIC program files cannot be relocated.

This routine is useful for making executable files of those programs which normally destroy part of the DOS when they load into memory. Generally, programs written for tape-based systems do this, since they load on top of the DOS resident module. You may move a tape-based (SYSTEM-type) file to disk using the tape-to-disk utilities of your disk system, then use this option to move the file so that it does not destroy the DOS until it is safely loaded into memory. Once the program is in memory, it can be relocated to its normal addresses for execution.

Super Utility Plus will first ask you to supply the filename to be offset. It will then scan the file on disk and tell you the load module range of that file, that is, the present starting address, ending address and transfer (or execution) address of that file.

You will then be asked to supply the new load address, i.e., where you want it to reside when loaded from disk. You will then be asked if you want to add the block move APPENDAGE to the file. This is a short routine which will cause the file to be block-moved to its normal execution addresses after loading into memory. Normally, machine-language files must reside in a particular place in memory in order to execute properly, so if you want the file to execute, you will want to add this appendage.

If you elect to have the appendage added to your relocated file, Super Utility Plus will check to see that there is enough disk space allocated to the file to accommodate it. More disk space will not be allocated to a file by this routine. It will display an error message if insufficient room exists for the appendage routine.

You will then be asked if you want the interrupts disabled or not. Some programs will not execute properly unless the interrupts are disabled. You will have to determine whether the program you are offsetting will work correctly with the interrupts enabled or disabled, and set this option accordingly.

When you press ENTER to complete this last prompt, the file will be written back to disk in its relocated form, and from then on will always load into the new locations when executed from DOS READY level. If you used the block move appendage, it will preserve the contents of the Z-80 registers at load time so that if your program requires them, it will still function correctly.

VII. FILE LOCATIONS

This option will display complete directory information about each file on a designated disk. You will be asked to enter the drive number to be scanned, and you may enter more than one drive number. Each drive specified must contain a standard formatted disk with a readable directory track. The disks will be scanned one by one, and information about each file in the directory will be displayed. The screen dump will pause to give you a chance to examine the information; press ENTER to scroll to the next screen.

For each active file on each disk, five lines of information will be displayed. For example,

```
SYS1/SYS                        SIP=7
FPDE-:0,TRACK=20,SECTOR=05,BYTE=00H,DEC=03H
EOFS = 00005, EOFB = 00H, LRL = 0
EACC = C352H,  EUPD = C220H, GRANS = 01
EXTENTS / 19,00,00006 / EOF
```

The first line gives the file's name and attributes. In this example, the file is SYS1/SYS and has the attributes SIP=7, meaning it is a system file, invisible, with a protection level of 7 ("no access").

The second line gives information about the file's directory entry. FPDE stands for "File Primary Directory Entry," and this entry is located on drive 0, track 20 (the directory track), sector 5, starting at relative byte 00H.

If you were to display this sector on the screen using DISPLAY DISK SECTORS, you would see an entry for SYS1/SYS at that particular location.

DEC stands for "Directory Entry Code," and is the relative byte position of the file's "hash code" in the HIT table. In this example, DEC =03 would mean that relative byte 03 in the HIT sector of the directory contains the "hash code" for SYS1/SYS. The DEC's position in this table is relevant to the position of the FPDE, and this table is used by the DOS to locate files in the directory at high speed, without the need to search through the directory sectors one by one.

The third line gives information about the file's EOFS (end of file sector) and EOFB (end of file byte). The EOFS value is the last sector used by the file (not necessarily the last sector allocated to the file) and the EOFB is the last byte of the file within the EOFS.

The fourth display line gives the encoded values of the passwords. EACC stands for Encoded ACCESS password. It is followed by the two byte hash code of the file's password, if any. EUPD is the Encoded UPDATE password and is the two-byte hash code of the file's update password. Note that the bytes are displayed in LSB/MSB order, which is the way that you would see them on the disk if you viewed the directory using Disk Zap. For example, the hash of a password may be 42E0H, but would appear on the screen as E042H, which is also they way it would appear if you viewed the directory records directly.

Finally, the number of grans occupied by the file is given on this line.

The last line gives the actual location of the file on the disk. The first number is the track where the file is located, the second number is the starting sector (in this case, sector 0) and the third is the length of the file (actually the number of allocated sectors).

On a standard DOS directory, an exceptionally large file may require an extra directory entry, known as the FXDE or File Extended Directory Entry. This is very similar to the FPDE except that the filename is not contained in it, and it is not displayed when a directory is requested from DOS or Super Utility Plus. If such a file was encountered by Super Utility Plus,

additional information would have been displayed, giving the FXDE's directory entry code and extents.

TRSDOS 1.3 does not allow FXDE's, so this display should never appear when a TRSDOS 1.3-formatted disk was being scanned.

VIII. DRIVE STATUS

This option will cause Super Utility Plus to check the status of all active drives in your system and report back on each. Drives which are disabled from the configuration tables, or not powered, will be reported as NOT IN SYSTEM. Drives with no disks but otherwise powered up will be reported as such.

If drives are found to be NOT IN SYSTEM, the settings for them will automatically be changed in the configuration table to reflect this fact. Thus if you had a drive turned off when you executed this procedure and later turn it on in order to use it, you must return to the configuration table and restore it to an active status by changing the active/inactive indicator for that drive (the plus or minus sign in front of the drive number -- see Chapter 1 for full details on configuring).

IX. SECTOR ALLOCATION

This option will let you enter a track and sector number on a mounted disk and will report which file that particular sector is assigned to, if any. You will be asked to supply the drive number, track number, and sector number. The disk to be scanned must contain a readable directory track, since this routine will use the directory information to determine whether the specified sector is assigned to any active file. If the specified sector was assigned to a file that has been killed, it will be reported as unassigned.

X. BUILD FILE

This routine will allow you to create and pre-allocate space on a disk for a file in as contiguous a manner as possible. The pre-allocated space will be noted in the file's primary directory entry, so that when you write to this file later on, using your DOS, you will do so faster since the DOS will not need to keep returning to the directory to find space for a new granule every so often.

You will be asked to supply the filename along with the drive number in standard TRSDOS filespec format. Super Utility Plus will then scan the disk and report back the disk's name, date, number of formatted tracks, number of free granules and free directory slots. It will then ask you to enter the number of granules you wish allocated to this new file. When you press ENTER, the information will be written into the disk's directory.

XI. CLEAR FILE

This routine will request you to enter the filename of a pre-existing file on a disk. It will permit you to remove all the data from that file without actually removing or killing the file from the directory.

You will be asked, "Are you SURE you want to clear it ?" This will give you a chance to change your mind. Enter "Y" to proceed with the operation, otherwise enter "N". If you press "Y", Super Utility Plus will zero out the data in that file. The file will still be present in the directory but it will in effect be empty.

Be careful in using this routine, as there is absolutely NO way of recovering data from a file that has been cleared.

XI. DISK ALLOCATION

This routine will display a disk allocation map of the disk in the specified drive. The tracks will be listed in the leftmost column. To the right of each track will be slots for each granule, with one of four possible symbols: the letter "X", a "." an underscore, or the letter L. A granule slot with the letter X indicates that that granule is in use, that is, assigned to an active file. A slot with a period indicates that this granule is available for use. If an "L" appears, then that granule was locked out by the DOS during the formatting process, perhaps due to a flaw in the disk, and is not available. Underscore characters will appear in those slots beyond the disk boundaries, that is, beyond the last formatted track on the disk. The letter "D" will appear on those granules which are part of the disk's directory track.

XIII. COMPUTE HASH CODE

This routine will calculate the one-byte HIT table hash code for any filespec. You will be asked to supply the filespec (passwords and drive numbers are not required) and the routine will return a one byte hash code in hexadecimal. This code is what is used by the operating system in the Hash Index Table of the directory.

When the operating system must look up a file, it first calculates this code, then goes to the HIT table to locate it. If this code is found in the HIT table, then its position in the table will correspond to the position of the main directory entry in the following sectors. The system is then able to quickly locate files this way, without having to scan through the entire directory each time.

The hash codes produced by this routine, however, are not unique. That is, two or more filespecs, though different, can generate the same code. This is known as a "collision." The system handles the collision by first checking with the main directory entry and comparing it with the user-supplied filename. If the two do not match, then the system goes back to the HIT table to continue the search.

XIV. COMPUTE PASSWORDS

This routine will allow you to either encode or decode passwords using the algorithms employed by the DOS. When this option is selected, you will be asked whether you want to Encode or decode a password. To encode a password, simply enter E. You will be asked for the password to encode. Type it in, and the routine will return the 2-byte hash of that password. The algorithm used by this routine will depend on the configuration of drive 0. If drive 0 is configured for TRSDOS 1.3 (T3D) then the TRSDOS 1.3 algorithm will be used; otherwise the standard algorithm will be used. You may have to enter the configuration routine to all the settings for drive 0 to get a correct encode of your password. There are no overrides for the compute passwords routine.

To decode a password, enter D. You will be asked for a filename. This filename must exist on one of the mounted disk. When you enter the filename, Super Utility Plus will read that file's directory entry into memory. It will then display the two hexadecimal bytes which make up the ACCESS password. At this time you will see a furiously-changing graphics character at the bottom of the screen. This indicates that Super Utility Plus is in the process of decoding that password. When it finds a password which translates into the correct hash code, it will display it, and go to work on the UPDATE password. Again the graphics character will appear. When it successfully decodes the update password, it will display it.

If no password is displayed for either the update or access passwords, it usually means that the two-byte hash is a result of BLANK characters, meaning there is NO password.

Pressing CLEAR while the graphics character is on the screen will abort the operation and proceed to the next step.

With the exception of TRSDOS 1.3 and 2.7DD, all disk operating systems use the same password encoding algorithm. TRSDOS 1.3 and 2.7DD, however, use a different algorithm. This means that a password which encodes one way on other systems will not encode the same way on these two systems. Also, the TRSDOS 1.3 algorithm is written in such a way that there is one uncodable value -- 0000H. While certain combinations of characters will produce a 0000H hash encode under TRSDOS 1.3, TRSDOS automatically increments the value by 1 when this happens. The 0000 byte pair is used as a protection scheme on certain TRSDOS files, to prevent access.

This difference can produce certain problems, most notably that when files are copied from one system to the other, formerly valid passwords suddenly stop working, and files which had no passwords at all suddenly acquire them. The easiest way to deal with this problem is to strip the passwords from the files after the transfer. Super Utility will strip any passwords during a Copy Files operation, so this is not normally necessary.

CHAPTER 10 - MESSAGES

Below is a list of the messages that Super Utility Plus may generate at various points, along with a brief explanation of each.

R>ETRY, S>KIP, C>ONTINUOUS, N>ONSTOP OR Q>UIT?

When a disk I/O error is encountered, in most cases, you will be presented a chance to retry the operation. This prompt gives you several options. R will retry the operation once; C will retry the operation over and over until it succeeds; N is the same as C but no error messages will be displayed during the process; S will skip the portion of the disk where the I/O error occurred; Q will abort the operation completely and return you to the menu.

DRIVE n DE-ACTIVATED

The specified drive has been disabled in the Super Utility Plus configuration table. Change the +/- setting for this drive to bring it into the system.

NO DISKETTE IN DRIVE n

Self-explanatory

DRIVE TIME OUT !

The drive shut down before the I/O operation could be completed.

INTERRUPT ON PENDING COMMAND

A disk I/O operation was interrupted while a command to the floppy disk controller was still pending.

WRITE PROTECTED DISK

The diskette has a write-protect tab on it, or the drive has been declared write-protected in the configuration tables.

HARDWARE WRITE FAULT

An attempt to write to a disk failed due to a defect in the drive or controller hardware.

SECTOR NOT FOUND

Super Utility Plus attempted to read a sector that either was not there (unformatted) or had a non-standard and unreadable format.

ID CRC ERROR

The CRC byte for the track and sector ID fields was wrong. Usually indicates a flawed format.

DATA CRC ERROR

The CRC bytes for the sector data were wrong. This may have several causes: the sector is a "protected" sector, or the disk drive may be at fault. If it is intermittent, suspect a drive fault (e.g., a worn head pressure pad).

DATA LOST

Data was lost during a read/write operation. Usually due to software problems, (the disk transfer code of the program was not fast enough to keep up with the floppy disk controller).

DRIVE DROPPED READY

A selected drive dropped its "ready" status bit before the I/O operation could be completed.

DISK READ ERROR

An attempt to read a disk failed.

DISK WRITE ERROR

An attempt to write to a disk failed.

WRITE FAULT

Usually signals a hardware problem with the disk drive's write circuitry.

DATA LOST ON TRACK WRITE

While writing data to a track, the timing was off by a sufficient amount so that data was lost before it could be written. This may happen if the TRS-80 CPU has been slowed down.

nn SECTORS NOT COPIED

During a copy sectors operation, several sectors could not be read for some reason or another. This message will usually appear in conjunction with some other error message pinpointing the cause of the failure.

nn SECTORS NOT ZEROED

During a "Zero sectors" operation, several sectors could not be written to. Possible causes are a hardware drive fault, or incompatible ("protected") formats on the target sectors.

nn SECTORS NOT LOADED

Several sectors could not be read. Possible causes are the same as above.

nn SECTORS NOT WRITTEN

Sectors could not be written during an I/O operation. This message will usually be accompanied by other error messages pinpointing the cause.

ERROR ON TRACK WRITE

An error occurred during an attempt to write memory to a disk track. May indicate a hardware problem.

TRACK READ ERROR

An error occurred during an attempt to read a track into memory.

TRACK IN BUFFER FROM xxxxH TO yyyyH
HIT <ENTER> TO DISPLAY

Appears on completion of a successful Track-to-Memory operation. The range of addresses occupied by the track data will be given. Press the ENTER key to view the track data.

nn GRANULES LOCKED OUT

Reports the total number of unusable granules on a disk following a format operation.

CANNOT WRITE DIRECTORY !

During any I/O operation which requires the updating of the diskette's directory, Super Utility Plus was unable to do the update. This may occur, for example, if the target disk is write-protected.

nn SECTORS LOST

During a copy sectors or move sectors operation, if some reason prevents one or more sectors from being successfully written to, this message will appear at the end of the operation.

nn SECTORS COULD NOT BE VERIFIED

Reports the total number of sectors that could not be verified during a Format or Verify Disk Sectors operation.

NON-STANDARD FORMAT

An attempt was made to read a disk that does not have a TRSDOS-compatible format.

DIRECTORY UNREADABLE

The directory is non-standard, or damaged.

CANNOT LOCATE DIRECTORY ON DRIVE n

Super Utility Plus could not read the directory on the specified drive. Usually indicates that the directory track has been written with incorrect or incompatible data address marks.

CANNOT LOCATE DIRECTORY! TRACK?

The program could not locate the directory and is requesting the user to specify its location.

A>LLOCATION TABLE OR E>NTIRE SECTOR?

During a GAT repair, Super Utility Plus may repair only the track allocation table, or alternatively, the entire sector including the disk name, date, auto command, etc. If you specify E for entire sector, Super Utility Plus will insert its own data for disk name, date, etc.

GAT TABLE IS BAD !

The Granule Allocation Table (GAT) has been found to be damaged or incorrect by the Check Directory routine.

HIT TABLE IS BAD !

The Hash Index Table (HIT) sector has been found to be damaged or contain errors by the Directory Check routine.

nn TOTAL ERRORS

Reports the total number of errors found during a directory check.

BAD EXTENTS !

A file has been found to have incorrect extents in the directory entry.

BAD BACKWARD LINK !

One or more of a file's Extended Directory Entries (FXDE) has been found to not point back to the preceding FXDE (or FPDE).

TRACK ALLOCATED !

An attempt was made to move a directory track to a track already occupied.

DIRECTORY THERE !

An attempt was made to move a track to the place occupied by the diskette directory.

SOURCE ?

Prompt for the source drive for a sector or file copy or a backup operation.

DESTINATION ?

Prompt for the destination drive or drives for a file copy or backup operation. You may enter more than one destination drive, separated by commas or spaces.

MOUNT DESTINATION DISKETTES

Prompt to mount the diskettes in all the specified destination drives. Press ENTER to proceed with operation.

NO DESTINATION DRIVES !

No destination drives were specified for a file or sector copy or backup operation.

UPDATING DIRECTORY TO nnn TRACKS

This message will appear as Super Utility Plus modifies the disk directory's GAT table to reflect the added space if FORMAT was used to increase the track count on a diskette, or if a backup was performed between two disks with differing track counts.

CANNOT UPDATE DIRECTORY

An attempt to update a diskette directory failed for some reason.

DRIVE x MOUNTED AND READY

Message returned by the Drive Status routine for all drives whichare powered on and have diskettes mounted.

DRIVE x NOT IN SYSTEM

Message returned by the Drive Status routine for all drives which are either not physically present, not powered, or disabled in Super Utility Plus's internal configuration tables.

OPEN DOOR ON DRIVE x

Self-explanatory.

INVALID FILESPEC !

The user entered a filespec in invalid or non-TRSDOS format. The correct format for a filespec is "FILENAME/EXT.PASSWORD:D." See your operating system manual for more details.

FILE NOT FOUND!

The specified filespec was not on the disk directory.

NEXT SECTOR OUT OF RANGE,
POSITIONED TO SECTOR xxxxx

During a Display File Sectors operation, the user attempted to page beyond the boundaries of the file being viewed.

SECTOR NOT ASSIGNED TO ANY FILE

Message returned when the sector specified in a Sector Allocation scan is not assigned to an active file. It may contain data from a KILLED file, however.

FILE IS NOT IN LOAD FORMAT !

An attempt was made to offset a file that is not in the correct load module format or was not a machine-language file.

ADD APPENDAGE?

Prompt to the user to specify whether or not the block move appendage is to be added to a file that has been offset from its normal load addresses. Reply "Y" or "N".

MISMATCH, RELATIVE SECTOR nnnnn, BYTExx

Indicates the position of a mismatch during a file compare or sector compare operation.

nn DISK ERRORS
nn SECTOR MISMATCHES

Reports the total number of disk I/O errors encountered, and the total number of sectors in which mismatches were found, upon completion of a file compare or sector compare operation.

EOF DEST FILE REACHED !

During a file compare operation, the end of the destination file was encountered unexpectedly.

*****

During a file copy operation, any file on the destination disk which has the same name as a file being copied over will be overwritten with the new data.

FILE ALREADY EXISTS !

An attempt was made to build a file with a name that already is in the diskette directory.

NO SPACE AVAILABLE !

An attempt was made to BUILD a file larger than the available space on a diskette, or to copy files onto a diskette with no free space remaining.

GRANS TO ALLOCATE ?

Prompt to the user to specify the size of the file being built. Enter the number of grans that the file is to be allocated.

CANNOT ESTABLISH DISK TYPE

Indicates that Super Utility Plus is not able to recognize a disk as being formatted by one of the valid DOS types.

APPENDIX A

Some Common Questions Answered

My Super Utility Plus disk sometimes has trouble booting up in my 35-track drive. What is the problem?

Your disk drive's read/write head may be slightly out of alignment. Head alignment is more critical with SU+ disks than normal disks, due to the presence of an 80-track loader track which resides BETWEEN tracks 0 and 1. If your drive's head is out of alignment even slightly, it will pick up enough garbage from this additional track to abort the boot process. You should get your drive's alignment checked.

In the table of DOS specifiers, several are given for each DOS type! Which one do I use?

Whichever one you like. All the items under the Model I or Model III columns for a particular DOS type are equivalent.

What DOS specifier should I use for a double density LDOS Model I data disk?

Any one of the Model III specifiers. Double-density LDOS disks (with the exception of those with the SOLE modification) are identical for Model I and III.

The CLEAR-I combination to toggle INKEY input does not seem to work! What is the matter?

It works, but only for those prompts which require only 1 key for a reply. If a prompt requires the entry of more than one key, ENTER must be used to terminate it. You can tell which prompts require only one key and which require more from the number of underscore prompts following the end of the prompt message.

Why can't I read a single-density TRSDOS disk directory on my Model III?

The problem lies in a hardware difference between the two computers. The Model I uses a floppy disk controller (FDC) chip that is different from the one found in the Model III. Model I TRSDOS 2.3 (and NEWDOS 2.1) writes its directory track using an address mark that the Model III FDC simply cannot read. Therefore, when you use SU+ to try and read the directory of a Model I single-density disk on your Model III, SU+ is unable to find the directory.

Is there any way around this?

Yes, there is. SU+ has a procedure in its REPAIR UTILITIES called "Read-Protect Directory." This procedure will take a directory track and write it back out using the correct address marks. If you perform this procedure on a Model I disk while using your Model III, you will make its directory track readable to the Model III FDC, and consequently, to SU+.

You should be aware, however, that doing so will make your disk's directory track unrecognizable to Model I TRSDOS, because it now has the wrong address mark! To restore the disk to its original condition, you must perform the exact same procedure on that disk, but using a Model I computer.

Why won't SU+ read-protect the directory for me?

It will, but you must tell it to. SU+ will not do anything to a disk unless you ask for it to be done.

Will the same problem occur in reading a Model III disk on a Model I?

Not as long as the Model I has a doubler board installed. The doubler board actually uses TWO floppy disk controller chips, one of each type, so all bases are covered.

Does this problem occur with all Model I disks?

No. It occurs mainly with TRSDOS 2.3 and NEWDOS 2.1 disks, which use the address mark that the Model III can't read. The newer operating systems such as LDOS, MultiDOS and DOSPLUS now write their directory tracks using an address mark common to both the Model I and Model III FDC's, so the problem does not arise with disks created by them.

When I copy a file to a TRSDOS 1.3 disk from another operating system's disks, I sometimes get the message "Attempt to read past EOF" or "Attempt to read outside of File Limits" when trying to use the file. Why does this happen?

The problem lies in the way TRSDOS 1.3 encodes a file's length in the directory. All other DOS'es maintain a file's length as the relative number of sectors, meaning that they start counting from ZERO. So if a file takes up 10 sectors on a disk, its length is encoded in the directory as 9. TRSDOS 1.3, however, maintains it as the absolute sector number, so the same file's length is coded as 10 instead of 9!

Super Utility Plus takes the information from the source disk's directory entry during a COPY FILES operation and places it into the destination disk's directory. In some cases -- but not all -- this results in a TRSDOS directory entry that indicates the file is one sector shorter than it really is. Note that the ENTIRE file has been copied, but the directory entry is wrong. This is the reason for the error message returned by TRSDOS when you try to use the file.

Why doesn't SU+ do the necessary correction?

It would be a simple matter if this happened with ALL files, but it doesn't, and the amount of code needed to determine whether it WOULD happen on any given file transfer is prohibitively large. There is very little room left for added code in the SU+ program, and it would not be possible to include such a routine.

So I'm up the creek on this one?

Not necessarily. See Appendix B for a step-by-step procedure on how to alter the directory entry of an affected file in TRSDOS 1.3.

Why can't I copy files to or from a double-sided disk?

The reason is that when copying files from one disk to another, SU+ needs to buffer the directory track from both the source and destination disks in memory. In order to buffer two double-sided double density directories, 18 K of memory would be required. This is over ONE-THIRD the total amount of RAM memory in a TRS-80, and we haven't even talked about putting aside buffer space for the data itself yet! There is simply not this much free memory available when SU+ is running.

APPENDIX B

Altering TRSDOS 1.3 EOF pointer entries

When a file is copied onto a TRSDOS 1.3 disk from another operating system disk, it will sometimes happen that TRSDOS will not see the last sector of that file. The errors returned by TRSDOS when this happens are Error 28, "Attempt to Read Past EOF," Error 29, "Attempt to Read Outside of File Limits," or Error 34, "Attempt to Use Non-Program File as a Program."

The Cause: In the directory entry for each file, there is one byte which points to the EOF (end-of-file) sector for that file. Most operating systems code this value in a relative fashion, i.e., the first sector is numbered zero, the second is numbered one, and so on. So if a file was ten sectors long, its EOF byte in the directory entry would be 9.

TRSDOS 1.3 does things differently. It codes the EOF byte as the absolute number of sectors occupied by a file, so the EOF byte for a 10-sector file would be 10 instead of 9.

During the COPY FILES operation of Super Utility Plus, information regarding the file to be copied is taken from the directory of the source disk. The file is then copied to the destination disk, and a directory entry built for it using the source disk information. The result is that when the destination disk is a TRSDOS 1.3 disk, the EOF sector byte is coded 1 too low. The entire file will be present on the TRSDOS disk, but the directory entry will be in error.

You should note that even under these circumstances, there are times when you will NOT get an error under TRSDOS when you attempt to use the file. Two conditions must be met before the error occurs: (1) The file's LRL (logical record length) must be 256, and (2) the EOF OFFSET byte must be non-zero. The EOF OFFSET byte is the pointer to the last byte in the last sector which is actually used by the file.

If neither of these conditions are met, then use of the file may not result in the TRSDOS errors mentioned above.

In order for SU+ to correct for this error, a routine would have to be included to check that both of the conditions are met. Unfortunately, at this point there is no more room left in the SU+ program for such a routine.

The Solution: The rest of this article gives a step-by-step procedure on how to change this byte in a TRSDOS directory entry. It should be emphasized that this procedure should not be undertaken indiscriminately. If carried out on files AND the conditions mentioned above are not met, even worse errors could occur.

Hypothetical Situation: You have just copied the file called AXORC/CMD onto your TRSDOS 1.3 disk. On attempting to execute the program from TRSDOS Ready, the system displays Error 28, "Attempt to read past EOF" (For AXORC/CMD, substitute the name of the file you are having trouble with).

Step 1. Boot up Super Utility Plus 3.0.

Step 2. Make sure Drive 0 is configured for a TRSDOS 1.3 disk. The correct configuration information is: T3D,40,17.

Step 3. From the Main Menu, select "ZAP UTILITIES."

Step 4. From the ZAP UTILITIES Menu, select option 1, "Display Disk Sectors."

Step 5. At the prompt for "Drive, Track, Sector," enter: 0,D (the "D" tells Super Utility Plus to go directly to the directory track. Make sure you place a comma or space in between "0" and "D".

Step 6. Using the left and right arrow keys, slowly page through the directory track until you see the entry for AXORC/CMD (or whatever the name is of the file you need to rescue). Here is what it might look like:

```
     00|100A 2A00 0041 584F 5243 2020 2043 4D44|..#..AXORC   CMD
HEX  10|EF5C EF5C 0600 0A42 FFFFFFFF FFFF FFFF|#\#\...B########
DRV  20|FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF|################
 0   30|0000 0000 0000 0000 0000 0000 0000 0000|................
TRK  40|0000 0000 0000 0000 0000 0000 0000 0000|................
 17  50|0000 0000 0000 0000 0000 0000 0000 0000|................
TRU  60|0000 0000 0000 0000 0000 0000 0000 0000|................
 17  70|0000 0000 0000 0000 0000 0000 0000 0000|................
SEC  80|0000 0000 0000 0000 0000 0000 0000 0000|................
 04  90|0000 0000 0000 0000 0000 0000 0000 0000|................
STD  A0|0000 0000 0000 0000 0000 0000 0000 0000|................
0DD  B0|0000 0000 0000 0000 0000 0000 0000 0000|................
     C0|0000 0000 0000 0000 0000 0000 0000 0000|................
     D0|0000 0000 0000 0000 0000 0000 0000 0000|................
     E0|0000 0000 0000 0000 0000 0000 0000 0000|................
+00  F0|2863 2920 3139 3830 2054 616E 6479 2020|(C) 1980 TANDY
```

Step 7. Set the Modification Base to hexadecimal by pressing H. Then enter Modification Mode by pressing M.

Step 7a. If necessary, use the arrow keys to bring the flashing cursor to the FIRST BYTE of the LINE that has the name "AXORC/CMD" (or whatever).

Step 8. Type: G14. DO NOT PRESS ENTER. The flashing cursor will be repositioned to the 14th relative byte from the start of the directory entry. This is the EOF byte.

Step 9. Note the value of this byte. It must be increased by 1. In this example, the value is 06. So type 07. Do not press ENTER yet.

NOTE: Don't forget that you are working with hexadecimal values! The next digit after a "9" is "A", and not "0"!! If the EOF byte value is 39, for example, the next one up is 3A, not 40!!

Step 10. Check your work. Make sure you have it right.

Step 11. Now press ENTER twice. The updated sector will be written back to the disk.

The file AXORC/CMD should now load properly under TRSDOS 1.3.