MISOSYS, Inc.
Post Office Box 239
Sterling, Va. 22170

UTILITY 1 L-32-070
Model 1/3 utilities
Serial Number 200862

MISOSYS, Inc.

UTILITY DISK #1

LOGICAL SYSTEMS INC.

C O M P

Compares two files, parts of files, diskettes, or parts of diskettes for a character for character match. The proper syntax is :

```
==========================================================
|   COMP filespec1 TO filespec2 (parm,parm,...)          |
|   COMP :drive1 TO :drive2 (parm,parm,...)              |
|                                                        |
|    The allowable parameters are :                      |
|                                                        |
|   REC=      Starting record number of the filespecs    |
|             at which the compare will begin default    |
|             is Ø                                       |
|                                                        |
|   NUM=      Number of records of a filespec or         |
|             sectors of a disk to compare               |
|                                                        |
|   ALL       Display each non-matching byte             |
|                                                        |
|   PRINT     Send display to *PR as well as *DO         |
|                                                        |
|   CYL=      Cylinder at which to start compare between |
|             two drives (default is Ø)                  |
|                                                        |
|   SEC=      Starting sector of a diskette to compare   |
|             (default is Ø)                             |
|                                                        |
|   Abbr: REC=R, NUM=N, ALL=A, PRINT=P, CYL=C, SEC=S     |
|                                                        |
==========================================================
```

This utility compares two files or two entire diskettes to determine whether or not the information in or on them is identical. It is usually performed after a BACKUP or a COPY to determine the validity of the data.

If the data is identical the following display will result:

```
COMP MAY/DAT:3 :4

COMP - LDOS file or disk compare program Version 1.Ø.Ø
Copyright 1982 By Logical Systems,Inc.


MAY/DAT:3    contains    17 sectors, EOF offset =   7Ø
MAY/DAT:4    contains    17 sectors, EOF offset =   7Ø
```

Notice that the second filespec was indicated by a drivespec. This is the ONLY exception to a complete filespec which is allowed. Since the files proved to be identical, only the number of compared sectors followed by the end-of-file offset were displayed.

In the case of differing files the following would occur :

```
COMP FISCAL82/DAT:3 FISCAL82/DAT:4 (R=4)

COMP - LDOS file or disk compare program Version 1.0.0
Copyright 1982 By Logical Systems,Inc.

Posn= X'0005,00   FISCAL82/DAT:3    = X'20,   FISCAL82/DAT:4    = X'00
    29 bytes did not match.

Posn= X'0005,B0   FISCAL82/DAT:3    = X'54,   FISCAL82/DAT:4    = X'00
    32 bytes did not match.


FISCAL82/DAT:3     contains    18 sectors, EOF offset =  100
FISCAL82/DAT:4     contains    18 sectors, EOF offset =  100
```

The display shows record number of a discrepant sector followed by the relative byte, and the contents of that byte in each filespec. The second line displays the total number of subsequent bytes which do not match. If the ALL parameter had been specifed, each of the sixty-one bytes would have been displayed in the first format revealing the content of both files.

To compare one disk to another use drive numbers instead of filespecs. The starting cylinder and sector number may be specified either in X'00' format or as a decimal integer. The number of contigous sectors to compare may also be specified by using the NUM= parameter.

Unlike file to file comparisons, the disk to disk compare will only display the utility name and return to LDOS Ready if no divergent bytes are detected. If discepant bytes are detected the following will appear on the video:

```
Cyl X'0D, Sec X'00, Byte X'00, Drive 2 = X'6D, Drive 3 = x'31
3078 bytes did not match.
```

If the ALL parameter had been specified then each different byte would display in the first line format. To send the ouput to the printer as well as the video, specify the PRINT parameter.

D C T

Displays Drive Code Table for any one of the eight logical drive numbers to the screen and allows it to be modified. The syntax is:

```
DCT drivespec

no parameters are allowed

drivespec is optional

abbr: colon in drivespec is unnecessary
```

The Drive Code Table (DCT) is the way in which LDOS interfaces the operating system with specific disk driver routines. Typing DCT at the LDOS Ready prompt will enter the utlity. The screen will clear and the following will appear:

```
DCT/CMD - Drive Code Table Creator
Copyright 1982 by Galactic Software Ltd.

Logical Drive Number <0-7> ?
```

If the drivespec had been specified the display would skip to the next display. Enter the drive number desired and the current DCT will appear. For example, if a 0 were entered the following might be displayed:

```
                          Logical Drive 0

        X'4700'=  C3  8D  FE  0C  11  00  97  1F  2F  4C

 X'C3'=  1100 0011          Enabled
 X'8D'=  1000 1101          LSB of Disk Driver
 X'FE'=  1111 1110          MSB of Disk Driver
 X'0C'=  0-0-0-0-1-1-00     SD, 5", Hard, Fixed
 X'11'=  0-0-0-1-0001       SDC, Dcyl=N, Alien DC, Start Head = 2
 X'00'=  0000 0000          Drive Select Code = 0
 X'97'=  1001 0111          Cylinder Count = 152
 X'1F'=  000-11111          Heads = 1, Sectors/Track = 32
 X'2F'=  001-01111          Gran/Cylinder = 2, Sectors/Gran = 16
 X'4C'=  0100 1100          Directory Cylinder = 76

 Do you wish to modify (Y/N) ? -
```

Each logical drive will be entered into the DCT in a 10 byte sequence starting at X'4700'. Each logical drive number will therfore be located at the following addresses:

Drive 0	X'4700'	-	X'4709'
Drive 1	X'470A'	-	X'4713'
Drive 2	X'4714'	-	X'471D'
Drive 3	X'471E'	-	X'4727'
Drive 4	X'4728'	-	X'4731'
Drive 5	X'4732'	-	X'473B'
Drive 6	X'473C'	-	X'4745'
Drive 7	X'4746'	-	X'474F'

The first line of the display indicates which logical drive is being displayed. The second line displays the address of the entry and the subsequent 1Ø bytes located there which comprise the information. In the example, drive Ø is shown at X'47ØØ' follwed by : C3 8D FE ØC 11 ØØ 97 1F 2F 4C

Each of the 1Ø bytes is then displayed in its own row. The first column is the byte to be shown. The second column breaks the byte into the appropriate bit pattern which is actually read by the system. The final column translates the bit pattern into an English mnemonic rendition of the information contained in the pattern. Each byte contains the information necessary to interface hardware of a non-standard type into the LDOS system. The explanation of each byte is as follows:

Byte ØØ

The first byte of a three byte vector to the disk I/O driver routines. In the example, this is an X'C3', which is a jump (JP) to the address following. All enabled drives will have a jump (X'C3') at this byte. If the drive is disabled this byte will contain an X'C9' (RET) instruction.

Byte Ø1 and Byte Ø2

These bytes will contain the jump address of the disk I/O routine which is handling this logical drive. Byte Ø2 is the MSB of the address and Byte Ø1 is the LSB of the address. In the example, the drive routine is located at X'FE8D'

Byte Ø3

This byte contains a series of seven flags as follows:

bit - 7

This is set to "1" if write-protected by software (SYSTEM WP command) or is a "Ø" if not software write-protected. The example, is software write enabled.

bit - 6

This is set to "1" if double density (DDEN) or set to "Ø" for single density (SDEN). The example indicates single density. The third column has an SD in it, which would a DD if the bit were set.

bit - 5

Will be set to "1" if this is an 8" drive and set to "Ø" if it is a 5.25" drive. The last column will translate the bit as 8" or 5"

bit - 4

A "1" will indicate that the disk's second side has been selected. A "Ø" will indicate selection of the first side. The bit value will match the side indicator bit in the sector header written by the Floppy Disk Controller (FDC). The example indicates the first side has been selected. The last column does not translate this.

bit - 3

When set to "1" this bit indicates a hard drive (Winchester) is in place. A "Ø" would indicate that the drive is either a 5.25 or 8 inch floppy. The third column would indicate "HARD" or "FLOPPY" as the translation.

bit - 2

This bit depends on whether the drive is a hard drive or a floppy drive. If it is a floppy drive, this bit will indicate the time delay between selection of a 5.25" drive and the first poll of the status register. The bit is set to "1" to indicate Ø.5 seconds of delay while a "Ø" represents a 1 second delay (can be set by the SYSTEM command). The purpose of the delay is to get the 5.25" drive up to speed if the motor has been shut down. Since the 8" drive motors are always on, this bit is not used on 8" floppy drives. If the drive is a hard drive this bit indicates whether the disk is a fixed or removable type. A "Ø" set indicates that the drive is removable and the bit set to "1" indicates a fixed type. The last column will translate as either ".5 sec" or "Fixed" if "1"; or either "1 sec" or "Removable" if "Ø"

bits - 1 and Ø

These bits will contain the floppy disk step rate specifications for the FDC (SYSTEM command). The stepping rate depends on the size of the drive according to the following chart

bits at	5 inch	8 inch
ØØ	6 ms	3 ms
Ø1	12 ms	6 ms
1Ø	2Ø ms	1Ø ms
11	4Ø ms	2Ø ms

The appropriate value will display in the last column of the screen.

Byte Ø4

This byte contains five more flags which are additional drive specifications

bit - 7

This bit is reserved for future use. Use of this bit may cause compatibility problems with future releases of LDOS.

bit - 6

If this bit is set to "1", the controller is a double density controller (DDC). If it set to "Ø" the controller is a single density controller (SDC). Either DDC or SDC will be translated in the last column.

bit - 5

The information carried by this bit is dependent on whether the drive is hard or floppy. If floppy, the "1" indicates that the drive is two-sided and a "Ø" means single-sided. This should not be confused with bit 4 of byte Ø3. That bit informs the controller what side current I/O is to be on. This bit shows that the disk is two-sided. If the drive is hard, a "1" indicates that the amount of cylinders on the disk is double the amount indicated in byte Ø6. If "Ø" then no doubling is to occur. The third column will translate a "1" to be either "Sides=2" or "Dcyl=Y" and "Ø" to be either "Sides=1" or "Dcyl=N".

bit - 4

This bit will be set to "1" to reflect an alien disk controller and set to "Ø" to indicate a standard disk controller. Most floppies use the disk controller in their own interface (standard). If the drive is hard, this bit will be "1". The message displayed will be "Alien DC" or "STD DC" respectively.

bits - 3 through Ø

If the drive is floppy, this nibble (half a byte) contains the physical drive address by direct binary conversion (ie. ØØØ1=1, ØØ1Ø=2, Ø1ØØ=4, 1ØØØ=8). The system will only support one bit set at a time. A "Phys #n" will be displayed in the translation column where n is the physical drive number (1,2,4 or 8). If the drive is hard, this nibble represents the starting head number offset from one. (A starting head of 3 would be stored as ØØ1Ø.) Although this is a four bit field, only the 3 least significant bits are used. A "Start Head=n" where n is the starting head number, will be displayed in the translation column.

Byte Ø5

This byte contains the current cylinder position of the floppy drive. Its normal purpose is to store the track register of the FDC. A "Current Cylinder=n" where n is the last accessed cylinder, will be shown for this byte. In case of a hard drive, this byte may show a drive select code and a "Drive Select Code=n" will be displayed, where n is the drive select code.

Byte Ø6

This byte contains the highest numbered cylinder on the drive. Since cylinders are numbered from zero, this byte will represent the total number of cylinders minus one. Therefore, a 35-track would be X'22', 4Ø-track X'27, and 8Ø-track as X'4F'. If the drive is hard and the double bit (Byte Ø4 bit 5) is set, this byte would contain half of the total cylinder count. The display will translate "Cylinder Count=n" where n is the true calculated cylinder count regardless of floppy, hard or double conditions set.

Byte Ø7

This byte contains certain allocation specifications set into two fields.

bits - 7 through 5

This 3 bit field represents the number of heads on a hard drive. The eight possible heads are numbered Ø to 7 in binary (ØØØ-111). To get the number of heads, add one to the binary number (e.g. Ø11=3 +1 = 4 heads). The display will show "Heads=n" where n is the true number of heads, only if this is a hard drive. Otherwise, no display will occur for this field.

bits - 4 through Ø

This 5 bit field contains the number of sectors per track. Since sectors are numbered from zero, add one to find the actual amount of sectors per track. If there were 1Ø sectors per track the field would be X'Ø9'. If the drive is two-sided (Byte Ø4 bit 5) the sectors per cylinder would be twice this number. The display will show "Sectors/Track=n", where n is the true number of sectors per track on the drive. The number of sectors per cylinder is not shown.

Byte Ø8

This byte is broken into two fields which contain additional allocation parameters.

bits 7 through 5

This field contains the number of granules per track (a granule is the minimum number of sectors used for creating or extending a file). Again, add one to compute the actual quantity. A maximum of 8 granules per track is allowed. If two-sided operation is indicated (Byte 4 bit 5), the granules per cylinder will be twice this number. On a hard drive, this is the total granules per cylinder. The display will show the actual quantity in the expression "Gran/Track=n".

bits 4 through Ø

This field designates the number of sectors per granule that was used in the formatting operation. This field must be offset by adding one to the stored value. The display will show the calculated result as "Sectors/Gran=n" where n is the number of sectors per granule.

NOTE

The following formula must be met or the system will be susceptable to crashing:

$$\frac{\text{Sectors}}{\text{Granule}} \quad X \quad \frac{\text{Granules}}{\text{Cylinder}} \quad = \quad \frac{\text{Sectors}}{\text{Tracks X Sides}}$$

The DCT values contained in Byte Ø7 and Ø8 should conform to the standards used by LDOS, as illustrated in the following table. Note also that the values listed are for **TRACKS** not **CYLINDERS**. For double sided drives, the track values must me doubled.

Drive Type :	Sectors / Track	Sectors / Granule	Granules / Track
5" Single Density	1Ø	5	2
5" Double Density	18	6	3
8" Single Density	16	8	2
8" Double Density	3Ø	1Ø	3
*5" Hard Drive	32	16	2
*8" Hard Drive	32	32	1

* May vary depending on hard drive type.

Byte Ø9

This byte contains the directory cylinder number. For any directory access, the system will first attempt to use this value prior to examining the BOOT sector directory storage byte in case the READ was unsuccessful. The display will translate this as "Directory Cylinder=n" where n is the number of the directory cylinder.

Modifying the DCT information

To modify the DCT associated with a given drive, answer the modification prompt with "Y". A series of questions will appear on the screen. By answering these questions, you will be allowed to directly modify the DCT information. The prompts that will appear will vary, depending on the drive types.

For all drive types, the following prompts will appear in the order listed:

Enable or Disable drive <E/D>
Driver Address (in Hex) ?
Software Write Protect <Y/N> ?
Single or Double Density <S,D> ?
Drive Type <Ø = 5", 1 = 8"> ?
Floppy or Hard drive <F,H> ?

For any DCT modification prompt, if the information pertaining to the question does not need to be modified, press <ENTER>, and the particular piece of DCT information will be untouched. Otherwise, answer the prompt with the changes to be made. If an invalid response is entered for a DCT modification prompt, an error message will appear on the screen and the prompt will be repeated. If <BREAK> is pressed in response to any modification prompt, the DCT utility will be aborted. Control will resume at the LDOS Ready prompt, and the DCT information for the drive in question will remain unchanged (no changes made to the DCT information prior to pressing <BREAK> will be reflected).

If the "Floppy or Hard drive" prompt is answered with "F" (or the drive was seen by the system initially to be a floppy drive and <ENTER> was pressed), the following set of prompts will appear.

```
Delay time <0 = 1 sec, 1 = .5 sec> ?
Stepping Rate <0 = 6ms, 1 = 12ms, 2 = 20ms, 3 = 30ms> ?
FDC capable of Double Density <Y/N> ?
One or Two sided operation <1,2> ?
Standard Floppy Disk Controller <Y,N> ?
Physical Drive Address <1,2,4, or 8> ?
Maximum Cylinder count ?
Sectors per Granule ?
Granules per Track ?
Directory Cylinder ?
```

If the "Single or Double Density" prompt is answered with "D" (or the drive is currently configured for double density), the "FDC capable of Double Density" prompt will not appear.

If the "Floppy or Hard Drive" prompt is answered "H", the following prompts will appear in place of the prompts listed above.

```
Removable or fixed <R,F> ?
FDC capable of Double Density <Y/N> ?
Standard Floppy Disk Controller <Y,N> ?
Starting Head Number <1-8> ?
Drive Select Code ?
Maximum Cylinder count ?
Set Double Bit <Y,N> ?
Number of Heads <1-8> ?
Sectors per Granule ?
Granules per Track ?
Directory Cylinder ?
```

After all of the modification prompts have been answered, the following prompt will appear:

Install Modifications (Y/N) ?

To incorporate the changes made into the DCT, answer this prompt with "Y". This will cause the DCT to be modified, and the status display will reappear on the screen with these changes.

To not incorporate the changes made into the DCT, answer the above prompt with "N", and the DCT will remain the same as it was prior to being asked the first modification question. The DCT utility will stop and return to LDOS Ready.

D I R C H E C K

Checks diskette directory for flaws and attempts to fix detected errors. The proper syntax is:

```
============================================================
| DIRCHECK drivespec (parm,parm)                           |
|                                                          |
|      drivespec is optional                               |
|                                                          |
|      The parameters are :                                |
|                                                          |
| P    sends error messages to line printer.               |
|                                                          |
| N    non-stop screen display of error messages.          |
|                                                          |
| R    must be specified if the target drive is a          |
|      Radio Shack Hard Drive.                             |
|                                                          |
| L    must be specified if the target drive is a          |
|      Laredo Hard Drive.                                  |
|                                                          |
| H=   must be specified if the target drive is            |
|      a hard drive with no cylinder locked out. If a      |
|      number is specified also then that cylinder         |
|      is assumed to be locked out.                        |
|                                                          |
| abbr : colon in drivespec is optional                    |
|                                                          |
============================================================
```

To verify the integrity of a diskette at LDOS Ready type:

DIRCHECK

The DIRCHECK (P) syntax will send any error messages to the printer device as well as the video display.

Hard drive directories differ from those found in floppy drives in that they have no lock out table for flawed cylinders detected during formatting. However, certain machines will have cylinders purposely locked out in order to reserve it. It is for that reason that the following parameters are to be used with hard drives.

DIRCHECK (R) or (L) are specified for a Radio Shack or Laredo hard drive. This ignores a GAT error of "Allocated but not used" (see errors below) for cylinder X'Ø1' or X'77' respectively. These cylinders have been deliberately locked out for future use but this establishes the cited error condition.

For hard drives with no locked out cylinders, specify the "H" parameter. If this parameter is modified with a number (H=n : where 1 <= n <= 2Ø2), then the cylinder specified by the number will be treated as a locked out cylinder. Note that this could be used in lieu of the "R" or "L" parms with the proper cylinders specified.

DIRCHECK will prompt for the logical drive number containing the diskette that needs to be checked. Enter a number Ø through 7. Pressing <BREAK> at this prompt will return to the LDOS Ready prompt.

If no errors are detected, the message:

Directory Check Complete, Ø Errors

will appear and control returned to LDOS Ready. If any errors are detected a message with the appropriate information will appear. Error messages will be displayed as they are detected. If there are multiple problems within the directory, the messages will pause every 15 lines. To continue, press any key except <BREAK>.

To abort press <BREAK>. If the "N" parameter was specified the video display will not pause. Note that if "P" is specified that "N" is assumed.

After all errors have been detected, the prompt:

Directory Check Complete, X Errors
Attempt to Fix Directory (Y/N)?

will appear where X is the total number of errors detected. Answering with "Y" will cause the utility to TRY and repair the damage. Answering "N" will abort the utility and in either case control is resumed at LDOS Ready. It may be advantageous to run FIXGAT first before repairing with DIRCHECK.

Certain errors cannot be corrected. Among these are: Directory Read Error (due to partial erasure or worn media), two or more files allocated to the same disk space, GAT Read Error or other "hard" errors. The utility will abort to LDOS Ready if any non-recoverable error occurs.

If the FIX prompt is reached, there is still no guarantee of total success. After a fix has taken place, run DIRCHECK again to make certain all errors have been repaired. It is NOT recommended to utilize a diskette with a flawed directory. If a partial reconstruct has occured but some errors remain, recover files by copying to a known good diskette.

DIRCHECK error messages

NOTE

Throughout the list of error messages the phrase "Total Recovery is possible" should be taken with emphasis on the last word. If there is a media flaw or hardware malfunction, there is little that can be done by software to precipitate recovery. What is meant is that chance of recovery is theoretically fairly good.

Unable to Log in Drive - No recovery is possible. Aborts utility.

Insufficient Memory to Check Directory - Most likely to occur on large volume drives that are quite full. No recovery is possible but there is not necessarily anything wrong. The utility had to abort.

The following are all Hash Index Table (HIT) errors.

Missing Hash Code at HIT X'nn' - A file has no entry in the Hash Index Table at hexadecimal table position X'nn'. Total Recovery is possible.

Unnecessary Hash code at HIT X'nn' - A HIT entry exists at position X'nn' with no file. Total Recovery is possible.

Incorrect Hash code at HIT X'nn' - A file has the wrong HIT entry at position X'nn'. Total Recovery is possible.

The following are all Directory Entry Code (DEC) errors.

Non-existent FXDE - > DEC X'nn' - A directory extension should exist at Directory Entry Code X'nn' but does not. No recovery is possible.

FXDE Chain Error - > Extent ddd DEC X'nn' - A directory extension exists at X'nn' that does not point back to the original file. (ddd is the decimal number of the extent.) Recovery is sometimes possible.

Illegal DIR/SYS Entry - the entry in the directory of itself is incorrect. Total Recovery is possible.

The following are all Granule Allocation Table (GAT) errors.

GAT Re-allocation - > Cylinder ddd Gran d - Two or more files think they own the same disk space. Automatic recovery is not possible. As a last resort, kill the offender and run DIRCHECK again. Chances of recovery by this method are rather slim.

Granule(s) allocated in locked out cylinder ddd - During FORMAT unreadable cylinders are locked out to prevent their use. However, certain files have been allocated to these unreadable cylinders. Nothing is done by the DIRCHECK utility to correct the problem but it may be possible to COPY the file to another diskette if in fact the EOF has not progressed into the locked out cylinders.

Granule d of Cylinder ddd, allocated but not used - This means that the granule has been indicated to be in use but in fact, is not. Total Recovery is possible.

Granule d of Cylinder ddd, used but not allocated - This means that a file is actually occupying what is supposed to be available space. Total Recovery is possible.

The following are disk Input or Output errors (I/O).

Directory Sector X'nn' is unreadable. No recovery is possible of any files in that sector.

Directory Write Error - Sector X'nn'. The attempt to write corrected information was not able to be verified. A "Retry Write (Y/N)" prompt will accompany this error. Answer "Y" to try again. If the error is constant, enter "N" to proceed. Recovery is not possible.

F I X G A T

Attempts to repair an unusable Granule Allocation Table (GAT) in the directory of a 5.1.X formatted diskette. The proper syntax is:

```
==============================================================
| FIXGAT drivespec                                           |
|                                                            |
|      drivespec is optional                                 |
|                                                            |
|      There are no parameters                               |
|                                                            |
| abbr : colon in drivespec is optional                      |
|                                                            |
==============================================================
```

If during any disk access, an error is received such as "Illegal Drive Number" and there is a diskette in the drive, the problem could be caused by the GAT sector of the directory containing incorrect information. Other GAT related problems could include improper disk name or master password, incorrect date, or an AUTO command which cannot be accessed. The FIXGAT utility might help in any of these circumstances.

Hardware errors such as disk I/O errors, parity read errors, or directory read errors might be due to worn media, improper formatting, or hardware faults. These latter types of errors cannot be repaired by software.

The purpose and use of the GAT sector are explained in chapter six of the LDOS manual. FIXGAT simply overwrites information contained in relative bytes GAT+X'CB' through GAT + X'FF' of the first directory sector. After a series of prompts, the utility will write the block of information to the section described. It makes no determination regarding the validity of the existing information simply because there can be no guarantee that the stored data is correct.

After a FIXGAT has been performed, therefore, the GAT has only those characteristics which were given as answers to the prompts. If the information supplied was not technically correct then unpredictable results will occur.

If the media or disk drive are not functioning properly then this utility might not function correctly. Pressing <BREAK> for any prompt will return to LDOS Ready. The following chart will describe what happens in detail.

Prompt	Action performed
Diskette Name (Default = XXXXXXXX) ?	Whatever is contained in X'DØ through X'D7' is displayed as Default. To change the diskette name enter a new name. To leave it as is, press <ENTER>.
Date (Default = XX/XX/XX) ?	The diskette date is used as the default unless it is unreadable, in which case the current system date is displayed as the default date. Enter a date in the MM/DD/YY format to change it or press <ENTER>. Bytes X'D8' through X'DF' will be overwritten.
Cylinder Count (Default = xxx) ?	The default is taken from the target diskette which might be totally incorrect if the GAT is defective. BE CAREFUL! Enter the proper cylinder count or press <ENTER>. ENTERING THE INCORRECT COUNT means that some files might never be seen again! Byte X'CC' is overwritten with the proper computed value.

Number of Sides <1,2> (Default = X) ?	All defaults from now on are read from byte X'CD'. Type answer or press <ENTER>. Bit five of byte X'CD' is set or reset based on this response.
Density <S,D> (Default = X) ?	Bit six of byte x'CD' is set based upon this response.
Granules per Cylinder (Default = X) ?	Bits two through zero of byte X'CD' are set based upon this response.

The chart below represents the LDOS default values for the media type and size. In order to determine granules per cylinder, multiply granules per track times the number of surfaces.

Drive Type :	Sectors / Track	Sectors / Granule	Granules / Track
5" Single Density	10	5	2
5" Double Density	18	6	3
8" Single Density	16	8	2
8" Double Density	30	10	3
*5" Hard Drive	32	16	2
*8" Hard Drive	32	32	1

*may vary depending on manufacturer

No answers are required but the following changes happen automatically. Byte X'CB' is set to X'51'. The master password of the diskette becomes "PASSWORD" (bytes X'CE'-X'CF'). Any AUTO sequence is removed (bytes X'E0' through X'FF').

The prompt : "Install Modifications (Y/N)" will now occur. Answer "Y" to implement the changes or answer "N" to abort the write.

It must be realized that FIXGAT only changes certain "signs" in the GAT. Changing the "sign" does not change the reality. It is not correct to assume that this utility will change a single to a double density diskette or the alter the number of cylinders etc.

After FIXGAT has been used, the DIRCHECK utility should be used. After all possible repairs are completed, MAKE A BACKUP by class ($:s TO :d) IMMEDIATELY. Note that a mirror image backup would merely reproduce the defective directory to the destination diskette. The repaired disk is, in all probability, incapable of properly allocating files especially if any locked out tracks were present on the disk.

H I G H

Displays upper memory allocation as well as highest usable memory location. The syntax is:

```
=============================================================
|    HIGH                                                    |
|                                                            |
|    no parameters are allowed                               |
|                                                            |
|    abbr: NONE                                              |
|                                                            |
=============================================================
```

The HIGH command reports the addresses of any upper memory modules which affect HIGH$. The screen will display HIGH$ followed by the name of a properly headed file and its start address in hexadecimal format. The presence of a high memory module without a header will be detected and displayed with the name of unknown.

To use this command, type HIGH at the LDOS Ready prompt. The following is an example of the output.

```
X'F55D'    HIGH$
*********************************
X'F55E'    Unknown
X'F844'    RTEPR
X'F96E'    SLASHØ
X'F993'    BOLD
X'F9D7'    KSM1
X'FAC8'    JKL1
X'FB1E'    TYPE1
X'FCD5'    KI1
X'FE4Ø'    BLINK
X'FE8D'    LSIHD
*********************************
```

In order to get user module names to display when using HIGH the following convention should be utilized at the beginning of the code.

```
        ENTRY    JR      START    ;branch to actual start of program
                 DW      nnnnH    ;where nnnn is HIGH$ prior to loading this module
                 DB      N,'name';N= len of name in quotes (e.g. DB 5,'LSIHD') and the
                                  ;name is the name of this module
        START                     ;actual start of code
```

Using this format will cause the "name" of the module, as well as the load address to appear in the HIGH report.

M A K E

Allows the creation of a disk file filled with a character of choice. The proper syntax is :

```
MAKE filespec (parm,parm,...)

The allowable parameters are :

REC=      Number of records to be in the file. May be
          specified in hex (X'0'-X'FFFE') or decimal
          (0-65534)

LRL=      Length of each record in either hex or
          decimal. 256 is assumed.

SIZE=     Desired size of file in K. (1024 bytes). Do
          not use this parameter when using REC or LRL

FILL=     Fills each record with a specified byte of 0
          to 255. May also be X'00' to  X'FF'.

CLOSE=    Closes the file. The default is NO.

CREATE=   Switches the create bit in the Directory.
          The default is yes.

Abbr: REC=R, LRL=L, SIZE=S, FILL=F
```

The MAKE utility differs from the CREATE utility in two primary ways. MAKE can be set not only to reserve room for a file so that it will not be overwritten by another file, but can also write an optional character to every position in the file. Thus, if a neutral or non-ASCII character were written, it would be an easy matter to detect whether the diskette space had ever been used for the file.

This can also cause the overwritting of the random characters found on a used diskette to one predefined character so that random data does not enter a file reading routine.

The following examples show how to accomplish this :

```
MAKE AR/DAT:0 (R=1000,L=64,F=X'00')

MAKE AR/DAT:0 (S=64,F=X'00')
```

Both examples would write 64K of X'00' to the entire file. However, the file would be listed in the directory as having zero records. MAKE provides a means of CLOSING the file with the exact number of records or the size specified.

```
MAKE AR/DAT:0 (R=1000,L=64,F=0,CLOSE)
```

By using the optional CLOSE parameter, a file is extended to the last specified record rather than the space merely reserved.

The CREATE option resets the create flag bit in the directory. This means that the space allocated to the file will "shrink" down to the last record accessed when the file is closed. Ordinarily, LOF is determined (Random Access) by the highest record written. Upon closing the file, the directory would indicate the space allocation to that file based on that record.

With the create flag set, however, the allocation of space is determined by the file opening under the MAKE command. In this case, space allocation does NOT depend on upon the highest record written.

For example, suppose that the space allocated were for 1ØØØ records and the create flag disabled. If the highest record written when the file was accessed for the first time was record 3ØØ, then the directory would indicate that the file had 3ØØ records with the appropriate space. Had the create flag been set, however, the file at that point would still have space for 1ØØØ records.

It is assumed that any MAKE file has the create flag set. In order to prevent the flag from being set, specify the CREATE option.

```
MAKE (S=64,F=191,CREATE=N)
```

The file will dynamically allocate based upon the highest written record after the initial access. The space will no longer be reserved at that point.

M A P

Lists the allocation of existing or some deleted files by extent, cylinder, sector and granules reserved for its use. The proper syntax is :

```
==============================================================
|     MAP filespec (parm,parm,...)                           |
|                                                            |
|     The allowable parameters are :                         |
|                                                            |
|     P         Causes the listing to be sent to the *PR     |
|               device instead of the *DO device             |
|                                                            |
|     DEAD      Means that the filespec is no longer         |
|               active but once existed on the diskette      |
|                                                            |
|     Abbr:  DEAD=D                                          |
|                                                            |
==============================================================
```

The MAP utility will show or print the position of a file on a diskette. The file can be any valid filespec or in cases where the directory entry has not been overwritten, a file that has been deleted through use of either the KILL or PURGE commands.

At LDOS Ready type MAP with a filespec. If the output is desired in hard copy, specify the (P) parameter. If the filespec has been deleted the DEAD (D) parameter must be specified. A "File not Found" error can occur if the DEAD parameter is used for a current file or if the deleted file is no longer a directory entry.

An Example of the output is :

```
MAP EXAMPLE/BAS

MAP  -  File Mapping Utility  Copyright (C) 1982 by  Logical  Systems Incorporated
Version 1.Ø

EXAMPLE/BAS:Ø       29-Jul-82  LRL=256  Recs=89

Extent #1     Cylinder X'92'  Sectors X'ØØ' - X'1F'    xx

Extent #2     Cylinder X'2Ø'  Sectors X'ØØ' - X'ØF'    x#

Extent #3     Cylinder X'26'  Sectors X'ØØ' - X'ØF'    x.

Extent #4     Cylinder X'2B'  Sectors X'ØØ' - X'ØF'    x*

Extent #5     Cylinder X'2C'  Sectors X'1Ø' - X'18'    .x
```

The number of extents and cylinders used per extent could vary. On LDOS there are up to four extents per directory entry. The more extents a file contains the slower the access time. Reduction of the number of extents can usually occur by using COPY or BACKUP by class when the destination diskette is nearly empty.

Following the sectors on the MAP listing is the granule use indicator which is an extension of the one normally seen by using the FREE drivespec command. In the first line of the example this is "xx". This means that cylinder 92 has two granules each of which are allocated to the file being mapped. The following convention will be used in this section of the display.

symbol	meaning
.	indicates a free granule
x	indicates that the granule is allocted to the current file
#	indicates an occupied granule belonging to another file
*	indicates a locked out granule
?	indicates a granule which is in fact occupied by the file but is not allocated to it in the Granule Allocation Table (See DIRCHECK)

In the example, extent number one starts in cylinder X'92', sector X'ØØ' and uses space contiguously through sector X'1F'. The number of sectors per cylinder is dependent on the size and type of media. Each cylinder is divided into granules again dependent on the size and type of media.

Drive Type :	Sectors / Track	Sectors / Granule	Granules / Track
5" Single Density	1Ø	5	2
5" Double Density	18	6	3
8" Single Density	16	8	2
8" Double Density	3Ø	1Ø	3
*5" Hard Drive	32	16	2
*8" Hard Drive	32	32	1

*This may vary on hard drives.

To derive the number of granules per cylinder multiply the number of granules per track times the number of sides (e.g. on a two sided 5" double density drive there would be 6 granules per cylinder).

RAMTEST

Tests user memory from X'4000' to the end of RAM. The syntax is:

```
==============================================================
| RAMTEST                                                     |
|                                                             |
|     no parameters are allowed                               |
|                                                             |
| abbr: NONE                                                  |
|                                                             |
==============================================================
```

The RAMTEST utility performs a read/write/verify test to memory. It consists of two phases of four tests each. The first phase is HIGHTEST and does four passes on memory locations from X'8000' to the end of memory. HIGH$ is not respected and anything in high memory will be overwritten. After completion of pass four, the second phase, LOWTEST will automatically execute. This phase tests memory from X'4000' to X'7FFF'.

Anything in this area is overwritten. After completion of the fourth pass, the test recycles back to HIGHTEST.

An alive bug is displayed in the upper right corner of the screen to indicate activity of the utility. The current phase and pass are updated on the screen as well.

If an error is detected, the screen will show:

BAD RAM at location X'nnnn'

where nnnn is the hexadecimal address of the failure. At this time the test will cease. Make note of the address and repeat the test. If the test fails at the same location then there is more than likely a problem. If the test fails at random locations, check power source, clean connections, and repeat the test.

If no error is detected the test will loop indefinitely. In order to exit the test, RE-BOOT the system by pressing the appropriate reset switch.

RDTEST

Reads the entire diskette starting at cylinder zero sector zero through the last sector of the last cylinder to see if the diskette is totally accessable. The proper syntax is :

```
==============================================================
|     RDTEST :d (parm,parm,...)                              |
|                                                            |
|     The allowable parameters are :                         |
|                                                            |
|     PRINT      Causes the error listing to be sent to the  |
|                *PR device instead of the *DO device        |
|                                                            |
|     T=         Sets the number of times that an entire     |
|                test is to take place. Default is 1.        |
|                                                            |
|     Abbr:  PRINT=P                                         |
|                                                            |
==============================================================
```

The RDTEST will force a read of every existing non-locked out sector of a specified diskette. Any unreadable sectors will be reported to the video display unless the P parameter is specified.

R E A D I I

Allows transfer of files from a Model II 2.Øa TRSDOS formatted diskette to an LDOS formatted diskette. The proper syntax is :

```
================================================================
|    READII :s  :d (parm,parm...)                              |
|    READII partspec w/wcc:s TO :d (parm,parm...)              |
|    READII -partspec w/wcc:s TO :d (parm,parm...)             |
|                                                              |
|    :s      the SOURCE (Model II) drive # (may not be Ø)      |
|    :d      the DESTINATION (LDOS) drive #                    |
|                                                              |
|        The allowable parameters are :                        |
|                                                              |
|    DIR     Indicates that only a directory will display      |
|            and no files will transfer                        |
|                                                              |
|    SYS     Indicates include system files                    |
|                                                              |
|    OLD     Indicates only those files already existing       |
|            on the destination disk                           |
|                                                              |
|    NEW     Indicates only those files not already            |
|            existing on the destination disk                  |
|                                                              |
|   QUERY    Will prompt for confirmation prior to the         |
|            transfer of each file. The switch ON or OFF may   |
|            be specified. ON is assumed                       |
|                                                              |
|   Abbr: DIR=D, SYS=S, OLD=O, NEW=N, QUERY=Q                  |
|                                                              |
================================================================
```

This utility will transfer files from a Radio Shack TRS-8Ø Model II formatted diskette version 2.Øa to an LDOS 5.1 formatted diskette. Naturally an 8" disk drive on a Model I or Model III is a prerequisite.

All non system files will transfer over unless limited by one of the parameters or by specifying a partspec or a not-partspec. System files will transfer only if the SYS parm is specified.

Using the DIR parm will simply read the Model II diskette's directory and no file transfer will occur. All other parameters and/or partspecs or exclusions may be used in conjunction with the DIR parameter.

Unless QUERY=NO is specified, READII will ask for approval to transfer each file to the destination diskette. Answer the prompt with <Y> to move the file or press <N> to bypass it.

A filespec/partspec or not-partspec (exclusion) may be used to determine which files to move. Wildcard characters are also acceptable. Since LDOS must run the show, drive Ø cannot be the source drive. The following are examples of the use of READII:

READII :3 :1

This example would cause all non system files to move from drive 3 to drive 1. An approval prompt will occur before each file is transferred. If the file already exists on the destination drive, one more confirmation will be required.

READII /DAT:2 :1 (NEW,Q=N)

This example would transfer all files with an extension of DAT whose filenames were not on the destination drive from drive 2 to drive 1. No approval will be required prior to moving any file meeting the criteria /DAT and NEW.

READII $AR/BAS:3 :2 (OLD)

This example would transfer any /BAS file, whose second and third letter were AR, providing it already existed on the destination drive from drive 3 to drive 2. Prompts before each transfer will occur. Since Model II BASIC uses different tokens for BASIC keywords, it is necessary to transfer BASIC programs in ASCII.

READII :3 (DIR)

Reads the directory of a Model II diskette mounted in drive 3.

R E A D 4 0

This utility will allow reading 5.25 inch 35 or 40 track diskettes in an 80 track drive. The proper syntax is:

```
=============================================================
| READ40 :d (parm)                                          |
|                                                           |
| :d  drivespec to be affected (must be a 5" floppy)        |
|                                                           |
| Allowable parameters are as follows:                      |
|                                                           |
| ON or YES     install the driver (default)                |
|                                                           |
| OFF or NO     remove the driver                           |
|               If HIGH$ has not been changed by other      |
|               programs after loading READ40, the memory   |
|               used by the driver will be released.        |
|                                                           |
| TABLE         display current disk drivers in the system  |
|               without loading or removing the program.    |
|                                                           |
| abbr :        YES=Y, NO=N, TABLE=T                        |
|                                                           |
=============================================================
```

After loading READ40, BACKUP, COPY, CONV (using the 40 track disk as the source) and other operations involving only reading may be done in an 80 track drive.

Using the DEVICE or LOG command to log in the directory when a different disk has been inserted in the drive is recommended to improve efficiency, although it is necessary only if an 80 track disk has been accidently read in the drive after loading the READ40 program.

Do NOT use the DEVICE or LOG command if a Model III TRSDOS disk has been inserted for reading with the CONV utility program.

If READ40 detects an error, several re-tries will be performed, then the error message will be displayed and the operator will be prompted to:

<A>bort, <C>ontinue, <I>gnore, or <R>etry

Pressing <A> will exit from any program to LDOS Ready. <C> will return to the running program with the disk error. <I> will return to the running program without flagging the error, although the data is incorrect. <R> will cause the attempted disk access to be repeated.

The <I> option should be used only as a last resort, to allow completion of a COPY or BACKUP. The data for the affected sector will not be correct and will cause problems later if not corrected.

T Y P E I N

Allows character input to the keyboard in any program which does not continuously scan the keyboard for specific characters. The proper syntax is :

```
==============================================================
|    TYPEIN filespec,filespec... (parm,parm,...)             |
|                                                            |
|     The allowable parameters are :                         |
|                                                            |
|    LINES=      Number of lines to be used when TYPEIN is   |
|                incorporated in a JCL file                  |
|                                                            |
|    X1=X'fftt' Xlates hex character ff to hex character tt  |
|                                                            |
|    X2=X'fftt' Xlates a second character ff to tt           |
|                                                            |
|    X3=X'fftt' Xlates a third character ff to tt            |
|                                                            |
|    All parameters as well as the filespec are optional.    |
|                                                            |
|    Abbr: LINES=L                                           |
|                                                            |
==============================================================
```

WARNING: This utility will not work with any program that constantly calls the keyboard driver looking for the <BREAK> key or some other abort key.

TYPEIN allows the user to construct a sequence of characters which then acts to control either the operating system or application programs. The use of TYPEIN is similar to the use of JCL except that TYPEIN can also input to single character keyboard routines such as INKEY$, @KBD, and @KEY. Another difference is that TYPEIN is not compiled and, therefore, has no macros.

TYPEIN can be used in three different ways.

<u>METHOD ONE : Direct input</u>

To use this method at LDOS Ready type : TYPEIN
The following prompt will appear:

```
TYPEIN - Batch mode key entry
Copyright (C) 1982 by Logical Systems Incorporated
```

At this point begin entering the desired command or character sequence. No line can be greater than sixty-three characters in length. Press the <ENTER> key to signal the end of an entry. Press the <BREAK> key to end the session and execute the sequence. Note that the LINES= parameter cannot be used in the direct input sequence.

If it were desired to load and print a document named LETTER/SCR after replacing all references of "Brown" to "Jones" in LSCRIPT, the following command seqence would be used after typing TYPEIN (X1=X'2A1D',X2=X'21B8'):

```
LSCRIPT<ENTER>
<*><L><SPACE><LETTER/SCR><ENTER>
<!><ENTER>
<*><R>Brown>Jones>
<*><P><ENTER>
<BREAK>
```

The X1 parameter translates the character X'2A'(*) to an X'1D' which is needed to get into the "Special Command" mode in LSCRIPT". Similarly, the character X'21' must be converted to X'B8', the repeat command. This manipulation was necessary because the character X'1D' or X'B8' will not be accepted at LDOS Ready.

METHOD TWO : Calling a File

To use this method a file consisting of the character sequence must exist. At LDOS Ready type : TYPEIN filespec. More than one filespec can be used. The utility will append and execute them in order of appearance. Note that only the X parms will work in this method. The utility will then use the file to send characters to the system as if it came from the keyboard.

If the prior example had been saved as an ASCII file called JONES by creating it in LSCRIPT, the following command sequence would cause identical results as were obtained in method one:

```
TYPEIN JONES (X1=X'2A1D',X2=X'21B8')
```

This method would allow TYPEIN to call a JCL file. If an existing /JCL were utilized to archive LSCRIPT files, the previous example could be augmented to utilize it.

```
LSCRIPT<ENTER>
<*><L><SPACE><LETTER/SCR><ENTER>
<!><ENTER>
<*><R>Brown>Jones>
<*><P><ENTER>
<*><S><JONES><ENTER>
<*><L><ARCHIVE/JCL>
<*><R>filespec>JONES>
<*><S><,><A>
<*><END><ENTER>
DO ARCHIVE
```

where ARCHIVE/JCL is :

```
COPY filespec/scr:Ø :2
```

METHOD THREE : from within a /JCL file

This would allow the running of applications which utilize a predictable series of INKEY$ type (single-character entry) inputs. If an LBASIC program existed which utilized the INKEY$ function, normal JCL cannot be used. This can be surmounted by using TYPEIN within the JCL. For example, suppose a list-handling application program were to be executed to print labels. The questions that need to be answered are the following :

```
Which file should be used?
Alphabetic or Zip Code Sort (A/Z)?
Total or Partial file (T/P)?
(If P is chosen)...Enter first element..(then) Enter last element
OK to print (Y/N)
```

If it were desired to regularly print a certain range of Zip Codes, the following ASCII file could be created:

```
LBASIC RUN"MAIL/BAS"
CUSTOMER/DAT<ENTER>
<Z><ENTER>
<P><ENTER>
<53201><ENTER>
<53299><ENTER>
<Y><ENTER>
```

Or several similar modules. This module would be called MKE. The following JCL file called MAIL/JCL, would be an example of this method:

```
FILTER *PR PR/FLT (CHARS=35)
. Print Metro Labels
. Enter 1 for Chicago
. Enter 2 for Los Angeles
. Enter 3 for Milwaukee
//KEYIN Select 1 - 3
//1
TYPEIN CHI
//2
TYPEIN LAX
//3
TYPEIN MKE
///
```

In a real circumstance the operator would now save considerable time by getting the desired ouput from a single command line DO MAIL <ENTER> 3 <ENTER> and essentially allow JCL in concert with TYPEIN to run the computer.

It must be emphasized that this utility is primarily provided to build applications systems around. Existing systems will commonly provide continuous scans of the keyboard for an abort sequence. (eg. "Press @ to stop print") Such scans will, of course, "empty" the TYPEIN buffer.

TYPEIN should not be used for any high memory placement such as filtering, setting, routing, etc. because unlike a JCL, TYPEIN uses high memory. Any high memory placement will, therfore, trap the TYPEIN module in high memory and that memory cannot be reclaimed except by boot or an applicable global reset.

U N K I L L

This is used to re-instate a specified file which was previously deleted by using the KILL or PURGE command. The syntax is:

```
===========================================================
|    UNKILL  filename/ext:d                               |
|                                                         |
|                                                         |
|     no parameters are allowed                           |
|                                                         |
|    abbr:  NONE                                          |
|                                                         |
===========================================================
```

Since KILL and PURGE only reset bits in the directory entries, reset the HIT and GAT and don't erase a file, it is possible to re-allocate file space provided the file hasn't been overwritten by another entry and all space that it previously occupied has not been re-allocated. Therefore, UNKILL should be used as soon as possible after the deletion has occured.

To re-activate a deleted file, type the following command at the LDOS Ready prompt:

UNKILL filename/ext:d

The filespec must be identical to the previous file including any extensions or drivespecs. If no drivespec is specified the command will default to drive Ø but will not search any other drives. If a password existed on the file, it will also be re-activated, although a password is <u>not</u> required to UNKILL it.

An example of the UNKILL command is :

UNKILL ACCOUNT/DAT:1

The previously deleted file called ACCOUNT/DAT on drive 1 will now be re-activated if possible.

Some of the errors that could occur are :

<u>Illegal Filename</u>

This means that an improper filename was used. Check the spelling or syntax and try again.

<u>Directory Read Error</u>

The diskette directory is no longer readable in part or whole. Try again and if that produces the same error, switch drives and try again. If the same error occurs on more than one drive, the diskette is most likely the cause and no recovery of the file is possible.

<u>Directory Write Error</u>

Verification of the attempted write was not obtained. Attempt to recover as if it were a read error.

File Already Alive

An attempt was made to UNKILL an active file. If the deleted file has the same filename as a current file, RENAME the current file and attempt to UNKILL again.

No File By That Name

No such filename is in the specified drive directory. Either the wrong disk was specified or another file has already overwritten the directory entry. In the latter case, no re-activation is possible.

Cannot Unkill File

The name of the deleted file was found but the diskette space allocated to it has already been used by another file. No re-activation is possible.

W R T E S T

Writes the entire diskette starting at cylinder zero sector one through the last sector of the last cylinder to see if the diskette is totally accessable. The proper syntax is :

```
==============================================================
|     WRTEST :d (parm,parm,...)                                |
|                                                              |
|     The allowable parameters are :                           |
|                                                              |
|     PRINT     Causes the error listing to be sent to the     |
|               *PR device instead of the *DO device           |
|                                                              |
|     T=        Sets the number of times that an entire        |
|               test is to take place. Default is 1.           |
|                                                              |
|     Abbr:  PRINT=P                                           |
|                                                              |
==============================================================
```

The WRTEST will force a write of every existing non-locked out sector of a specified diskette. Any unwriteable sectors will be reported to the video display unless the P parameter is specified.

This test will destroy all data on a diskette.