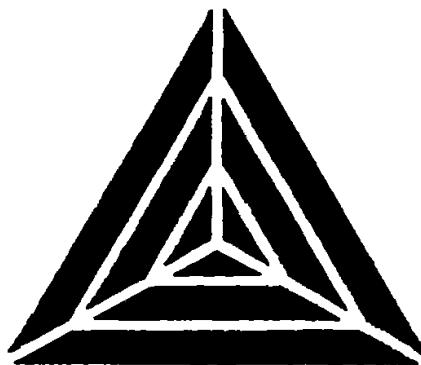


Copyright 1991 MISOSYS, Inc. All rights reserved

# **LDOS 5.3.1**

## **Upgrade Documentation**

**Cat. No. M-10-X33  
LDOS 5.3.1 Upgrade  
to LDOS 5.1.4 for Model I/III**



**MISOSYS INC.**

# MISOSYS, Inc.

---

September 9, 1991

LDOS 5.3.1 Model I Users

Dear Folks:

This letter provides four patches which are for the Model I version of the LDOS 5.3.1 system recently released. The patches are listed on the reverse side of this letter. Although the letter is being sent to all recipients of the Model I 5.3.1, some disks will already have the first two patches applied. Please confirm this by noting the contents of the README/TXT file found on the LDOSXTRA diskette. The following notes the necessity for each patch.

FDUBL1: This is applied to FDUBL/CMD to permit it to operate properly with a Percom type doubler; FDUBL works correctly when used with a Radio Shack doubler. I recommend it be applied even if you are using a RS-type doubler for uniformity.

SOLE1: This is applied to SOLE/CMD to enable it to properly install the booting driver for a Percom-type doubler. SOLE works correctly when used with a Radio Shack doubler. I recommend it be applied even if you are using a RS-type doubler for uniformity.

SOLE2: This alters SOLE so that the entire cylinder 0 is shown as allocated after the double density booting driver is installed. This must be applied if you are using SOLE. After the patch is applied, you must re-SOLE your double-density booting disks to guard against overwriting the boot driver at some future time when your system disk gets full.

SYS011: This is a mandatory patch to SYS0/SYS. I cannot understand how the error caused by an addressing problem during program generation escaped detection. It appears to have been masked by a quirk in the Z80 CPU which caused execution of an invalid Z80 instruction in a manner consistent with a valid Z80 instruction, but under only certain circumstances. In any event, apply the patch, then re-boot for it to take effect.

Sincerely,

(signed)

Roy Solt off

Post Office Box 239

Sterling, Virginia 22170-0239

703-450-4181

## Table of Contents

LDOS 5.3.1 Update.....	1
BOOT-up changes .....	2
LDOS 5.3.1 Installation instructions.....	2
Additions/changes to included files .....	2
Backing up your Master LDOS disk .....	2
Updating your existing disks .....	3
Updating DATA disks .....	3
Updating SYSTEM disks - Model III.....	3
Updating SYSTEM disks - Model I .....	4
Updating Radio Shack HARD disks .....	5
File passwords.....	6
LDOS customer support .....	6
LIBrary command changes.....	7
ATTRIB .....	7
AUTO .....	7
CLOCK.....	8
CLS.....	8
COPY .....	8
CREATE.....	8
DATE .....	11
DEBUG .....	11
DEVICE.....	11
DIR .....	11
DO .....	12
FORMS.....	12
FREE .....	13
KILL .....	13
LIB.....	13
LIST.....	13
MEMORY .....	14
RESET .....	14
ROUTE.....	15
SETCOM .....	15
SPOOL.....	16
SYSTEM .....	16
TIME .....	18
TOF .....	18

Utility Command Changes .....	18
BACKUP .....	18
FORMAT.....	19
LCOMM .....	19
PATCH .....	20
SOLE .....	20
TED .....	21
KI/DVR.....	21
RS232x/DVR .....	21
MINIDOS/FLT .....	21
PR/FLT .....	21
DATECONV.....	22
 BASIC Enhancements.....	23
Single-key edit commands.....	23
INPUT@ .....	23
CMD"V" .....	24
 TED - ASCII Text Editor .....	25
Summary of editing commands .....	25
Invoking TED .....	26
Text entry modes.....	26
Loading a text file .....	27
Entering text.....	27
Cursor positioning manipulations .....	28
Text deletion .....	28
Block operations.....	29
Filing away your text to a disk file.....	31
Text search.....	31
Text search and replace .....	32
Printing text.....	33
Obtaining a Directory .....	33
Exiting from TED .....	33
Text recovery .....	33
 Software License Agreement.....	34
 Limited Warranty .....	34

## LDOS 5.3.1 Update

The LDOS 5.3 release is a major enhancement from the 5.1.x release of LDOS. Although there have been improvements to most library commands, utilities, and BASIC, it is not prudent at this time to provide documentation as replacement pages to your user manual. You are being provided documentation covering the changes since the 5.1.4 release. A completely new user manual may be made available at a future time. Documentation of a technical nature used by programmers has appeared in an issue of *THE MISOSYS QUARTERLY*, available on a subscription basis. There may be a file named "README/TXT" on the LDOS disk (Disk 2 for Model I). If so, this file will contain important information which may not appear in this printed documentation. You should read this file by issuing the command:

### **LIST README**

You can continue to use your 5.1.x disks with version 5.3 as long as you update them as explained in the installation instructions. The following is a brief guide to version compatibility.

Disks from earlier versions of LDOS should be converted with the DATECONV utility program before being used with 5.3. Unconverted disks will work but the time and date information will not be correct. Disks formatted and written to by a DOS other than LDOS should first be REPAIRed (or use CONV) then processed with DATECONV. If a disk written to with 5.3 is used with an earlier version of LDOS (or some other DOS), it may appear to have an unknown password. If the file was previously password protected, access may not be possible without knowing the owner password. To prevent this, use the COPY library command with the (CLONE=NO) parameter when moving files from version 6.3 to earlier versions or other DOS disks.

You should not use an LDOS 5.3 disk with earlier versions of LDOS nor any version 6 TRSDOS except LS-DOS 6.3 (or later). LDOS 5.3 disks may be freely used with LS-DOS 6.3 disks.

## BOOT-up changes

Since version 5.3 now stores a modification time as well as a date, you will be prompted for the time when booting the computer. You can enter the time in the format, "hh:mm:ss" or "hh:mm" or "hh", according to the prompt. If you wish to suppress this time prompt, issue the command:

### **SYSTEM (TIME=OFF, DRIVE=d)**

where "d" references the disk drive containing your BOOTing floppy. The time prompt may be re-activated by the command noted above if "OFF" is replaced with "ON".

### **LDOS 5.3.1 Installation Instructions**

The LDOS 5.3.1 release is an upgrade or the LDOS 5.3.0 and 5.1.4 operating systems. Several important changes have been made to extend and enhance the operating system and its utilities. The date ranging has been expanded to accept dates from January 1, 1980 through December 31, 2011. The directory information for a file will now include a modification time stamp (hours and minutes) in addition to the date stamp. The DATECONV/CMD program is provided to update versions 5.1.4 or earlier LDOS disks to the 5.3 style of dating. Note that this directory convention is that used in LS-DOS release 6.3. The "access" password has been eliminated from the system in order to allow ample directory space for the date and time extensions. The "update" password has been retained. In order to convey a naming convention similar to LS-DOS 6.x, this password is now termed the "owner" password; thus, any reference to the "update" password is assumed to reference the "owner" password and vice versa.

### **Additions/changes to Included files**

The following is a brief description of files added since 5.1.4:

BASIC/CMD	The disk BASIC interpreter previously named LBASIC/CMD
BASIC/HLP	A file of HELP information for BASIC
BASIC/OV4	A BASIC overlay to dump a list of active variables
DATECONV/CMD	A utility to update LDOS disks earlier than 5.3
DOS/HLP	A file of HELP information for DOS
HELP/CMD	A utility used to access the ???/HLP files
SOLE/CMD	A Model I utility to install a DDEN boot driver
TED/CMD	A full Screen ASCII text editor
FDUBL/CMD	A Model I double density disk driver which replaces RDUBL/CMD and PDUBL/CMD

Note: The 5.1.4 files designated LBASIC/OV1, LBASIC/OV2, and LBASIC/OV3 are designated as BASIC/OV1, BASIC/OV2, and BASIC/OV3 under 5.3.1.

### **Backing up your master LDOS disk**

It is recommended that you make several backup copies of the 5.3.1 master disk(s). The simplest way in a two-drive Model III system is to boot up the 5.3.1 disk, insert a blank (or otherwise re-usable) disk in drive 1, and type the command:

**QFB :0 :1**

This procedure assumes that both drives 0 and 1 are 40 cylinder drives. When the copy finishes, you can insert another destination disk and duplicate another copy. Model I users will find QFB on disk 2; thus, you may boot up the 5.3.1 system disk, place QFB in drive :1, issue the command noted above, then replace the LDOS disk in drive :1 with

the blank diskette. Repeat the sequence using the LDOS disk 2 as the drive :0 source disk after the prompt from QFB.

It is important that you do NOT switch between versions 5.3.1 and any earlier version system disk in drive 0 without rebooting your computer. Also, once you have updated a disk with the DATECONV/CMD program, do not write to that disk using an earlier release of LDOS. As noted above, LDOS 5.3.x data disks are completely compatible with LS-DOS 6.3.x data disks. Either may be read from or written to with either LDOS 5.3.x or LS-DOS 6.3.x.

## **Updating your existing disks**

To update your existing LDOS 5.1.x disks, boot up with a fresh copy of your new version 5.3.1 LDOS disk. The LDOS 5.3.1 disk should remain in drive :0 throughout the update procedure. There are two separate methods for updating 5.1 floppy disks, depending on whether they are system disks or data disks. Note that you need do nothing to 5.3.0 data disks.

### **Updating DATA disks**

Place the LDOS 5.1.4 DATA disk in drive 1 and issue the command:

**DATECONV :1**

This completes the conversion of a DATA disk.

### **Updating SYSTEM disks - Model III**

Updating system disks will take two or more steps. Place the 5.1 disk in drive 1, and issue the command:

**BACKUP :0 :1 (INV,SYS,OLD)**

If you want to use any of the new files included with the 5.3 release, move them to drive 1 with a command of the form:

**BACKUP filespec :1**

where “filespec” designates the new file you wish to copy. Once the BACKUPs are completed, issue the command:

**DATECONV :1**

This completes the updating of a SYSTEM disk. Note that Model 4P users must ensure that the file MODEL A/III is on your working LDOS system disk. The MODEL A/III file is distributed with your DOS 6.x disk.

### **IMPORTANT NOTE**

If you have a sysgend configuration file on your old (now updated) SYSTEM disks, you will have to boot that disk while holding down the <CLEAR> key to inhibit the old configuration; then you will have to recreate your desired configuration once again and save it [using SYSTEM (SYSGEN)].

### **Updating SYSTEM disks - Model I**

It is recommended that you do not attempt to update your old Model I system disks. Instead, use working copies of the new LDOS 5.3.1 system disks. If your Model I is equipped with a double density adapter, you may wish to make a double density boot disk using the following procedure:

1. Boot the LDOS 5.3.1 system disk and issue the command:

**FDUBL (Tandy | Percom)**

choosing one of the two parameters depending on the type of adapter present in your machine. Note that FDUBL defaults to the “Tandy” parameter.

2. Place a blank disk in drive :1 and issue the command:

**FORMAT :1 (SYSTEM)**

and respond to the queries with the desired diskette configuration data.

3. Move the files from the system disk to the newly formatted diskette with the command:

**BACKUP :0 :1 (SYS,INV)**

4. Save your configuration by issuing the command:

**SYSTEM (SYSGEN,DRIVE=1)**

5. Install the double density boot file by issuing the command:

**SOLE :1**

6. At this point, remove both diskettes then place the double density DOS disk into drive :0. Insert the LDOS diskette #2 into drive :1 and copy the files from it to drive :0 and re-save the configuration with the commands:

```
BACKUP :1 :0 (INV)
SYSTEM (SYSGEN)
```

7. Make additional copies of this double density system diskette onto blank diskettes in drive :1 by issuing the commands:

```
FORMAT :1 (SYSTEM)
BACKUP :0 :1
```

Note: You cannot use QFB to backup a dual-density diskette.

## Updating Radio Shack HARD disks

For most hard disk installations, the following instructions can be used. They assume a 2-head 5 Megabyte hard drive partitioned as 2 drives (0 and 1) and 2 floppy drives (2 and 3). Other installations should be able to adapt these instructions to their specific hardware configuration.

1. Copy your hard disk driver file from your old LDOS disk to a backup copy of your new LDOS 5.3. For a Radio Shack hard drive, this file is named TRSHD1/DCT, TRSHD3/DCT, or TRSHD5/DCT.

2. Manually invoke the driver with the command:

```
SYSTEM (DRIVE=2,DISABLE,DRIVER="TRSHDx")
```

Respond to any driver queries as you have done in the past. This may include "drive select address" answered as 1; "number of beads for this partition" answered as 1; "starting head" answered as 1.

3. Next, bring up the second head with the command:

```
SYSTEM (DRIVE=3,DISABLE,DRIVER="TRSHDx")
```

As before, this is drive select address 1, 1 head for the partition, but starting head is answered as 2.

4. Now BACKUP all of the files from the floppy SYSTEM disk to the old hard drive's SYSTEM partition (assumed to be head 1 currently assigned to drive 2) with the command:

**BACKUP :0 :2 (S,I)**

5. Update the date convention of the hard drive partitions via the commands:

**DATECONV :2**  
**DATECONV :3**

6. Now switch the hard drive SYSTEM partition to drive 0 and the data partition to drive 1 via the commands:

**SYSTEM (SYSTEM=2)**  
**SYSTEM (DRIVE=1,SWAP=3)**

7. Now install the new configuration onto the BOOTing floppy via the command:

**SYSTEM (SYSGEN,DRIVE=2)**

Note that you may also install any other sysgenable filters and drivers prior to issuing this SYSTEM command (i.e. KI/DVR, MINIDOS/FLT, etc.).

### **IMPORTANT NOTE**

If you ever have to re-FORMAT a hard drive using the TRSFORMx/CMD utility, you will need to use the DATECONV/CMD utility after the formatting is complete. This ensures that the hard drive partition is marked as using the extended dating facility.

### **File passwords**

Passwords have been applied to most files on your LDOS disk. This protection is for your benefit and will avoid inadvertent deletion or overwrite of a needed file. For your information, the following owner passwords have been applied to all password protected files:

<b>SYSTEM</b>	All files with an extension of "/SYS"
<b>BASIC</b>	All files with a filename of "BASIC"
<b>FILTER</b>	All files with an extension of "/FLT"
<b>DRIVER</b>	All files with an extension of "/DVR"
<b>HELP</b>	All files associated with the HELP facility
<b>UTILITY</b>	All other files with an extension of "/CMD"
<b>RSOLTOFF</b>	A master password for all files

### **LDOS customer support**

Support for this LDOS product is available through writing from

MISOSYS, Inc., P0 Box 239, Sterling VA 22170-0239.

## LIBRARY command changes

### ATTRIB

The UPDATE password field has been re-designated as the OWNER password field - this is a name change only. The access password field has been removed from all files, so the ACC parameter is no longer valid. The DOS now assumes a blank access password for all files. If a protection level has not been assigned to a file, full access will automatically be granted regardless of any owner password. The OWNER password will still be required for full access on password protected files that have a protection level other than FULL. To have a file that allows no access whatsoever without the use of the OWNER password, change the protection level to "NO".

```
ATTRIB filespec (OWNER=a,PROT=b,VIS | INV)
ATTRIB :d (LOCK,UNLOCK,MPW= "aa" ,NAME= "bb" ,PW= "cc" )
```

Parameters for filespec attributes:

**OWNER=** the new owner password  
**VIS** makes file "visible" to a directory command  
**INV** makes file "invisible" to a directory command  
**PROT=** the new protection level: [NONE, EXEC, READ, WRIT, NAME, KILL, FULL]

Parameters for disk attributes:

**Lock** Locks the current disk master password onto all visible, non-system files not already protected.  
**Unlock** Removes the password from all visible, non-system files with passwords matching the disk master password.  
**MPW=** Specifies the disk's current master password.  
**NAME=** Changes the disk name.  
**PW=** Changes the disk master password.

### AUTO

This command has had several improvements. You can now install, display, or invoke an AUTO command on any drive - not just the SYSTEM drive. The syntax is now:

```
AUTO (=|?)(:d)(*)(dos-command)
```

= executes the AUTO command on :d.  
? displays the AUTO command currently on :d.  
**:d** the designated drive otherwise drive 0 is used.  
\* Disables the <ENTER> and <BREAK> keys during BOOT.  
**dos-command** any DOS command up to 31 characters in length.

## **CLOCK**

The clock command has been moved to the TIME library command. The new syntax to enable or disable the screen display of the real time clock is:

```
TIME (CLOCK(=ON|OFF))
```

## **CLS**

This command has been added to the library. It clears the video screen. It's syntax is:

```
CLS
```

## **COPY**

When copying from a 5.1.x or earlier version disk to a 5.3 disk, the old access password, if any, will be removed and the 5.3 style date/time information will be established, the time being set to 00:00:00.

Note: COPY should NOT be used to move System (/SYS) files from one disk to another. Use BACKUP for this purpose.

## **CREATE**

New parameters, FILL and SHRINK, have been added. The fill parameter allows you to specify a particular character to propagate throughout the created file. Shrink is used to flag the file for de-allocation of unused space once data is written to the file. To refresh your mind, here is the syntax of this command:

```
CREATE filespec (parms)
```

- |               |   |
|---------------|---|
| <b>Lrl=1</b>  | Sets the Logical Record Length (1-256) [256].   |
| <b>Rec=r</b>  | Allocates r records of LRL length.  |
| <b>Size=s</b> | Allocates s K-bytes of space in K (1024 bytes). SIZE may not be specified if LRL or REC are.              |
| <b>SHRINK</b> | If specified, the file is marked for future de-allocation of space.                                       |
| <b>Fill=f</b> | The created file will be populated with this byte. If omitted, the file is not initialized with any byte. |

## **DATE**

The acceptable range of dates is from January 1, 1980 through December 31, 2011; years 1980 through 1999 are entered as "80-99" while years 2000-2011 are entered as "00-11".

## **DEBUG**

The DEBUG command has been improved to allow for the removal of the extended debugger from resident memory using the command:

**DEBUG (OFF,Ext)**

provided the memory-resident module was the most recent module installed into memory.

## **DEVICE**

The DEVICE command has had both its display and its operation improved. A new parameter, N, has been added and the display has been paged. To refresh your mind, here is the command's syntax:

**DEVICE (parms)**

<b>B=OFF</b>	Suppresses the "byte I/O" portion of the display.
<b>D=OFF</b>	Suppresses the drive portion of the display.
<b>N=ON</b>	Set non-stop display mode. Assumed if P is selected.
<b>S=OFF</b>	Suppresses the options status portion of the display.
<b>P=ON</b>	Duplicates device status to the printer.

## **DIR**

The directory command has been considerably enhanced. The display now includes the time stamp and byte EOF position for each file. Also, certain characteristics of the disk itself are displayed. A replica of this display is:

Drive #	diskname	diskdate	cyl	dh	Free=fffff.f	sssss.s	FI=mmm/nnn
Filespec					Attrib	LRL	#Recs EOF DE File Size MOD Date Time
Filename/ext	SIP*+	pp	lrl	rrrrrr	eof	de	s=sssss.s dd-mon-yy hh:mm

#	drive number
diskname	name of the disk
diskdate	date of the disk
cyl	number of cylinders on the disk

d	disk density: S=single, D=double, H=hard
h	Number of heads on the disk
fffff.f	Amount of free space in K
sssss.s	Total formatted space in K
mmm	Number of free file slots
nnn	Total number of file slots
filename	The name field of a file
ext	The extension field of a file
SIP*+	Attributes indicating: Sys, Inv, Prot, PaDS, Mod
pp	The protection level: NO, EX, RE, WR, NA, KI, FU
lrl	The file's logical record length
rrrrr	The number of records in the file
eof	The position of the last byte in the last sector
de	The number of directory extents occupied by the file
sssss.s	The total space used by the file's records
dd-mon-yy	The date of last modification
hh:mm	The time of last modification, 00:00-23:59

## **DO**

The DO command has been improved. First, parameter strings of any ASCII character are acceptable by enclosing them within double quotes. Second, JCL compilation will search for the first available drive which is not write protected for writing the SYSTEM/JCL file.

## **FORMS**

This command has been added to allow you to display or alter the operating parameters of the PR/FLT forms filter once the filter has been installed. The syntax of the FORMS command is:

**FORMS (parms)**

<b>Default</b>	Sets all parms to defaults.
<b>Addlf</b>	Adds a linefeed after a carriage return.
<b>Chars=n</b>	The number of characters per printed line.
<b>Ffhard</b>	Use 12d for form feed, rather than linefeeds.
<b>Indent=n</b>	Indents n spaces on lines longer than chars [0].
<b>Lines=n</b>	The number of lines printed per page [66].
<b>Margin=n</b>	Sets the left margin [0].
<b>Page=n</b>	Sets physical page length [66].
<b>Tab</b>	Expands tab characters to tab stops [OFF].
<b>XLATE=X'aabb'</b>	Translates aa to bb [00 00].

## **FREE**

The display format of the FREE command has been improved. It will now display the identical disk configuration information as displayed by the header of the DIR command.

## **KILL**

This command has been improved to allow you to specify more than one device specification on the command line. Its new syntax is:

```
KILL filespec | *devspec (filespec | *devspec)
```

## **LIB**

The LIB command has been improved to always produce a correct display regardless of the compressed spaces vs. special characters state of the video driver.

## **LIST**

This command has been considerably revised. The hexadecimal display mode now produces a combined hexadecimal/ASCII display of a complete 256-byte record. The older format is still available by turning off the "compressed" display mode via the "C=OFF" parameter. Both the hexadecimal and ASCII display modes will produce paged displays, stopping when the display screen fills. Depress <ENTER> for the next page or <C> to continue. The "N" abbreviation for the "NUM" parameter has been eliminated; "N" now allows an override to the paged display to produce a non-stop display. This operation is similar to the "N" parm of the old (and new) DIR command. Finally, the "TAB" parameter has been enhanced to accept a tab column number between 1 and 32 with a default of every 8 columns. To refresh your mind, the LIST command's syntax is now:

```
LIST filespec(/TXT) (parms)
```

<b>Ascii8</b>	Will allow full 8-bit output
<b>NUM</b>	Sets line numbering mode for ASCII text.
<b>N=ON</b>	Sets non-stop display mode; assumed if "P" is selected.
<b>Hex</b>	Sets hexadecimal output format.
<b>C=OFF</b>	Turn off compressed display mode for hexadecimal output.
<b>Tab=n</b>	Set tab stops and expansion for ASCII list (n=1-32) [8].
<b>P</b>	Directs output to the printer.
<b>LINE=n</b>	Starts listing at line n in ASCII list.
<b>Rec=n</b>	Starts listing at record n in HEX list.
<b>Lrl=</b>	Sets logical record length in HEX list.

## **MEMORY**

This command has its display improved to report on the modules resident in protected memory. This requires that all installed modules adhere to the LDOS design specifications of memory resident modules which has been a part of the LDOS system since January 1, 1982. An added "P" parameter allows for printing the memory map.

## **RESET**

Parameters LRL and DATE have been added. **RESET filespec (LRL=n)** alters the logical record length of the file to the value n; appropriate values of n are in the range 0-255, with 0 implying a record length of 256. The DATE parameter is used to allow the file's directory date to be altered. Specifying **RESET filespec (DATE=OFF)** restores the file's directory entry to the old-style dating of pre-x.3 release. The old user password field will be set to blanks. This may be useful for files which must be transferred to an older DOS release, such as by the TRSDOS 1.3 CONVERT utility. Specifying **RESET filespec (DATE=ON)** will establish the file's directory date as that of current system date and time.

Note that when this option is specified, the MOD flag will be set to indicate a file change. The syntax is:

```
RESET filespec ((Lrl=n) | (DATE=ON|OFF))
```

## **ROUTE**

This command has been improved in two ways. First, a new parameter, REWIND, has been added which allows you to start the routed file at its beginning. Second, the command has been improved by allowing it to reuse the previously installed but currently unused high-memory support module of a previous ROUTE of the same device. To refresh your mind, the ROUTE command's syntax is as follows:

```
ROUTE *devspec1 (TO) (filespec|*devspec2) ((parms))
```

**NIL**              Specifies that \*devspec1 is routed to a bit-bucket,  
**Rewind**            Resets the file pointer of filespec so existing data can be reread or overwritten.

## **SETCOM**

This command has been added to the library. It allows you to change and/or display the parameters of the RS232 R or T device driver after the driver has been installed into memory. Its command syntax is as follows:

**SETCOM (parms)**

<b>DEFAULT</b>	Set to system defaults shown below in brackets.
<b>Baud=n</b>	sets the BAUD rate [300].
<b>Word=n</b>	sets word length, 5-8 bits [7].
<b>Stop=n</b>	sets stop bits, 1 or 2 [1].
<b>Parity=</b>	sets PARITY; ON, OFF, "EVEN", "ODD" ["EVEN"]
<b>BREAK=n</b>	sets logical break character to n [OFF].
<b>DTR=</b>	Data Terminal Ready [ON]
<b>RTS=</b>	Request To Send [OFF]
<b>DSR=</b>	Data Set Ready [IGNORE]
<b>CD=</b>	Carrier Detect [IGNORE]
<b>CTS=</b>	Clear To Send [IGNORE]
<b>RI=</b>	Ring Indicator [IGNORE]

## **SPOOL**

Three parameters have been added to this command. Pause allows you to temporarily suspend despooling; Resume allows you to continue despooling after a pause; and Clear empties the spooling buffers without printing. The command syntax for these added parameters is:

**SPOOL (Pause|Resume|Clear)**

The spooler has also been improved to free up the memory it uses when turned off, provided the spooler was the most recent module installed. If the memory it used was trapped by a subsequent module's installation, the spooler will re-use its previous memory allocation and inhibit alteration of its previous memory and disk parameters.

## **SYSTEM**

The SYSTEM command, used to select DOS features, has been improved by the inclusion of additional parameters and enhancement of some existing ones. The ALIVE, BLINK, and UPDATE modules can now be removed from memory. The DRIVE parameter now pertains to the SYSGEN parameter so that the configuration may be targeted to a specific drive. The Model III (FAST|SLOW) parameters have been adapted to utilize the Model 4 hardware clock speedup while still maintaining an accurate time clock. Model III users should not specify (FAST) without an appropriate hardware

speedup. A SMOOTH parameter has been added for faster disk I/O involving floppy drives aligned precisely to 300 rpm. A SWAP parameter has been added that, with DRIVE, allows you to switch the logical drive assignments of any two drives even with a JCL command line. The SYSRES parameter has been improved to have the system overlay use only the amount of high memory normally used by the overlay when it is resident in the system overlay region. The TRACE command has been removed as a separate LIBrary command and is now a parameter of SYSTEM. The TYPE parameter was altered to just inhibit the type-ahead operation rather than removing the type-ahead task. Lastly, an option to enable or disable printer time-out has been added. Here is the syntax of the changed parameters:

**SYSTEM (PRTIME=ON|OFF)**

This option is only applicable when the PR/FLT is installed as it uses the printer driver resident within the filter. Specifying **SYSTEM (PRTIME=ON)** establishes an approximate 10-second time-out on printer unavailable. Specifying **SYSTEM (PRTIME=OFF)** disables any time-out; if the printer is unavailable, the DOS will hang! The default is OFF.

**SYSTEM (SMOOTH(=ON|OFF))**

Alters the floppy disk driver so that the system interrupts are disabled earlier than what would otherwise occur. This has the effect of providing faster I/O with disk drives precisely aligned to 300 rpm where extra sector retries would be necessary. Note that when SMOOTH is turned ON, you will not be able to type ahead during disk I/O nor will you be able to effectively use dump-to-disk ON with LCOMM even at 300 baud.

**SYSTEM (SWAP=s,DRIVE=d)**

Exchanges drive "s" with drive "d". The command is functional while JCL is in execution even if one of the referenced drives holds the executing JCL file. If one of the designated drives is the SYSTEM drive and JCL is in execution, the other designated drive must contain a SYSTEM.

**SYSTEM (SYSGEN(=ON|OFF)(,DRIVE=d))**

This command creates or deletes a configuration file. "ON" creates a configuration file on drive d. "OFF" removes the configuration file from drive d. "DRIVE=d" specifies the drive for SYSGEN [default is 0].

**SYSTEM (SYSTEM=n)**

If the designated drive does not contain the DOS and JCL is in execution, the command will abort, otherwise you will be prompted to enter a SYSTEM disk. This command is equivalent to '**SYSTEM (SWAP=n,DRIVE=0)**'.

### **SYSTEM ( TRACE(=ON|OFF) )**

Displays or removes the Z80 Program Counter in the upper right corner of the video display.

## **TIME**

The TIME command has been enhanced to provide for turning on or off the video screen clock display. The syntax of this command is:

```
TIME (hh:mm:ss) ((CLOCK(=ON|OFF)))
```

**hh:mm:ss** Sets Time to hh hours (00-23), mm minutes (00-59), and ss seconds (00-59); "ss" defaults to 00 if omitted.

**CLOCK** Turns on or off the video clock display.

## **TOF**

This command has been added to the library. It will emit a form feed character (12d) to the \*PR device. If the printer is currently unavailable, the command does nothing.

## **Utility Command Changes**

### **BACKUP**

This utility has been changed to allow it to construct a SYSTEM disk from a formatted DATA disk. The VIS parameter has been dropped. The INV parameter has been altered to designate that files invisible to the directory are to be included as well as visible files. The SYS parameter has been altered to designate that system files are to be included in addition to visible files. SYS also reconfigures a DATA disk into a SYSTEM disk by allocating directory entry codes for /SYS files in the hash Index Table. Both Model I and Model III versions of BACKUP can perform a mirror image duplication of a Model I dual-density system diskette provided the destination diskette has been previously formatted with a dual-density configuration.

### **FORMAT**

This utility has been modified to generate a DATA disk after formatting. DATA disks reserve only two file slots out of the total number of directory slots available. SYSTEM disks, configured by the BACKUP utility, reserve 14 additional directory slots for /SYS files. This facility provides 14 additional file slots for DATA drives over that previously available under earlier releases of LDOS.

For Model I systems, the SYSTEM parameter has been added for 5.25" floppy drives to designate cylinder 0 as single-density when the diskette is specified for double density formatting. This creates a dual-density configuration suitable for subsequent use as a booting system disk once the operating system has been moved and the SOLE utility applied.

## **LCOMM**

This communications program has been enhanced via the addition of parameter control over the codes used for XON and XOFF. Also, LCOMM now makes use of the new @CMNDR command-and-return vector of LDOS. This feature provides you the ability to access LDOS library commands while running LCOMM. The command mode is available via the keystroke sequence, <CLEAR><SHIFT><0>. Here is the syntax for invoking LCOMM:

**LCOMM devspec ((parms))**

<b>Devspec</b>	is the RS232 device installed via SET, usually *CL.
<b>XlateS=X'fftt'</b>	Translates character ff to tt before sending.
<b>XlateR=X'fftt'</b>	Translates received character ff to tt.
<b>NULL=OFF</b>	Prevents nulls (00's) from being received.
<b>XON=xxx</b>	Sets the XON character to xxx [CTRL-Q]
<b>XOFF=xxx</b>	Sets the XOFF character to xxx [CTRL-S].
<b>&lt;CLEAR-8&gt;</b>	provides a menu of LCOMM commands while on-line.
<b>&lt;CLEAR&gt;&lt;SHIFT&gt;&lt;0&gt;</b>	provides the execution of a DOS library command.

## **PATCH**

This utility has been enhanced to support the required finding of bytes matching a particular "F-format" patch line prior to installing a "D-format" patch. A new parameter, OPTION, allows you to force or inhibit the required matching (the default is to not require "F-format" patch lines). Coincidentally, a new parameter, REMOVE, has been added to un-install a "D format" patch that was installed with the appropriate required matching of "F-format" patch lines. The syntax of the PATCH utility is now:

**PATCH filespec1(/CMD) (USING) filespec2(/FIX) (parms)**

<b>Yank</b>	Nullifies the effects of a previous X-type patch
<b>Remove</b>	Nullifies the effects of a previous D/F-type patch.
<b>Option=ON</b>	Match F records before installing D patch [OFFJ].
<b>X'hhhh'=</b>	Change file starting at address hhhh to data
<b>Drr,pp=</b>	Change file starting at record rr offset pp to data.

**Frr, ppm**      Finds data at record rr byte offset pp.  
**Lnn**              Designates DOS library command ISAM number.  
**data**              hexadecimal values of the form: xx xx xx... ASCII values of the form:  
                      "abcdefg..."

## **SOLE**

SOLE is a Model I utility file which is used to install a double-density booting driver onto a system disk which has been formatted in dual-density. The Model I FORMAT utility will format a double-density disk with a single-density cylinder 0 when both the DDEN and SYSTEM parameters are specified. This assumes that the double-density disk driver, FDUBL, has been installed. After the operating system has been moved to the dual-density diskette, the booting driver is applied by invoking:

**SOLE (:d)**

where "d" references the drive containing the dual-density disk; if not specified, it will default to :0. Afterwards, move the disk to drive :0 and save the configuration with SYSTEM (SYSGEN).

## **TED**

TED is a new text editor which has been added to LDOS. It is a full-screen ASCII text editor with many editing features. It provides the facility of editing ASCII text files; it is very useful for those LDOS users not already owning a text editor or word processor. Documentation for TED is a separate section of this update documentation.

## **KI/DVR**

The keyboard driver has been improved to allow the key combination, <CLEAR><SHIFT><0>, to generate a code of 160d - the same as <CLEAR><SPACE>. Note that it will not toggle CAPS LOCK; only <SHIFT><0> will toggle CAPS LOCK. Another enhancement to LDOS will restrict KI/DVR from being inadvertently loaded by issuing it as a command: it requires installation via the SET library command. If Model III LDOS is running on a Model 4 computer, installing the KI/DVR will automatically load a driver which supports the CTRL key, the function keys, and CAPS.

## **RS232x/DVR**

The serial driver provided with your system has been enhanced to provide for the selection of a received logical BREAK character rather than arbitrarily using a code of 01d. Note that BREAK normally defaults to OFF; thus, no received character will be

interpreted as a BREAK for the purposes of the system's keyboard scanner. The driver parameters may also be altered after the driver is installed by using the SETCOM library command documented earlier.

## **MINIDOS/FLT**

This keyboard filter has been improved to allow the "repeat last DOS command" to properly function regardless of the filter's position in the keyboard device chain.

## **PR/FLT**

The PR/FLT filter program has been enhanced to allow its operating parameters to be changed after the filter has been installed into memory. See the FORMS library command previously documented. The printer driver portion also supports a ten-second timeout enabled or disabled with the SYSTEM (PRTIME) command.

**HELP** - The HELP facility has been added to provide on-line help information for both DOS commands and disk BASIC statements/functions. The HELP displays use the following notations:

- ( )      Optional arguments or parameters; note that some Model I Systems will display these characters as    and
- |          Alternative selection: ON|OFF means ON or OFF.
- < >      Key combination you press.

Help screen format:

- Brief description
- Command Syntax
- Parameter descriptions

Help information is retrieved by entering commands of the form:

**HELP file command ((P))**

- file**      is the name of an available help data file
- command**    is the name of one help file entry; if omitted, the list of entries is displayed.
- P**            sends the help display information simultaneously to the printer.

## **DATECONV**

This utility program has been added to convert DOS data disks earlier than this 5.3 release to the extended date and time stamp usage. SYSTEM disks must first have the DOS files moved to them via BACKUP, then processed by DATECONV.

## **BASIC Enhancements**

The disk BASIC interpreter has been improved by several additions and the name has been changed to "BASIC/CMD". First, two more edit functions have been added. These are the ability to copy a BASIC program line to another line number and the ability to move a BASIC program line to another line number. No automatic renumbering of imbedded line number targets is done so you will still have to adjust the targets of any GOTOS, GOSUBs, etc. These two edit additions have been added to the BASIC/OV3 overlay file so that overlay must be present on-line for you to be able to use those two commands.

### **Single-key edit commands**

< >	Cursor up one line
< >	Cursor down one line
< >	Cursor to first line
< >	Cursor to last line
<.>	Display current line (period)
<,>	Edit current line (comma)
<A> n1[,inc]	AUTO line n1 with increment
<C> n1,n2	Copy line n1 to line n2
<D> n1[-n2]	Delete line(s) n1 through n2
<E> n1	Edit line n1
<L> n1[-n2]	List line(s) n1 through n2
<M> n1,n2	Move line n1 to be line n2

## **INPUT@**

This provides a much-requested facility of controlling keyboard input to a string variable with prompting and screen field highlighting. Two forms of INPUT@ are supported. One provides the minimal improvement of allowing the input prompt to appear starting at a designated video position similar to a PRINT@ statement. The second form is more powerful. It allows you to specify an input field width, an input field fill character, whether the input should be alphanumeric or just numeric, and whether the input should automatically terminate when the "field width" number of characters have been entered rather than requiring a hard <ENTER>. The two forms of INPUT@ are:

```
INPUT@pos, "message";var(,var)
INPUT@pos, "message", fw, "$|#(*)($)" ;var$
```

- pos** The screen position for the message or field.  
**var** Variables used to store the data retrieved.  
**fw** The maximum input field width <1-240>.  
**\$** Specifies acceptance of characters <32d-127d>.  
**#** Specifies acceptance of characters <0-9, period, minus, plus>.  
**\*** Specifies immediate <ENTER> on maximum input.  
**f** Designates a field fill character [\_\_].  
**var\$** The string variable to receive the input.

When using the second form, the single input variable will always be a string variable and the input, regardless of designated type. will always be a string of characters. Note that since the "immediate <ENTER> on maximum input" character is examined before the programmed fill character, you cannot designate an asterisk as the fill character unless you specify the forced immediate <ENTER>. Thus, "#\*\*" will be interpreted as accepting numeric only, immediate <ENTER> on full field, and use an asterisk as the fill field character. On the other hand, "#\*" will default the field fill character to an underline.

## **CMD"V"**

This BASIC extension can be used to dump a list of active variables and their values and user defined functions while a program is running (or after it was interrupted or ended). The syntax of this command is:

```
CMD"V (*DO|*PR) (-S) (-A) (=x)"
```

- \*DO | \*PR** Designates the output to either device [\*DO]  
**-S** Restricts output to scalars only.  
**-A** Restricts output to array variables only  
**=x** Restricts output to variables starting with "x"

## TED - ASCII Text Editor

The text editor (TED) is a full screen "quick" text editor with typical word-processing type features (four-directional cursor movement; bi-directional scrolling; text directional delete; large text buffer; etc); however, TED was not designed to be a full featured word processor. TED was designed for you to be able to rapidly enter a full-screen text editing environment while accomplishing many of your text file editing tasks.

### Summary of editing commands

The following are the command keys and their functions as supplied by TED. Once you become familiar with the operation of TED, this section may be all you need to refer to from time to time to jog your memory.

Action	Key Entry
Move the cursor one position left	[ ] or [CTRL H]
Move the cursor one Position right	[ ] or [CTRL I]
Move the cursor one position down	[ ] or [CTRL J]
Move the cursor one position up	[ ] or [CTRL K]
Move the cursor to the beginning of the line	[SHIFT LEFT]
Move the cursor to the end of the line	[SHIFT RIGHT]
Move the cursor to the end of the text	[SHIFT DOWN]
Move the cursor to the beginning of the text	[SHIFT UP]
Toggle overstrike/insert modes	[CTRL A]
Specify a BLOCK...	[CTRL B]
Specify DELETE	[CTRL D]
FILE the text buffer to disk	[CTRL F]
GO find the next search string match	[CTRL G]
LOAD a text file into the buffer	[CTRL L]
Command confirmation or advance to next line	[ENTER] or [CTRL M]
Go to the NEXT video page	[CTRL N]
PRINT the entire text buffer	[CTRL P]
QUERY a directory	[CTRL Q]
REPLACB searched string with new string	[CTRL R]
SEARCH for a string	[CTRL S]
Go UP to the previous video page	[CTRL U]
EXIT the text editor	[CLEAR SHIFT =]

### Invoking TED

TED is invoked via the command:

**TED (filename(/TXT))**

TED will display a welcome message on the bottom line of the video screen. This display line will also be used for the display of status, prompting, and error messages. TED displays three different types of messages during its operation. Error messages are indicated by a terminating exclamation point, "!". Queries which need a response are

indicated by a terminating question mark, "?". Informative messages use no special character for their termination. Thus, "Marker!" is an error, "String?" is a query, and "Block" is information.

### **Text entry modes**

TED will accept only displayable ASCII characters in the range 20H through 7FH for text entry. Any other character value will be interpreted as a command entry. If it matches a value in the command table, that command will be invoked; otherwise, the entry will be ignored.

TED operates in two text entry modes: overstrike and insert. The initial mode established when TED is first invoked is the "overstrike" mode. While TED is in "overstrike" mode, it will use an underscore as the cursor character. When you toggle to "insert" mode, the cursor is changed to a full graphics block. You toggle from one mode to the other via the [CTRL A] command.

When TED is in "overstrike" mode, any acceptable text entry typed character is written over the character which appears under the blinking cursor. You can overstrike a newline character (i.e. [ENTER], which is displayed as small graphics block). You can also overstrike either a "begin" block marker or an "end" block marker. You can be in overstrike mode when you come to the end of the text (or starting from an empty text buffer, for that matter) and still be able to enter text in this mode.

When you switch to "insert" mode, anytime you enter an acceptable text entry character, the entire text will be pushed down one position starting from the character under the cursor to make room for the inserted character. The video screen will be constantly updated as text is inserted.

The text entry mode is only changed via the [CTRL A] command. Going into "delete" mode does not change the mode of text entry.

### **Loading a text file**

The [CTRL L] command is used to load a text file into the text buffer area. When you depress [CTRL L], you will be prompted for the name of the file. If the extension is omitted from your entry, "/TXT" will be automatically provided.

The LOAD command will not automatically clear any text remaining in the text buffer prior to the LOAD. The new text is not inserted at the cursor position but rather is appended to the end of the current text. If you wish to load the new file over the old text, simply invoke the command sequence, [SHIFT ] followed by [CTRL D] then [SHIFT Z]. This will delete the entire text buffer.

If the file is too large to fit into the available text buffer, the error message "No room!" will be displayed and no text will be loaded. If any disk read error is encountered while

reading the text file into the text buffer, the message "IO error!" will be displayed. The text which was loaded up to the point of encountering the error will be retained in the text buffer.

### **Entering Text**

Entering text is easy, you just type away. If you already have text in the buffer and wish to enter new text at the end, just move the cursor to the bottom (via the [SHIFT Z] key), then type in your text. If you wish to enter new text at some other point, just position the cursor, toggle to the "insert" mode, then type away. TED will stay in insert mode until explicitly toggled back to overstrike mode.

As you are entering text, any word which is too long to fit at the end of a video line will be split at the 64th column and continued onto the next line. These "long words" are not automatically bounced onto the subsequent line, as is the ease with the typical word processor.

### **Cursor positioning manipulations**

The ARROW keys are the primary tools to move the cursor. They will move the cursor in the direction indicated by the arrow. The shifted arrow keys will be interpreted as cursor movement requests unless TED is in the DELETE or BLOCK modes.

The [SHIFT ] request will move the cursor to the first position of the current line. The [SHIFT ] request will move the cursor to the last position of the current line. You can position the cursor to the first position of the text buffer by a [SHIFT ] request. Finally, the [SHIFT Z] position to the end of text.

The page up, [CTRL U], command will refresh the video screen so that the new first displayed line is thirteen lines previous to the current first displayed line. The page next, [CTRL N], request will refresh the video screen so that the new first line displayed is the last line of the current displayed text. If the video display has fewer than 14 lines of text displayed, the page next request will be ignored.

### **Text deletion**

TED provides five forms of text deletion in addition to the block deletion discussed later. To delete the single character which appears under the cursor, invoke the delete command via [CTRL D]. This action will get rid of the character and all text which succeeded that character will be pulled back one position. The [CTRL D] command also puts you into DELETE mode which is made apparent by the display of the word "Delete" in the status line. The DELETE mode is active for only the next keyboard entry. There are only four subcommands associated with the DELETE mode: delete to beginning of line (bol), delete to end of line (eol), delete to top, and delete to bottom. TED will always prompt before performing one of these deletes.

<b>Deletion desired</b>	<b>Command sequence</b>
delete to bol	[CTRL D] then [SHIFT ]
delete to eol	[CTRL D] then [SHIFT ]
delete to top	[CTRL D] then [SHIFT ]
delete to bottom	[CTRL D] then [SHIFT Z]

After typing [CTRL D], the character now under the cursor is the character which was to the right of the deleted character. Since in the case of delete to bol and delete to top, you are deleting text which is in front of the cursor, you really don't want to delete the character which is under the cursor after the [CTRL D]. Well, you don't have to worry about that because those two subcommands properly backup one position before continuing the deletion.

### **Block operations**

The BLOCK command, [CTRL B], has six subcommands: Begin, End, Copy, Delete, Move, and Print. These subcommands are specified by entering the first letter of the subcommand word ([B], [E], [C], [D], [M], or [P]). The entry may be in either upper or lower case. Note that these subcommands are NOT control key combinations but normal alphabetic single-key entries. When you invoke the BLOCK command, the word Block will be displayed in the status line.

Anytime you need to deal with a block; say to copy it, move it, or delete it, you have to first mark it. The beginning and ending positions of a block are marked by first positioning the cursor over the first character of the block and then entering the two command sequence, [CTRL B] followed by [B]. This is followed up by positioning the cursor over the character immediately following the last character of the block and then entering the two command sequence, [CTRL B] followed by [E]. The beginning position will be indicated on the display by a "begin" marker which is inserted by TED into the text. The marker is displayed as a graphic left bracket. The ending position will be indicated on the display by an "end" marker which is also inserted by TED into the text. The marker is displayed as a graphic right bracket. These markers occupy ordinary text positions; thus they may be deleted or overstriked. Any remaining in the text buffer at the time a FILE command is performed will be written to the disk file just as if they were ordinary text characters.

Although you can mark as many blocks as your heart desires, TED provides no way to differentiate between marked blocks in other than the BLOCK-DELETE function. For copying and moving blocks, the first block marked in the text is the one chosen for copying or moving. On the other hand, a BLOCK-DELETE request requires that the cursor be positioned within the interior of the marked block which is to be deleted.

To COPY the first marked block in the text to some other position, simply mark the beginning and end of the block as discussed above, move the cursor to the position in the

text where you want the marked block copied into, then invoke the block copy command via the sequence, [CTRL B] followed by [C]. Note that the block which will be copied is the first marked block found in the text buffer. A few things could go wrong with your request. If TED can find no properly marked block, it will display the error message "Marker!" and terminate the block mode. Another error which could occur is when the position you wish the block copied into happens to be in the interior of the block itself. You will be informed of this prohibition by a display of the error message "Cursor!".

The successful block copy operation only copies the marked text; the markers are not copied as well. In fact, the marked text remains in its original position relative to the text which surrounds it. The cursor position relative to the text will be unchanged after the block is copied; however, the screen may be refreshed and the physical location of the cursor on the screen may be different.

A block of text may be MOVED from one position to another by a command sequence similar to the block copy. In this case, simply mark the beginning and end of the block as discussed above, move the cursor to the position in the text where you want the marked block moved to, then invoke the block MOVE command via the sequence, [CTRL B] followed by [M]. Again note that the block which will be moved is the first marked block found in the text buffer. This operation is essentially one of copying and automatic deleting without the double check prompt. As in the case of the BLOCK-COPY, the same errors are possible with similar diagnostic messages when things are not as they should be. With the BLOCK-MOVE command, the new cursor position will be the new position of the moved block. The screen may be refreshed and the physical cursor position altered to accommodate this request.

The block operation, deletion, is similar to the above functions, you first must mark the block's beginning and ending positions. You must then position the cursor to the interior of the marked block and invoke the command with the sequence, [CTRL B] followed by [D]. As a safety check, TED will prompt you before deleting the block. It is necessary to depress [ENTER] to affirm your intentions. Any other character entry (including a [Y]) will cause TED to ignore the block delete request.

The same errors as for copy and move can occur: however, the messages may not be for the same reasons. When a block delete is requested, TED will first look for an ending block marker starting from the cursor position. If none is found, the error displayed will be "Marker!". This doesn't mean necessarily that a properly marked block is missing. On the other hand, if an ending marker is found past the cursor position, TED next scans forward for a beginning block marker. A "Marker!" error will also be posted if none is found. If a marker is found but is also past the cursor position, a "Cursor!" error will be posted.

The PRINT block operation will cause the first marked block to be printed.

### **Filing away your text to a disk file**

The [CTRL F] command is used to FILE the contents of the text buffer area into a disk file. When you depress [CTRL F], you will be prompted for the name of the file. If the file specification you wish to use has an extension of "/TXT", you do not have to enter the extension. If the extension is omitted from your entry, "/TXT" will be automatically provided.

The FILE command will save the entire text buffer, excluding the terminating NULL but including any block markers, into the disk file identified by your input. If any disk write error is encountered while saving the text buffer into the disk file, the message "I/O error!" will be displayed. In any case, the text buffer is left undisturbed.

### **Text search**

TED provides the SEARCH command to scan the text buffer for a specified string of characters. You specify the search by invoking the command with [CTRL S]. TED then prompts you for the search string with the query message "String?".

You can enter up to 23 characters to be used for the search string. Terminate your search string with an [ENTER] which is not included as one of the 23 characters. TED will then look for the string starting with the first character following the cursor. The matching is case sensitive which means that characters entered in upper case must be found in upper case and characters entered in lower case must be found in lower case. If the search string cannot be found, the message "Can't!" will be displayed. At this point, the cursor location remains unchanged. If a matching string of text is found in the text buffer, it will be displayed. The display window will be redrawn starting with the line which contains that string. The cursor will be repositioned to the first character of the matching string.

If you press ENTER only in response to the "String?" query, then the search will proceed with the last entered search string, providing one was available. Using this procedure, you can advance the cursor to each occurrence of the search string in question.

Another way to find each occurrence of a search string is with the GO command, [CTRL G]. Each depression of [CTRL G] is identical to the sequence, [CTRL S] followed by [ENTER].

### **Text search and replace**

TED also provides the capability of replacing a text string matching the search string with a different string - the replacement string. When the REPLACE command is invoked via [CTRL R], the query message "String?" will be displayed. The query, same as for SEARCH, is asking you for the replacement string. You can enter up to 23 characters to be used for the replacement. Terminate your string with an [ENTER] which is not included as one of the 23 characters. TED will then look for the currently pending SEARCH string starting with the first character IMMEDIATELY under the cursor. If the

SEARCH string cannot be found, the message "Can't!" will be displayed. At this point, the cursor location remains unchanged. If a matching string of text is found in the text buffer, it will be replaced with the REPLACE string. The display window will be redrawn starting with the line which contained the string which was replaced. The cursor will be repositioned to the first character immediately following the replacement string.

If you wish to replace the next occurrence of text which matches up with the SEARCH string with that same REPLACEMENT string, all you need do is [CTRL R] ENTER.

The GO command, [CTRL G], still functions to find the next occurrence of the SEARCH string. If that occurrence of the string is beyond the text currently displayed on the screen and you wish to confirm its replacement, simply GO to the next occurrence then REPLACE, as necessary.

### **Printing text**

TED provides the [CTRL P] command to print the entire text buffer. If you want to print just a block of text, use the BLOCK PRINT command.

### **Obtaining a Directory**

TED provides the [CTRL Q] command to obtain a directory of files. After the "Drive" prompt, enter the desired drive specification as either ":d" or "d". You may also restrict the display to files matching a particular file extension by entering a 4-character string preceding the drive, as in "/TXT:1"; a dollar sign in any extension character position designates a match on any character. Depressing [ENTER] after a directory query will restore the text screen image.

### **Exiting from TED**

To exit TED and return to DOS Ready, use the [CLEAR SHIFT =] command. If the text buffer is empty, TED will immediately terminate. If, however, if there is any text in the buffer, you are provided an opportunity to retract your request. TED will display the prompt message,

**Press ENTER to exit**

### **Text recovery**

If you exit the TED application inadvertently without saving the edited text, TED permits you to re-enter with the asterisk parameter which provides a chance to recover your text. Instead of automatically clearing the text buffer as TED does, TED \* will display whatever is in the text buffer memory area. Thus, if you have not altered any of the information in that memory area, you can always go back and recapture it.

Since all of the text pointers normally established by TED will not be initialized when invoking TED via the \* parameter, it will be necessary to scroll through the text until reaching its last character prior to doing any other operation. This may also be performed using NEXT PAGE.

## **Software License Agreement**

The LDOS 5.3.1 software update and documentation is copyrighted with all rights reserved by MISOSYS, Inc. The buyer is granted a license to use this software on one computer system at a time, and may make copies for personal use and archival purposes only. Copying this software for sale or other distribution is expressly prohibited.

## **Limited Warranty**

MISOSYS warrants the diskette media on which the LDOS 5.3.1 is recorded to be free from defects under normal use for a period of 30 days after the date of original purchase. If during this 30 day period a defect in the diskette media should occur, you may return the diskette to MISOSYS for replacement without charge.

Except as provided for above, MISOSYS makes no warranties, either expressed or implied, with respect to this software or its fitness for any purpose. MISOSYS software is provided "as-is". Should the software prove defective, the buyer assumes the entire cost of all necessary servicing, repair, or correction and any incidental or consequential damages. MISOSYS assumes no liability for direct, indirect, incidental or consequential damages resulting from any use of or any defects in the software.

**MISOSYS, Inc.  
P.O. Box 239  
Sterling, VA 22170-0239**

Copyright 1991 MISOSYS, Inc. All rights reserved

- . FDUBL/FIX - Patch to Model I 5.3.1 FDUBL/CMD
- . Patch corrects installation for FDUBL (Percom}
- . Apply via, PATCH FDUBL.DRIVER FDUBL1

X'5265'=CD 74 55  
X'5574'=3E 2C 32 0D 54 21 14 54 C9

- . Eop

- . SOLE1/FIX - Patch to LDOS I 5.3.1 SOLE/CMD
- . Corrects installation when using Percom-type doubler
- . Apply via, PATCH SOLE.UTILITY SOLE1

D00,8D=55  
F00,8D=52  
D00,8F=08 56  
F00,8F=04 52

- . Eop

- . SOLE2/FIX - Patch to Model I 5.3.1 SOLE/CMD
- . Fix alters SOLE to force cylinder 1 fully allocated
- . Apply via, PATCH SOLE.UTILITY SOLE2

D01,15=CD 2C 54 1E 02 CC  
F01,15=FD 56 09 1E 02 CD  
D01,41=45  
F01,41=2C  
D02,6B=FD 56 09 1E 00 CD 45 4B 36 FF CC 18 53 C9  
F02,6B=50 68 61 73 65 20 32 20 69 73 20 6E 6F 77  
D02,8F=20  
F02,8F=0A

- . Eop

- . SYS011/FIX - Patch to Model I 5.3.1 SYS0/SYS
- . Apply via, PATCH SYS0/SYS.SYSTEM SYS011

D05,EF=CB 4B  
F05,EF=FF 46  
D0A,39=FD E9  
F0A,89=00 00

- . Eop