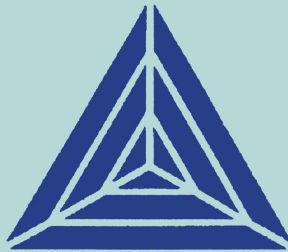


LS-DOS 6.3.1™
Upgrade Documentation

CAT. NO. 700-2297
LS-DOS 6.3.1 Update
to TRSDOS 6.2.X
For Model 4/4D/4P

*Copyright © 1990 MISOSYS, Inc.,
All rights reserved*



MISOSYS, Inc.

LS-DOS 6.3.1 Update Documentation

Table of Contents

Software License Agreement	1
LIMITED WARRANTY	1
LS-DOS™ 6.3.1 Update to TRSDOS® 6.2.X	2
LS-DOS 6.3 installation instructions	2
Updating DATA disks	3
Updating SYSTEM disks	3
Updating HARD disks	3
Update Information	5
Boot up changes	5
Library command changes	6
Utility Program changes	8
TED - ASCII Text Editor	10
Summary of editing commands	10
Invoking TED	11
Text entry modes	11
Loading a text file	12
Entering text	12
Cursor positioning manipulations	12
Text deletion	13
Block operations	14
Filing away your text to a disk file	16
Text search	16
Text search and replace	17
Printing text	18
Obtaining a directory	18
Exiting from TED	18
Text recovery	18

LS-DOS 6.3.1 Update Documentation

Changes and Additions to BASIC	19
Renum	19
Immediate keys	19
Single letter commands	19
Copy command	20
Find command	20
Move command	20
Search command	21
High speed Load and Save	21
Interface to SVCs	21
BASIC Cross Reference Utility	24
Technical Information Changes	26
Changes to directory structure	26
Changes to SVCs	26
Password changes	27
Addendum 1A	28
About MISOSYS Customer Support Services	28
Most asked questions	28

LS-DOS 6.3.1 Update Documentation

Software License Agreement

The LS-DOS 6.3.1 software update provided by MISOSYS, Inc., is intended only for legitimate owners of a TRSDOS 6.x operating system. By accepting this software license, the buyer expressly warrants and represents that he is a legitimate owner of a TRSDOS 6.x operating system.

The LS-DOS 6.3.1 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, 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 LS-DOS 6.3.1 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
(703) 450-4181

LS-DOS 6.3.1 Update Documentation

LS-DOS™ 6.3.1 Update to TRSDOS® 6.2.X

The *LS-DOS 6.3* release is an upgrade for the *TRSDOS 6.2* operating system. 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 through the year 2011. Files will now carry a modification time as well as a date. The *DATECONV/CMD* program is provided to translate version 6.2 or earlier disks to the 6.3 style dating. The *user* password has been eliminated from the system. The *owner* password still remains. The library command *ID* displays your customer service status. Several enhancements have been made to *BASIC*. The new *DISKCOPY/CMD* program will duplicate 5" double density floppy disks.

Because the *LS-DOS 6.3* update is a series of enhancements to *TRSDOS 6.2*, the primary documentation remains the 6.2 manual and Technical Reference manual. If you have a version of *TRSDOS* earlier than 6.2, you can obtain the manuals from Radio Shack under the catalog numbers 26-0316 (*TRSDOS Version 6* [6.2 DOS manual and disk]), 26-2110 (*Model 4/4D [6.2] Technical Reference manual*), or 26-2117 (6.2 DOS manual only). This documentation should be treated as an addendum to the *TRSDOS 6.2* information.

LS-DOS 6.3 installation instructions

Before performing the upgrade, it is recommended that you make several backup copies of the 6.3 master disk. The simplest way is to boot up the 6.3 diskette, insert a blank diskette to receive the copy in drive 1, and type the command:

```
DISKCOPY :0 :1 [ENTER]
```

When the copy finishes, you can insert another destination disk and make another copy.

IMPORTANT NOTE

It is important that you *DO NOT* switch between version 6.3 and any earlier version system disk in drive 0 without rebooting the computer.

LS-DOS 6.3.1 Update Documentation

To update your existing TRSDOS 6.2 disks, boot up with a fresh copy of your new 6.3 system disk. The 6.3 system disk should remain in drive 0 throughout the update procedure. There are two separate methods for updating 6.2 floppy disks, depending on whether they are system disks or data disks.

Updating DATA disks

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

```
DATECONV :1 [ENTER]
```

This completes the conversion of a data disk.

Updating SYSTEM disks

Updating system disks may take two or three steps. Place the 6.2 disk in drive 1, and issue the command:

```
BACKUP :0 :1 (I,S,OLD) [ENTER]
```

If you wish to use the new BASIC enhancements, type the command:

```
BACKUP BASIC:0 :1 (I) [ENTER]
```

Once the backups are completed, give the command:

```
DATECONV :1 [ENTER]
```

This completes the updating of a system disk. At this time, you can add any of the new utility files you wish.

IMPORTANT NOTE: If you have sysgened a configuration file, you will have to re-create it the next time you boot the disk.

Updating HARD disks

For most hard disk installations, the following instructions can be used. Boot up the hard disk as you normally would. Typically your system will have the hard disk as drive 0, and the first floppy drive as drive 4. If your configuration is different, change the drive num-

LS-DOS 6.3.1 Update Documentation

bers accordingly in the following instructions. Place the 6.3 disk in drive 4, and type the following commands:

```
BACKUP :4 :0 (I,S) [ENTER]  
DATECONV :0 [ENTER]
```

Remove the 6.3 disk from drive 4, and insert your normal boot disk. Type the following commands:

```
BACKUP :0 :4 (I,S,OLD) [ENTER]  
DATECONV :4 [ENTER]  
MEMORY (A=X'3B',B=0) [ENTER]  
MEMORY (A=X'85',B=99) [ENTER]  
MEMORY (A="Y",B=255) [ENTER]  
SYSGEN (D=4) [ENTER]
```

Now, reboot the system and use *DATECONV* on your other 6.2 hard drive partitions.

IMPORTANT NOTE: If your hard drive configuration has system files sysgened in memory, you will have to re-create the configuration file after putting the 6.3 operating system on your hard drive and your booting floppy disk.

Reformatting hard drives

If you ever have to re-format a hard drive using the TRSFORM6/CMD program that comes with TRSDOS 6.2, you should use the DATECONV utility after the formatting is complete. This will make sure the hard drive is marked as being version 6.3. Alternatively, you can apply the following patch to TRSFORM6/CMD that will permanently alter it to format version 6.3 partitions.

```
. TF6X/FIX - Patch for 6.3 compatibility  
. Apply via, Patch TRSFORM6/CMD.UTILITY TF6X  
D04,C0=88;F04,C0=80  
D05,01=63;F05,01=62  
D09,2B=33;F09,2B=32  
D0D,AA=63;F0D,AA=62
```

LS-DOS 6.3.1 Update Documentation

Update Information

This section describes the changes to existing commands, and explains new commands and utility programs. You can continue to use your 6.2 disks with version 6.3 as long as you update them as explained in the installation instructions. The following is a brief guide to version compatibility.

USING 6.2 OR EARLIER DISKS WITH 6.3 - Disks from earlier versions of TRSDOS should be converted with the new *DATE-CONV/CMD* program before being used with 6.3. Unconverted disks will work, but the time and date information will not be correct. If a disk written to with 6.3 is used with an earlier version DOS, it may appear to have an unknown user 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 from version 6.3 to earlier versions.

USING 6.3 DISKS WITH EARLIER VERSIONS - This normally should not be done if the disk is to be used again with 6.3, as the year and time information will not be correct if the disk is written to.

USING LDOS DISKS WITH 6.3 - Files can be moved from LDOS 5.1 disks to 6.3 disks with either the *COPY* library command or with a backup by class. When moving files from 6.3 to LDOS 5.1 disks, only the *COPY* command with the parameter (*CLONE=NO*) should be used. The *DATECONV* program will not work on LDOS 5.1 disks.

Boot up changes

Since version 6.3 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 hours, minute, and seconds, or just the hours and minutes. Pressing only [ENTER] for this prompt will start the time at 00:00:00. If you wish to suppress the time prompt, give the command **SYSTEM (TIME=OFF)** once the DOS Ready prompt appears. If you wish to reactivate the prompt after you have turned it off, use the command **SYSTEM (TIME=ON)**. These commands do not have to be sysgened.

LS-DOS 6.3.1 Update Documentation

Library command changes

ATTRIB - The user password field has been removed from all files, so the **USER** parameter is no longer valid. The system now assumes a blank user 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** (or **NONE**). Although the documentation for the **ATTRIB** library command lists **EXEC** as the highest protection level, the use of **NO** is valid and provides the highest level of protection.

COPY - When copying from a 6.2 or earlier version disk to a 6.3 disk, the old user password (if any) will be removed and the 6.3 style date/time information field will be established, the time being set to 12:00am.

CREATE - Parameters **FILL** and **SHRINK** have been added to this command. Specifying (**Fill=dd**) populates the **CREATED** file with the specified value **dd**. Specifying (**SHRINK**) constructs the designated file for future deallocation of unused space.

DATE - The acceptable range of dates is from January 1, 1980 to December 31, 2011. Years from 2000 to 2011 are entered as 00-11. The binary storage for the year is excess 1900; thus years 1980-2011 are stored as 80-111.

DIR - The directory display now shows a **mod time** immediately following the **mod date** for all 6.3 diskettes. The time will be in 12 hour format, with **AM** or **PM** indicated. Disks from earlier versions of **TRSDOS** will display a blank time field.

DO - When generating the **SYSTEM/JCL** file, **Do** will automatically search for the first available non-write-protected drive to create the output.

ID - **LS-DOS 6.3.1** extended customer support is via a separate support contract; as such, the **ID** command displays "No service contract".

LS-DOS 6.3.1 Update Documentation

LIST - The display output defaults to paged display mode. 23 lines of ASCII information (16 lines of hexadecimal) will be displayed, then the listing will pause. <ENTER> will resume the display. Specify **LIST ... (NS)** to force the listing to display in non-stop mode. NS may be abbreviated to "N"; note that "N" is no longer an acceptable abbreviation for the "NUM" parameter. If you select the (P) output mode (i.e. output to a printer), (N) is automatically specified.

MEMORY - The display output now includes the status of switchable memory banks known to the DOS. It also displays a map of modules resident in I/O driver system memory and high memory. Furthermore, MEMORY now accepts a (Print) parameter to direct its output to the line printer.

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-6.3 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.

SPOOL - The BANK parameter entry may now be in the range 0-30 instead of 0-7. This may be useful for machines with extended memory installed.

SYSTEM - Three parameters have been added to the SYSTEM command. **SWAP** enables the swapping of any two logical drives by switching their drive control table assignments. Specifying **SYSTEM (DRIVE=d1, SWAP=d2)** switches drive d1 for d2. Either may be the system drive, and a Job Control Language file may be active on either of the swapped drives. Note that **SYSTEM (SYSTEM=d)** is equivalent to **SYSTEM (DRIVE=0, SWAP=d)**.

LS-DOS 6.3.1 Update Documentation

Printer time-out when line printer output is requested and the printer is unavailable can be enabled or disabled with **PRTIME** by specifying **SYSTEM (PRTIME=ON|OFF)**, and is configurable with the **SYSGEN** command. Specifying **SYSTEM (PRTIME=ON)** establishes an approximate 10 second time-out and generation of the *Device not available* error code on printer unavailable. Specifying **SYSTEM (PRTIME=OFF)** disables any time-out; if the printer is unavailable, the DOS will hang!

The display of the time field in the **DIR** command's output can be altered to display the time in 12-hr clocktime or 24-hr clock time. Specifying **SYSTEM (AMPM=ON)** will cause a 12-hr clock display; specifying **SYSTEM (AMPM=OFF)** will cause a 24-hr clock display. This option is configurable and pertains only to the **DIR** display.

SYSGEN - Job Control Language, if active, will be temporarily suspended during the configuration generation; **SYSGEN** will no longer abort if within a JCL file. Also, **SYSGEN** will now abort with an appropriate error if the drive specified to receive the configuration file is write protected.

TIME - The time may now be entered with the seconds being optional. The full syntax for the **TIME** command is:

TIME [HH:MM[:SS]] [(CLOCK=ON|OFF)],

Utility Program changes

BACKUP - When doing a backup by class from a 6.2 or earlier version disk to a 6.3 disk, the user password (if any) will be removed and the 6.3 style date/time information will be established. It will NOT be permissible to backup **SYSTEM** files from a 6.2 or earlier disk to a 6.3 disk during a backup by class.

DATECONV/CMD - Converts pre-6.3 version (non-system) disks to use the new time/date format. The syntax is:

DATECONV :d

There are no parameters for this program. If conversion of a TRSDOS 6.2 or earlier **SYSTEM** disk is attempted, an appropriate error message will be displayed. To convert this type of disk, you must first

LS-DOS 6.3.1 Update Documentation

use backup to move 6.3 system files onto it before using DATECONV. For example,

BACKUP :s :d (S,I,OLD)

Data disks from versions earlier than 6.3 will be converted in all cases and marked as a 6.3 type disk.

Conversion of LDOS 5.1 formatted disks will not be allowed. Although these disks can be read and written to, the time and year will not be updated on modified files. Files should be copied or backed up onto 6.3 formatted disks.

DATECONV does not normally need to be used on a 6.3 disk, since the COPY and BACKUP commands automatically adjust the time/date storage when moving from LDOS or TRSDOS 6.x to 6.3. For this reason you should NEVER move files to a 6.3 disk when booted up under any earlier version. If you do use DATECONV on a 6.3 disk, only those files that have an old style blank user password will be changed. This is to protect the time stamp on proper 6.3 files.

DISKCOPY/CMD - Does a single-pass format and copy of a 5" double density floppy disk. It will duplicate both single and double sided disks. Attempting to copy other types of disks will abort the program with an appropriate error message. The syntax is:

DISKCOPY :s :d

The :s and :d are the *source* and *destination* drive numbers, respectively, and cannot be the same drive. After starting DISKCOPY, you will be prompted to insert the source and destination disks.

After the copy is complete, you will be prompted to do another copy. If you wish to copy another disk, press the [Y] key. You can switch both the source and destination disks at this point if you desire. Pressing [N] will exit back to DOS Ready.

If an error is encountered during the copy, the program will display the error and return to DOS Ready.

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 J]
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 display	[CTRL Q]
REPLACE 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 =]

LS-DOS 6.3.1 Update Documentation

Invoking TED

TED is invoked via the command:

TED

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 doesn't match a valid command, the entry will be ignored.

TED operates in two text entry modes: overstrike (initial mode) and insert. 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 only 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.

LS-DOS 6.3.1 Update Documentation

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 ↓]. 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 `I/O 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 ↓] 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 80th column and continued onto the next line. These "long words" are not automatically bounced onto the subsequent line, as is the case 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.

LS-DOS 6.3.1 Update Documentation

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 ↓] 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 twenty one 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 22 lines of text displayed, the page next request will be ignored.

Text deletion

TED provides five forms of text deletion in addition to 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, 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.

Deletion desired	Command sequence
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 ↓]

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 overstricken. 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 to some other text 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. If TED can find no properly marked block, it will display the error message

LS-DOS 6.3.1 Update Documentation

Marker! and terminate the block mode. If the position you wish the block copied into is in the interior of the block itself, TED informs you of this error by a display of the error message Cursor!.

The successful block copy operation copies only 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 precaution, TED will prompt you before deleting the block. You must depress [ENTER] to affirm your intent. TED will ignore the block delete request on any other character entry (including a [Y]).

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.

LS-DOS 6.3.1 Update Documentation

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]` (the `[ENTER]` character code is not included as one of the 23 characters). TED will then look for the string starting with the first character immediately 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, on the other hand, 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.

LS-DOS 6.3.1 Update Documentation

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 up with 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. Although the message is the same as for SEARCH, this query is asking you for the replacement character string. You can enter up to 23 characters to be used for the replacement string. Terminate your string with an [ENTER] (the [ENTER] character code 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, on the other hand, 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. Knowing this, if the next occurrence of the search 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.

LS-DOS 6.3.1 Update Documentation

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 the directory query will restore the text screen image.

Exiting from TED

It is easy to exit TED and return to DOS Ready. The [CLEAR SHIFT =] command tells TED you wish to exit. If the text buffer is empty, TED will immediately terminate. 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

There may be times when you exit the TED application inadvertently without saving the edited text to a disk file. 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.

Caution: 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.

LS-DOS 6.3.1 Update Documentation

Changes and Additions to BASIC

The changes and additions to BASIC involve a new parameter for the renumber command, the addition of several "immediate action" keys, single letter abbreviations, high speed load and save, and an interface to DOS SVCs.

The RENUM statement now allows for a fourth line number to be used. If used, it will be the last line of the program to be renumbered. For example:

```
RENUM 1000,500,10,600
```

This will renumber the lines between 500 and 600 only, making new line numbers starting at 1000 with an increment of 10. All other restrictions of the current renumber command still apply. For moving blocks of lines in a program, see the C and M commands described below.

There are 6 "immediate" keys that will function when pressed as the first character of a line. The up and down arrows list the previous and next line of a program, respectively. The left arrow lists the first program line, and the right arrow lists the last program line. The period lists the current line, and the comma enters the edit mode for the current line.

Several single letter commands have been added to BASIC. Four of these are abbreviations for existing commands.

A - AUTO
D - DELETE
E - EDIT
L - LIST

These commands work exactly as their full length counterparts, except that no space is necessary between the letter and the line numbers. For example, "L100-300" is the same as "LIST 100-300".

The following four commands are contained in the BASIC/OV2 file. This file must be present when using one of these commands, or a File not found error will occur. Like other BASIC editing com-

LS-DOS 6.3.1 Update Documentation

mands, the use of these will clear all variable values and close any open files.

C - Copy a single line

F - Find all references to a line, variable or keyword

M - Move a block of lines

S - Search and display a reference to a line, variable, or keyword.

The Copy command syntax is: C Num1,Num2

Num1 is an existing line number to be copied. *Num2* is the line number to create, and must NOT already exist. No renumbering will be done after the copy. If the line numbers are incorrect, an Illegal function call error will occur.

The Find command syntax is: F Object

Object is either a line number, variable, or keyword. The space after the F is mandatory when finding keywords. The resulting display will be all line numbers containing the referenced object. When finding variables, only the first 10 characters of the variable name will be significant. Also, type declarations !,%,#,\$,(must be used. For example, the command "F A" would not find A\$ or A().

The Move command syntax is: M Num1,Num2,Num3

Num1 and *Num2* are existing line numbers and define the block of lines to be moved. *Num2* must be greater or equal to *Num1*. *Num3* is an existing line number and is the line to insert the moved block after. The moved block of lines will be renumbered by one, and all references to these lines (if any) will be corrected. If there is not enough room in memory to move the lines, an Out of memory error will occur. If this happens, do multiple moves of smaller pieces. If the line numbers are non-existent or if there is not enough room between *Num3* and its following line to fit the block, an Illegal function call error will occur. For example, if you had program lines 100 and 110, a block of lines moved after line 100 could be no more than 9 lines long.

LS-DOS 6.3.1 Update Documentation

The Search command syntax is: S *Object* or S

Object is a line number, variable or keyword. The first line containing the object will be displayed. The space after the S is mandatory when searching for keywords. The S with no object following will search for the next occurrence of the previous object. Like the Find command, variables are limited to 10 significant characters, and any explicit type declarations must be used.

High speed Load and Save

When using the normal SAVE or LOAD program commands, execution should be two to three times faster than before. Programs saved with the ASCII parameter will not enjoy this speed increase, either when saving or when loading.

New user interface to SVCs

A user interface to the DOS SVC functions is now provided via the USR statement in BASIC. To use this interface, establish an INTEGER array with the first 6 elements containing the following information that may be needed by the SVC:

- Element 0: SVC number (Always needed!)
- Element 1: Value for register pair HL
- Element 2: Value for register pair DE
- Element 3: Value for register pair BC
- Element 4: Value for register pair IX
- Element 5: Value for register pair IX

The interface is accomplished by using a normally out of range USR argument, USR11. To execute the SVC, use the syntax USR11(VARPTR(ARRAY(0))). For clarity, "ARRAY" has been used for the name of the array in the example, but the actual name of the array used to pass the parameters to USR11 must be only one or two characters long. If the name of the array is longer than two characters, USR11 can't check the array type bit because of some technical details of the way VARPTR works. No DEF USR statement is required. After the SVC executes, the register pairs will be unloaded back into the array. The AF register pair will be placed in array position 0. If the array is not an integer type, or if the SVC

LS-DOS 6.3.1 Update Documentation

number is either zero or greater than 127, an Illegal function call error will occur.

The return condition of the SVC can be tested by checking the bits in ARRAY(0). Doing an "AND 64" will produce a non-zero value if the Z flag is set. Doing an "AND 1" will produce a non-zero value if the Carry Flag is set. For further explanation of DOS SVC usage and returned values, refer to the TRSDOS 6.2 *Technical Reference Manual*. Following is a short example of using the SVC interface.

```
100 DEFINT J,K:DIM J(5)' Important - integer array only for SVC
interface, must be a 1 or 2 character name
200 CLS:PRINT TAB(25)"SVC Demonstration Menu"
210 PRINT:PRINT TAB(25)"1) Set Scroll Protect"
220 PRINT TAB(25)"2) Toggle Caps Lock"
230 PRINT TAB(25)"3) Show DOS Version"
240 PRINT TAB(25)"4) Check Drive Ready"
250 PRINT:PRINT TAB(25);:INPUT "Make a selection ";A$
260 IF A$ [ "1" OR A$ ] "4" THEN 200
270 A=VAL(A$):ON A GOSUB 1000,2000,3000,4000
280 GOTO 200
1000 CLS:PRINT"Set number of scroll protect lines 0-7"
1010 PRINT:PRINT"0 will cancel scroll protect ";:INPUT A
1020 IF A [ 0 OR A ] 7 THEN 1000
1030 J(3)=&H700 + A 'Register B=7, Register C=line count
1040 J(0)=15:X=USR11(VARPTR(J(0))) 'Execute @VDCTL SVC
1050 IF A=0 THEN RETURN
1060 FOR K=1 TO 100:PRINT K;"Testing scroll protect to
100":NEXT:RETURN
2000 CLS:J(0)= 101:X=USR11(VARPTR(J(0))) ' @FLAGS SVC to get
keyboard flag
2010 K=PEEK(J(4)+10)' Get KFLAG value
2020 POKE (J(4)+10), K XOR 32 'Toggle Caps lock bit
2030 INPUT"Type some characters to check Caps lock, press ENTER
to end ";A$
2040 RETURN
3000 CLS:J(0)=101:X=USR11(VARPTR(J(0)))' Get flag table base,
@FLAG SVC
3010 K=PEEK(J(4)+27)' Peek out version number, base+27
3020 K=K-(&H60)' Earliest version was 6.0, K will be 0 to 3
3030 PRINT "This is DOS version 6 .";K
3040 PRINT:INPUT"Press ENTER to return"; A$;RETURN
4000 CLS:INPUT"Enter drive # to check 0 to 7 ";K
4010 IF K [ 0 OR K ] 7 THEN RETURN
4020 J(3)=K' Set drive number for SVC
4030 J(0)=33:X=USR11(VARPTR(J(0)))' @CKDRV SVC
4040 PRINT:PRINT"Drive";K;"has ";
4050 IF (J(0) AND 64)=0 THEN PRINT"no disk mounted.":GOTO 4080
4060 PRINT"a disk mounted."
4070 IF (J(0) AND 1)=1 THEN PRINT"The drive is write pro-
tected."
4080 PRINT:INPUT"Press ENTER to return ";A$;RETURN
```

LS-DOS 6.3.1 Update Documentation

This example program shows several methods of accessing information not normally available from BASIC. To make effective use of the SVC interface, you will need to have the *Model 4 Technical Reference Manual*.

The @CKDRV routine (lines 4000-4080) shows how the return status of an SVC can be checked. The Z flag can be tested by doing an "AND 64" against the array(0) position. If the result is zero, the Z flag was NOT set. The CF (carry flag) can be tested in the same manner by doing an "AND 1". Again, a zero result means the flag was not set.

The Caps lock and DOS Version subroutines show how information can be referenced in the system flag area with the @FLAGS SVC. There are several other flags and system storage areas that can also be referenced off of the flag table base.

Certain SVCs, such as @CMNDI, @CMNDR, or setting the high memory pointer must NOT be done. Generally speaking, anything that can legally be done from BASIC with @CMNDR can be done with the SYSTEM "command" statement, and should not be attempted through the SVC interface.

LS-DOS 6.3.1 Update Documentation

New BASIC Cross Reference Utility

BREF/CMD is a cross reference utility for BASIC programs. It is executed at the DOS Ready prompt, not from inside of BASIC. The BASIC program must NOT have been saved in ASCII. The syntax is:

BREF *filespec* [(parm,parm,...)]

Parameters are:

VAR=ON OFF	- Default is on
LINE=ON OFF	- Default is off
P=ON OFF	- Default is off
W=n	- width for printer, Default is 80

The *filespec* is the name of the BASIC program. The VAR parameter allows variables to be cross referenced. The LINE parameter includes a cross reference of line numbers. The P parameter allows the listing to go to a printer rather than the video. The W parameter can be used to specify the number of columns for the printer (generally either 80 or 132, although any values between 32 and 255 will be accepted).

Variable names will be displayed up to 14 characters. If the variable is longer than this, the remaining characters will be truncated for display. BASIC stores certain keywords in ASCII, and this makes them indistinguishable from a variable. For example the "AS" used in field statements. This will cause these words to be displayed in a variable cross reference list.

If there are more references than can be displayed on a single line, they will wrap to the next line. This line will have an asterisk in column one to denote the overflow.

Examples:

BREF PROG/BAS	- Sends only variable references to the video.
BREF PROG/BAS (LINE,VAR=OFF)	- Sends only line number references to the video.

LS-DOS 6.3.1 Update Documentation

BREF PROG/BAS (LINE,P,W=132) - Sends variables and line numbers to a 132 column printer.

Possible errors:

There are three error messages that may be generated by BREF.

Not a BASIC program

The general cause of this error is trying to use a program saved in ASCII, a program saved in the protected mode, or a non-BASIC program file.

Out of memory - can't cross reference

This will occur if the program is too large to fit into your available memory. The cure is to change your configuration to free up some memory.

Line nnnn, Error in original program

This message may occur if there is a syntax error in the original BASIC program. The line number should correspond to that line in the program.

LS-DOS 6.3.1 Update Documentation

Technical Information Changes

Changes to directory structure

GAT - The disk type byte (GAT+CDH) now assigns bit 3 as the date type bit. If this bit is set, the disk is assumed to use the new time/date fields.

DIR+2 - This byte continues to hold the day in bits 3-7 as documented for earlier versions. However, bits 0-2 should no longer be considered the primary year field. For compatibility with earlier versions, these bits should remain untouched by the user as certain system routines will modify this area.

DIR+18,19 - These two bytes are no longer assigned as the user password. Instead, they contain the new style time/year information as follows:

DIR+18 - bits 3-7 contain the hour
bits 0-2 contain the most significant bits of the minute

DIR+19 - bits 5-7 contain the least significant bits of the minute
bits 0-4 contain the year offset from 1980

Changes to SVCs

@CKDRV - This SVC now adjusts the appropriate bit in the Y flag to reflect GAT+CDH, bit 3 (the old or new style date bit) for the drive being accessed.

@FLAGS - The 'Y' flag byte has been assigned to mark the dating style of a disk. Bits 0 to 7 correspond to drives 0 to 7. If the bit is set, the drive is assumed to use the new style time/date information. The setting and resetting of this bit is handled by the @CKDRV SVC. The operating system version (flag table base+27) has been set to 63H. The release number (flag table base-47) has been set to 01H.

@FLAGS - The 'T' flag byte now uses bit-4 set to specify 24-hr clock display.

LS-DOS 6.3.1 Update Documentation

@HEXD - NEW SVC, number 95. (Similar to **@HEXDEC** SVC, number 97).

Converts a binary number in HL to decimal ASCII.

Entry conditions:

A=95 (X'5F')

B= expected number of output digits

HL= number to convert

DE= pointer to buffer to hold converted number

Exit conditions: Success always

DE=pointer to end of buffer+1

AF, BC, HL are affected by this SVC

This SVC is the same as **@HEXDEC** except that a 5 character buffer is not required. Instead, the B register should contain the maximum number of output characters expected, and DE should point to a buffer of at least that size. If the conversion exceeds this number of characters, the overflow will be placed in memory below the buffer pointed to by DE.

@INIT - If a file is created with a password, the protection level will now be set to NO. This means that no future access will be given without the use of the password.

@VDPRT - NEW SVC, number 107. This SVC performs a screen print. If the printer is not ready, the SVC will abort in approximately 20 seconds, depending on the configuration of PRTIME (see SYSTEM).

Entry conditions: A=107 (X'6B')

Exit conditions: Success always

BC, DE, HL are changed. No status is returned.

Password changes

The owner password for all system files [BOOT/SYS, DIR/SYS, and SYSn/SYS] has been changed from "LSIDOS" to "SYSTEM6".

About MISOSYS Customer Support Services

In a product as complicated and as large as an operating system there will always be problems found that are in need of correction. In an ongoing effort to support LS-DOS 6.3.x, MISOSYS may from time to time issue a revision statement as patches to the existing current release. You may, in most cases, make the revisions yourself or you may obtain a **replacement master disk**. MISOSYS provides a replacement master disk for \$15.00 plus shipping and handling.

ALL revisions should be available on the LDOS forum on CompuServe (PCS-49) and will be published in *THE MISOSYS QUARTERLY*, which you can subscribe to by contacting MISOSYS. Should you run into a problem with 6.3 that you consider a "bug", please forward your report by mail. The information you supply should be sufficiently detailed for us to reproduce your problem. When making a written report, be as specific as possible and include as much information as possible: screen prints, copies of disks, printouts, exact keyboard sequences, error messages, etc., are all helpful. Remember, vague questions will get vague responses.

As have received many of the same questions over the past few years concerning LS-DOS 6.3, here are the answers to the most common user problems.

Question: *I just received my 6.3 disk and I get a parity error (or other read error) when I try to make a copy of the master, or my master disk won't even BOOT ?*

Answer: This type of problem is most likely to occur if your Drive :0 is slightly out of alignment. Try this:

- > BOOT UP with a 6.2 disk as you normally would.
- > Place your MASTER 6.3 in drive :1
- > Type the command: `DISKCOPY.UTILITY :1 :0 [ENTER]`
- > At the insert disks prompt, place a blank disk in Drive :0
- > Press [ENTER]

If this procedure goes through without an ERROR message then you have succeeded in duplicating your MASTER disk to your drive :0. The disk in drive :0 will now be a fully functional and usable system in your boot drive. BUT, your drive :0 should be realigned.

LS-DOS 6.3.1 Update Documentation

If this procedure does not complete without ERROR then your MASTER disk probably is defective or was damaged in shipping. DON'T PANIC - just let us know and we will send you another master diskette.

Question: *How do I enable more FLOPPY disk drives with my LS-DOS 6.3?*

Answer: The easiest way is to refer to the SYSTEM command in your manual for information on ENABLING and DISABLING drives. The SYSGEN command will then save these desired states and initialize them when you boot up. The commands you would use to enable 2 additional drives as drives #2 and #3 are:

```
SYSTEM (DRIVE=2,ENABLE) [ENTER]
SYSTEM (DRIVE=3,ENABLE) [ENTER]
SYSGEN [ENTER]
```

This is all there is to it... simple and straight from the 6.2.0 manual.

Another way would be to "PATCH" the operating system so that the drives are enabled from the start. Although this is NOT the recommended method, it is safe and can be used without

problems. To make these patches you would execute the following commands:

```
PATCH BOOT/SYS.SYSTEM6 (D02,84=C3:F02,84=C9)
[ENTER]
PATCH BOOT/SYS.SYSTEM6 (D02,8E=C3:F02,8E=C9)
[ENTER]
```

Note: The first patch line ENABLES drive #2 the second patch line ENABLES drive #3. You must re-BOOT after installing for these patches to take effect.

One other method of turning on extra drives or other "environment" arrangements is to put them in a JCL file called INIT/JCL and then set an "AUTO DO INIT" on your boot disk. This method is most frequently used at MISOSYS.

LS-DOS 6.3.1 Update Documentation

Question: *I want to get a copy of the current 6.2 user and/or Technical Reference Manual; What do I do?*

Answer: Simple, get in touch with Radio Shack, the current 6.2 user manual is available as part #26-0316, the technical reference manual is part #26-2119. If you do not have these or have old versions, it is recommended that you obtain them while you can.

Question: *DATECONV gives a message "Can't convert dates on non-6.3 system disks"?*

Answer: Very simple to correct. At one time the disks' directory was marked as being a "SYSTEM" type disk. This can occur in several ways, none of which are damaging to your data. Why it happened is really not important at this point so let's cure it. It was suspected that this may become a problem for some of our users so we provided an "undocumented parameter" to override the checking for this in DATECONV. Use the command:

```
DATECONV :d (CS) [ENTER]
```

The (CS) is the parameter, it simply stands for Customer Service. This overrides the protection for NON-6.3 system disks, so be careful. Check any disk you use this on to make sure that it has only the /SYS files BOOT and DIR, if any other /SYS files are on the disk **do not use the CS parameter!** These /SYS files must be PURGED before you proceed. The (CS) parameter is to be used on data disks only.

Question: *How do I make a double sided system disk that will be laid out correctly?*

Answer: The main reason to have a system disk laid out "correctly" is so the system modules will be clustered around the directory to reduce access time to these frequently used /SYS files. Here is a very simple JCL which will do just that.

LS-DOS 6.3.1 Update Documentation

```
. DOUBLE/JCL
. Procedure to create a double sided SYSTEM disk
. Invoke via, DO DOUBLE (S=s,D=d)
.
FORMAT :#D# (SIDES=2,ABS,NAME="LSDOS63",Q=N)
MEMORY (A="A",B=17)
BACKUP /SYS:#S# :#D# (S)
MEMORY (A="A",B=11)
BACKUP :#S# :#D# (I)
MEMORY (A="A",B=1)
.
. A correctly structured, double sided SYSTEM
disk has been created.
. END
```

The above commands can of course be entered as individual commands at DOS ready with the same results. The important part of this procedure is the use of the MEMORY command to set the "A" flag. This is the Allocation flag for the operating system and the number in this flag is the cylinder on the disk where the file placement system will start to search-for space on the disk.

This is a very valuable thing to know about and use. This will allow you to structure your own disks in any manner that you wish, and to restructure and repack disks for efficiency. The normal setting of the "A" flag is for cylinder #1.

YES, those of you with hard drive should use a similar method when applying the system to your drive. Start the /SYS backup 2 cylinders below the directory of your hard drive, by setting the "A" flag accordingly. Do the rest of the backup starting about 5 cylinders below the directory. If you did not do something like this when you first set up your hard drive, you should reformat and do so now. This optimizing of the system on your hard drive will greatly increase the overall speed of your system.

Question: *On 6.2 I had a patch to have FORMAT default to double sided, double density and 40 tracks. What is the patch for this on 6.3?*

Answer: If the patch that you were using for 6.2 was:

LS-DOS 6.3.1 Update Documentation

```
PATCH FORMAT/CMD.UTILITY (D09,5B=02:F09,5B=00)  
[ENTER]
```

Then on 6.3 it would be:

```
PATCH FORMAT/CMD.UTILITY (D09,65=02:F09,65=00)  
[ENTER]
```

FORMAT will now default to DD/DS-40 track format. If you want a different format you will now have to specify same in the command line for format (see FORMAT in your 6.2 manual). **Note: This is not a recommended patch!** Users forget that they have made this patch and then call us saying "Your system is NO GOOD it won't even let me format a single sided disk".

***Question:** I followed all the instructions for updating my hard drive system but I still can't seem to get it to work properly. What do I do?*

Answer: The instructions for updating your hard drive system will work in almost all cases, but there are some systems that will require another method. If your system is one of the odd ones you can try a RE-SETUP.

Note: Before doing any upgrade to a hard drive, it is expected that you will have made a complete backup of all the important data and files off of that hard drive before you begin. This is only common sense.

First you **MUST** start with a **CLEAN** exact copy of your LS-DOS 6.3 DISK made with DISKCOPY. Next using *PURGE* or *REMOVE* get rid of some files you won't be needing on this disk, keeping in mind that this is to become your **BOOT DISK**. We suggest you *REMOVE DOS/HLP*, this should give you ample room on the disk.

Now, using *COPY*, put your hard disk **DRIVER** program onto this disk, this is the program used to set up your hard drive that has the extension */DCT*. Then copy the hard disk formatting program that came with your system onto the disk.

Now you must refer to the documentation that came with your hard disk software and perform a manual **SETUP** of the hard drive. You must set up the drive in such a manner that it will be partitioned in

LS-DOS 6.3.1 Update Documentation

the identical manner as it was before you began your upgrading. If a SYSTEM (SYSTEM=d) command is requested as part of the setup do not do it yet.

If all has gone well you should now have your hard drive accessible now. Test this by doing a DIR on each of the logical drive numbers now designated to be on your hard drive. If everything seems correct you should now put 6.3.0 onto the hard drive surface that contained your 6.2.0 do this with:

```
BACKUP :0 :d (S,I) [ENTER]
```

If you have followed the instruction drive 0 should still contain a fresh 6.3.0 that is missing the DOS/HLP file. Replace the "d" in the backup command with the present logical drive number for the hard drive partition that will become drive 0.

After the BACKUP has completed you should do a:

```
SYSTEM (SYSTEM=d) [ENTER]
```

Replace the "d" above with the same drive number used in the backup above. This will swap drive 0 with drive "d". Test things a bit by doing a DIR :0. If all is well you should get the directory of your hard drive SYSTEM platter. If this checks out OK then is time to save things, so do a:

```
SYSGEN (DRIVE=d) [ENTER]
```

Again replace the "d" with the same drive number as before. This will cause this SYSGEN to go to the 6.3.0 floppy that is in the PHYSICAL drive position :0. You should now be able to RESET the system and it should end up running correctly.

If you will be using the HELP command you should copy DOS/HLP onto hard drive :0 from a fresh copy of 6.3.0. This will complete the setup of the system. You now can reboot and then add any addition setup you may require in your SYSGEN or create an AUTO function. Just make sure that any SYSGEN or AUTO you do is specified to the BOOT floppy drive #.

LS-DOS 6.3.1 Update Documentation

If the above procedure fails in any way you will have to contact the company that provided you with your DRIVER and FORMATTER for the hard drive. You may have to RE-FORMAT the whole drive.

Question: *The anti-piracy protection scheme on LS-DOS 6.3 scares me. I'm afraid something might go wrong with it and I will lose valuable data?*

Answer: There is no "anti-piracy scheme" in LS-DOS 6.3.1! All purchasers of the product are expected to honor our copyrights.

Question: *This 6.2 program disk would "SYSGEN" when it was booted but after converting it to LS-DOS 6.3 it doesn't "SYSGEN" any more. How can I easily transfer the SYSGEN from a 6.2 to a 6.3 disk?*

Answer: Here is an easy procedure, but it doesn't work in all cases. If any system modules or options are part of your configuration, attempting this method may prove destructive; if only your own modules (say a hard disk driver) are part of your configuration, then you should have no problem. Lets assume that you have already made a copy of the 6.2 disk and updated it to 6.3 by following the usual instructions. Now you are ready to transfer the config/sys file. Boot up on the original (6.2) disk that has the configuration that you want to transfer and be sure to wait for the SYSGEN to complete. Next, put the disk you updated to 6.3 in drive :1 and do a SYSGEN (DRIVE=1). That is all there is to it, but test it very completely before you destroy your backups. This will fail if other files that are needed are not present. In this case, you should contact the supplier of that program for a version that will work on LS-DOS 6.3.

If you set up the original configuration that was SYSGEN'd, and just don't remember what it was, use the DEVICE command with the (B) parameter to look at the configuration.

Question: *I only have one 40 track drive so DISKCOPY doesn't work for me. How can I make a backup of the LS-DOS 6.3 master disk?*

Answer: DISKCOPY does not work on a single drive system, so a backup copy of your LS-DOS 6.3 master disk has to be made the old way. First, boot on LS-DOS 6.3, and then FORMAT :0 a fresh disk. Second, BACKUP :0 :0, using the LS-DOS 6.3 master as the source disk and the newly formatted disk as destination disk.

LS-DOS 6.3.1 Update Documentation

Everything else should proceed normally, using the instructions that you already have. You would update other system floppies from the hard drive after you have updated it to LS-DOS 6.3.

Question: *How can I make a bunch of blank, formatted disks, real quick?*

Answer: Format a blank disk in the normal fashion using the *FORMAT* command in LS-DOS 6.3. Now simply use *DISKCOPY* to make copies of this blank disk. *DISKCOPY* is very smart and only duplicates tracks that contain real data, blank tracks are just formatted and verified. Try it, you will be amazed at the speed.

Question: *My drive 0 is an 80 track drive. How can I make a backup of the LS-DOS 6.3 master disk?*

Answer: Surely, you have a 40 track drive somewhere in your system, use the same procedure as above for single drive systems, changing the drive number appropriately. An optimized 80 track system disk can be created using a procedure similar to the one given earlier in this addendum for creating an optimized double sided system disk.

Question: *I never upgraded to TRSDOS 6.2. How do I skip 6.2 and go directly to 6.3?*

Answer: *FORMAT* fresh disks with LS-DOS 6.3 and use 6.3's *BACKUP* to transfer the files from your 6.0 or 6.1 disks. Keep the originals until you are sure that all of your programs are working correctly on 6.3.

You still should have (or get) the TRSDOS 6.2 manual for the documentation of the many features that were added in 6.2.

Question: *Why doesn't TED work on BASIC programs or data files?*

Answer: This question surprised us but it keeps popping up. *TED* was never intended to be a direct *BASIC* editor. *TED* is an ASCII text editor. This means that *TED* is designed to edit text files that are in ASCII format. Most *BASIC* data files are not ASCII text files and there is no reason to expect *TED* to work on them at all. *TED* can work on *BASIC* programs that were saved with the "A" option and are, therefore, in ASCII format, but there will be no syntax checking.

LS-DOS 6.3.1 Update Documentation

Question: *After updating my disk and even using DATECONV, it still says TRSDOS62 in the heading when I list the directory.*

Answer: That field in the directory listing is just the disk name. You can change it to whatever you want with the ATTRIB command.

Question: *Why can't I edit out the "LIST OCATE" in my BASIC program?*

Answer: You may also be getting "EDIT" or "DELETE" or "AUTO" at the beginning of the some line in your BASIC program. Usually this will occur when you are loading in a program from some other source in ASCII format. What is happening is that if the first word in the line is not a reserved keyword, and if the first letter of the word is either an "A", "D", "E", or "L", that first letter is expanded to "AUTO", "DELETE", "EDIT", or "LIST", respectively. The way to correct it is to correct the word. There are always details of syntax to take care of when converting BASIC programs from one version to another. The search and replace function of a text editor or word processor can make it quite easy to take care of such details while you have the program in ASCII format.

Question: *My BASIC program worked on 6.2, but on 6.3 I get an out of memory message. What do I do about it?*

Answer: To add in the new enhancements to BASIC, it was necessary to increase the amount of memory that BASIC itself occupies, thereby reducing the available memory for programs. If your program uses close to the maximum available memory, it may not work on the enhanced BASIC, but that is not a big problem because you can use the BASIC from TRSDOS 6.2 with LS-DOS 6.3. The enhancements do not affect the run-time performance in anyway.

To put the 6.2 BASIC back onto your disks, first PURGE BASIC:d (INV) to delete all three 6.3 BASIC files. Then, do BACKUP BASIC:s :d (I), where drive s holds a disk with 6.2 Basic on it and drive d holds the destination disk.

Question: *Why aren't TED, DISKCOPY, and the new BASIC enhancements transferred to my disks in the update procedure?*

LS-DOS 6.3.1 Update Documentation

Answer: Because there may not be enough room! You may not want them on all of your system disks, anyway. It is easy enough for you to put these programs on any disk you want to with the *BACKUP* command. Don't forget that *BACKUP* can copy single files as well as groups of files.

Question: *Can I have the "KILL" command back?*

Answer: Sure, but there is a trade off. You can't have both "KILL" AND "REMOVE". "KILL" was just a synonym for "REMOVE", which means that they both did exactly the same thing. "REMOVE" is used in many commercial programs and if you use the following patch to rename "REMOVE" to "KILL", those programs may no longer work. With that caveat, here is the patch:

```
PATCH SYS1/SYS.SYSTEM6
(D02,15="KILLbb":F02,15="REMOVE")
```

The TWO lower case b's after KILL, are to be replaced with spaces before you do the patch and are important!

Question: *How can I read 6.3 files on my Model I or III?*

Answer: With the LDOS operating system on either the Model I or Model III, you can read LS-DOS disks. Moving between LDOS 5.3 and LS-DOS 6.3 will be no problem, of course. With LDOS 5.0 or 5.1, you may in some cases be denied access, which can be corrected by using TRSDOS 6.2 and ATTRIB to change the PROTection level to FULL and the owner and user passwords to blanks. LDOS 5.3 is not available for the Model I.

Question: *Why doesn't PRINT CHR\$(15) turn the cursor off like it used to?*

Answer: Actually, it still does, but there was a bug that is now fixed. What happens is this: when execution of a program in basic is ended, and you are returned back to the BASIC Ready prompt, BASIC is supposed to tidy up a few details, just in case the program left a mess. One of those details is to turn the cursor on. So when you do a PRINT CHR\$(15); from the BASIC Ready prompt, it looks like nothing happens. Indeed the net result is that you are back where you started. But put exactly the same statement inside a program,

LS-DOS 6.3.1 Update Documentation

followed by a statement that should present a cursor, INPUT A\$ for example, and the cursor is gone! Then comes the end of the program and the BASIC Ready prompt, complete with cursor. This can affect video that is routed to printers if they happen to respond to this code. If you BREAK out of a program, the cursor will be turned back on at Ready and will still be on when you CONTinue.

The current behavior is correct but for those of you that prefer it the way it was, here is a patch for you that makes the cursor behave just like it did before the bug was fixed:

```
PATCH BASIC/CMD.BASIC (D55,1B=00:F55,1B=EF)
```

Note! This is not a recommended patch.

Question: *What is the meaning of the LEVEL designator?*

Answer: Level is an indicator of the specific minor release within the major DOS version. Level designators may change when minor patches are implemented. This helps MISOSYS identify your specific DOS release. Please note that when MISOSYS creates revisions there may be no individual notification.

Question: *How can I access 6.3 data files from TRSDOS 6.X or LDOS?*

Answer: Restore the old style dating to a copy of the file or files by using the DATE=OFF parameter of the RESET command. For example, say you want to move a Scripsit document file, MYTEXT/DOC, from LS-DOS 6.3.1 to TRSDOS 1.3. FORMAT a disk under LS-DOS 6.3 specifying 35 cylinders single density. Copy the document file to that disk. Then issue the command, RESET MYTEXT/DOC:1 (DATE=OFF), assuming drive :1 contains that 35 track diskette. Then BOOT your TRSDOS 1.3 diskette and CONVERT :1 :0. That's all there is to it.

Question: *Will TANDY support LS-DOS 6.3, too?*

Answer: Yes. LS-DOS 6.3 is the official supported operating system for the Model 4/4/4D and BOTH Tandy and MISOSYS are supporting it.

