

LDOS  
S LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS  
OS LD  
DOS LD  
LDOS  
LDOS  
S LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LD  
OS LD  
DOS L

LDOS LDOS  
S LDOS LDO  
OS LDOS LD  
DOS LDOS L  
LDOS LDOS  
LDOS LDOS  
S LDOS LDO  
OS LDOS LD  
DOS LDOS L  
LDOS LDOS  
LDOS LDOS  
S LDOS LDO  
OS LDOS LD  
DOS LDOS L  
LDOS LDOS



LDOS LDOS  
S LDOS LDO  
OS LDOS LD  
DOS LDOS L  
LDOS LDOS  
LDOS LDOS  
S LDOS LDO  
OS LDOS LD  
DOS LDOS L  
LDOS LDOS  
LDOS LDOS  
S LDOS LDO  
OS LDOS LD  
DOS LDOS L  
LDOS LDOS

LDOS  
S LDO  
OS LD  
DOS LD  
LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS

LDOS  
S LDO  
OS LD  
DOS L  
LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS

LDOS  
S LDO  
OS LD  
DOS L  
LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS

LDOS  
S LDO  
OS LD  
DOS L  
LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS

LDOS  
S LDO  
OS LD  
DOS L  
LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS LDOS

V I E W F R O M T H E B O T T O M F L O O R

-----

First off I would like to correctly explain the construction of the LDOS version number, as it was not clearly explained in the last newsletter. In the last paragraph on the first page of that newsletter, a reference was made to (5.x.x). This reference, however, was not explained. Let us assume a Version number of 5.0.2A. The alphabetic character at the end of a version number is for maintenance purposes and will not always be present. It indicates small or insignificant maintenance patches to the system. If this letter appears or changes you may send in your disk along with \$5.00 and it will be updated.

The number on the right (in this case the "2") is the update number. This number is changed when substantial patches or minor feature changes or enhancements are applied to the system. These updates will usually include additional or replacement documentation. As long as the first and middle numbers in your version number match a new version you may send \$5.00 and receive this update.

The middle number (in this case the "0") indicates the series of the product. This number will change when major changes and enhancements are made to the system. This type of change will include large if not total changes in the manual. When this number changes we consider this to be a NEW product in the series. Trade-ups will be available if the first digit in your version number matches the new version. You will be changing products when going from a 5.0.x to a 5.1.x, and will receive a new manual, disk, serial number and a fresh warranty. Prices for a trade of this nature are dependent on the amount of work that went in to the new product and how different the new product is. Because this type of change will be a major one and the entire product will be replaced, these trade-ups will not be inexpensive.

Now for an explanation of the first number (in this case the "5"). When this number changes it indicates an entirely new system. Upward compatibility will be maintained for media transportability, and programs in LBASIC and most other high level languages or assembler programs that use the LDOS SVC (SuperVisory Call system) should function. There will be no updates, trade-ups, exchanges or trade-ins for this type of change as it will indicate a new era for the LDOS product line. However, this type of change will be very infrequent, probably years apart, as many man-years will go into a project that will change this number.

= = = = =

There is a Model I LDOS upgrade available. This version has been designated 5.0.3-A and is available for immediate shipment at the normal \$5.00 maintenance charge. Details of what this maintenance release contains will be found elsewhere in this newsletter. All patches that are used to create a 5.0.3-A are also in the newsletter. Even if you install all these patches to get a 5.0.3-A real quick, I would still advise that you send in your master disk for an "official" update. An official 5.0.3-A is identified by a sticker that will say 5.0.3-A and all files on a 5.0.3-A will be dated 09/05/81 or later. Sending in for an update will assure that you will have a "correct" update, as there are quite a few patches involved. It would be unfortunate if you mis-typed a patch and caused a major failure of your LDOS because of it.

If you already have an earlier 5.0.3 and wish to upgrade it to a 5.0.3-A then you have three patches to apply, details are given in the patch section.

It has come to my attention that hundreds of LDOS users are still using 5.0.0 or 5.0.1 LDOS. Please be aware that we did not put out 5.0.2 for the fun of it!! There were some serious problems that were corrected by 5.0.2. So if you are still running 5.0.0 or 5.0.1 send you Master disk along with \$5.00 for an upgrade to 5.0.3!!

= = = = =

All Model III LDOS users should check their disks! The current Version of Model III LDOS is 5.1.0-A and will have all files on your Master disk dated 09/15/81, or later. To check the dates on your Master disk, boot your Master disk, then do a DIR :0 (S,I,A). The date that appears in the last column of the display is the file date and should be 09/15/81 or later. If the dates on your files are earlier than 09/15/81 then you should return your Master disk, with \$5.00 to LSI for a maintenance update. There have been several small changes made to the system and some of them are important. So please check your disk and get an update if you need it.

= = = = =

For those of you that own a Model III from AM-Electronics, it has been found that some early versions of their controller board will not work with LDOS. This was solved promptly by AM-Electronics and a fix is available. If your system is from AM-Electronics and it doesn't seem to work with LDOS, contact AM-Electronics. They should be able to help you.

= = = = =

LDOS for the Model III TRS-80 was released in July and the reception has been fantastic! This new product is LDOS Version 5.1.0 and contains many new features that are not present in the 5.0.x series. The LDOS 5.1.x product is much larger than the 5.0.x LDOS. The manual is over 30 pages longer, a complete Disk Basic (LBASIC) is included. The Model I 5.1.x system will be ready soon and will come on TWO master disks (one just won't hold it anymore).

The list price of this new product, LDOS 5.1.x for both Model I & III, is \$169.00, effective October 1, 1981. LDOS 5.0.x will continue to be available at \$149 while supplies last. Of course, full support will continue for 5.0.x owners. NOTE: LDOS 5.1.x for the Model I will be available by the end of 1981. All users of LDOS 5.0.x will be allowed to "trade up" to 5.1.x.

This new upgrade is a COMPLETE product replacement. You will send LSI your entire LDOS package, manual, binder and master disk and of course a check for the correct amount. You will receive from LSI a completely NEW LDOS 5.1.x package, including a new one year warranty period.

To be fair to all LDOS owners who wish to convert to 5.1.x, a special program has been developed to allow for a varying cost for this upgrade. The price that you pay will be determined by how long you have owned the LDOS system.

Here is how it works:

The base price for the conversion is \$25.00. This was arrived at by the simple facts that the price for 5.1.x is \$20.00 more than 5.0.x and the shipping and handling cost is about \$5.00 - thus \$25.00.

You must add to this \$25.00 base price \$8.00 for EACH month or PART of a month that you owned the LDOS system (as of the day you send in for your upgrade). When you purchased the LDOS 5.0.x system will be determined by the proof of purchase that you provide to LSI. If you cannot provide proof of purchase then LSI will assume you purchased LDOS in February of 1981, and you must calculate your upgrade fee based on February. There will be NO EXCEPTIONS to this proof of purchase rule. We would prefer that you send a photocopy of your proof of purchase receipt, but if this is not possible you should send the original. We will try to return it with your 5.1.x package.

The Lowest price for a conversion will be \$33.00 under this system. The highest price, if you ordered the conversion in October of 1981, would be if you received your 5.0.x system in February or had no proof of purchase. In this case the conversion would cost \$97.00.

NOTE: Canada add \$5.00, all other foreign orders add \$20.00 for shipping.

If you wish to have the 5.0.3 upgrade and you are also upgrading to 5.1.x then you should send master disk along with your upgrade fee as computed above, an additional \$5.00 and a blank disk. We will put 5.0.3 on this blank disk and return it to you. You may then use your 5.0.3 until you receive your 5.1.x at a later date. (Your manual will also be required, but if you want to keep it until 5.1.x is ready we will send you a postcard when we want you to return your manual). IF YOU DO THIS BE SURE TO WRITE DOWN YOUR 5.0.x SERIAL NUMBER !!! This is very important as you will no longer have a master disk and you will need to know your serial number until you receive your new serial number with your 5.1.x LDOS. Your 5.0.x serial number will stay in force until you are sent your 5.1.x package. Your old serial number will be cancelled two weeks after you are sent your 5.1.x so be sure to register your new LDOS 5.1.x as soon as you get it !!

I M P O R T A N T

-----

Three important points should be made. First, although LDOS 5.1.x for the Model I is not going to be available until late in the year we will honor conversion orders starting October 1, 1981. If you intend to convert, then the sooner you send in, the less you will have to pay (at a rate of \$8.00 a month). Secondly - -

THIS IS NOT A MANDATORY CONVERSION !!!

The LDOS 5.0.x that you may have is an excellent product and LSI will continue to support it as long as there are users on their one year warranty or on renewed service agreements. You may want or need the new features of 5.1.x but you are not compelled in any way to convert to 5.1.x.

If you take advantage of this conversion offer you should be aware that you will get a fresh start on your warranty. All support will be dropped for your old serial number after you receive your 5.1.x, so make sure to register your new LDOS as soon as possible to be assured of full support.

Thirdly, we do not have an exact date for the release of 5.1.x on the Model I (other than by the end of the year). If you submit for your upgrade you will be without your manual, which would cause problems for all but the most proficient users. So, if you send in for the upgrade to 5.1.x and you DO NOT send in your manual, we will properly log in your order and when 5.1.x is ready you will receive a postcard asking for the return of your manual. When we receive your manual we will send you your 5.1.x. This way you should be without a manual for the shortest possible time, and can order your upgrade early to take advantage of a lower price.

= = = = =

It has been noted before, but I would like to remind all of you that our customer service hours are from 10am to 12 noon and from 4pm to 6pm Central time, Monday through Friday. I have heard from several users that they are having trouble getting through on the 800 line. Please try several times if the lines are busy. We do have TWO access lines and TWO customer service people, but occasionally both lines are tied up. I have watched this very closely and find that this is very rare. So if you can't get through, wait about 10 or 15 minutes and try again.

= = = = =

LSI is happy to announce the first in a series of extension packages for the LDOS product line. These packages will include utilities, filters, applications and other things useful to the programmer and user. The first package will be FILTER oriented, the second package will be FILE MANAGEMENT oriented. The prices for these packages will be in the \$50 to \$100 range.

The first package, called the LDOS FILTER PACKAGE, will be available by the end of the year and will sell for \$60.00.

LDOS FILTER PACKAGE IS FOR LDOS MOD I and MOD III ONLY !!

It will come complete with extensive documentation and will contain the following programs:

- XLATE/FLT ..... A complete translation filter system, for input and output. Included are a complete EBCDIC translate system and also a DVORAC keyboard translator. The user can very easily build any other translate tables that are needed for special use.
- STRIP7/FLT ..... Strips bit seven (the high bit) off of each character.
- UPPER/FLT ..... Converts every alphabetic character (a-z) to UPPER case.
- LOWER/FLT ..... Converts every alphabetic character (A-z) to LOWER case.
- SLASH0/FLT ..... Will cause a printer that is capable of backspacing, to do a backspace and type a "/" over every 0 (numeric zero) that is encountered.
- TRAP/FLT ..... Will trap and throw away a certain character each time that the character tries to go through the filter, any character (00 - FF) may be "trapped".

LINEFEED/FLT ..... Either add or remove a linefeed after each carriage return.

PAGEPAWS/FLT ..... Will stop pause after each Top-Of-Form character is printed and wait until enter is pressed to continue.

CALC/FLT ..... A keyboard filter to perform Hex/Decimal/Binary conversion. Hex addition and subtraction may also be done.

REMOVE/CMD ..... Removes each occurrence of a specified byte from a file.

ALIEN/DCT ..... Will allow special set up of a Drive Code Table. This will allow LDOS to READ alien double density disk formats such as NEWDOS and DBLDOS.

All these programs will be supplied on one single density LDOS disk. Documentation will be in LSI format for insertion in your LDOS binder. You must be a registered LDOS owner to purchase this product. You must provide your serial number when ordering this package. It will be shipped ONLY to the address at which your serial number is registered. Orders will be filled in the order that they are received, so order early if you wish to receive it shortly after release. This package is available from LSI distributors ONLY.

MISOSYS (703) 960-2998 / GALACTIC (414) 241-8030 / LOBO (805) 683-1576.

The FILE MANAGEMENT PACKAGE will contain a "screen oriented" TEXT EDITOR called "LED" (LDOS Editor) and a File oriented EDITOR for direct disk searching and modifying of bytes, both in HEX and ASCII (the name for this program has yet to be selected). This package will also contain a FILE-SPLIT utility, and an EXTENDED REPAIR command (to fix damaged directories) plus other little goodies. The FILE MANAGEMENT PACKAGE should be available early 1982. Full details on this package will be in the December newsletter.

= = = = =

As some of you may already know, LSI will introduce its first LDOS language package by the end of the year. This is an implementation of the "C" programming language, which we are calling "LC" (pronounced Elsie). It will be well documented (but will not be tutorial in nature). It is an integer-only "C" but we expect a floating point version to be available next year. Those who buy the integer version will be offered some sort of upgrade to the floating point "C" when it becomes available. The price for the "LC" will be in the \$100 to \$150 area. If you are interested, send us a letter (include your LDOS serial number) and we will send you a brochure. This product will be available to registered LDOS owners only.

More good news on languages! As most of you know, Microsoft Model I languages work just fine with Model I LDOS. This is not the case on the Model III, as Microsoft has not provided Model III versions of their languages to date. To solve this problem, you will find in this newsletter patches to allow Model I versions of the Microsoft packages to run on either the Model I or Model III. These patches are very small and should be easy to install. These patches pertain to Microsoft's MACRO-80, LINK-80, EDIT-80, FORTRAN and BASCOM. Also in this newsletter are small patches that can be applied to the Radio Shack packages RSBASIC and RSCOBOL so they will run properly on Model I & III LDOS.

LDOS users now have the following language products available: LBASIC, RSBASIC, BASIC/S, BASCOM, EDAS, MAC-80, FORTRAN, RSCOBOL and now LC, plus users have reported using many others. Of course SCRIPSIT and VISICALC can also be used.

We are actively considering other languages to be implemented in special LDOS versions. All languages that have become popular on micro computers are being considered. At this time we are certain we will have an EDITOR ASSEMBLER and probably a PASCAL. We will keep you posted as to what we are about to offer and when, as soon as firm decisions are made.

= = = = =

Some LDOS users are having a problem locating copies of the ORIGINAL MODEL I SCRIPSIT by Radio Shack to use with LDOS. Both MOD I and MOD III LDOS require a copy of MOD I Scripsit to be patched into an LDOS version. If you wish to use Scripsit on either LDOS and are not able to get a MOD I version #1.0 Scripsit, we will help. Buy a legitimate copy of MOD I or MOD III Scripsit from Radio Shack (any version will do). Now send LSI (Customer Service Dept.) the master disk out of that package, a new blank disk and a check for \$5.00. We will put a copy of both MOD I and MOD III Scripsit (already patched) onto the blank disk and return it to you along with your Master Scripsit disk.

= = = = =

We have been approached by many vendors to provide support for their hard disk Systems with LDOS. By the end of the year there should be many hard drive systems available featuring LDOS. There will be customized versions of LDOS for each vendor and the proper version will come with your hard drive hardware. LSI will directly support all of the "official" LDOS hard drive systems, but only those implementations that are done directly by LSI. Almost all software that runs on your present LDOS system will run with no alteration on an LDOS hard drive system. Watch for ads from the vendors supplying LDOS with their hard drive packages.

= = = = =

Many software vendors are now providing products that are either designed for use exclusively with LDOS or have been checked as to their compatability with LDOS. Rather than mention all their names here we suggest you call or write to the companies that you intend to buy software from and ask them about their support of the LDOS operating system. We are making every effort to assist all reputable software companies, who are willing, to make their products available in either LDOS specific or LDOS compatible versions.

= = = = =

We get a lot of calls and correspondence from users about LDOS working with other hardware and software. In most cases we are unable to be of much help. We do support LOBO hardware, Percom's doubler and now a doubler from Aerocomp. We also support several hard drive controllers. In software, we try to support major products from Tandy and Microsoft and we are mostly upward compatible with TRSDOS. We will support any software or hardware as long as our requirements are met by the vendor. These requirements are based on what the product is, how popular it is, whether our users want or need it, and will it increase the LDOS market.

We consider all these things (and more) and then determine IF we will support the vendor and what it will cost the vendor. There are very few things we have elected to support with our own funds. In most cases the costs of development must be born by the vendor seeking our support. So if you want some special hardware or software supported by LDOS, we are not the people to contact!! You should contact the vendor of that product, as they are the only ones that can arrange for this support.

= = = = =

We have been contacted by many hardware and software vendors that would like to advertise in this publication. This would be paid advertising of products that are supposedly compatible with LDOS. What we need to know is how our LDOS users feel about this. I would like to point out that if you, "THE LDOS USERS", do not want to see ads in your newsletter, then the newsletter will have to decrease substantially in size (maybe by 1/2). The production costs of this large of a publication (and we do wish to keep it large) can only be justified if LSI is compensated in some way. If we can get enough advertising at a high enough rate we may be able to send this newsletter by FIRST CLASS mail rather than bulk rate. You would then receive it about 3 weeks faster. Please send us a note and let us know what you think. NOTE: DO NOT call us on the 800 number to provide us with you feelings on this matter - please write!! The people that are handling this are not in the same department as customer service, so please write to LSI, ATTN: Marketing Dept.

U P D A T E   N E W S  
=====

Following are a series of patches to make a Model I 5.0.3 version disk. The following files have been changed from the 5.0.2 release, and should be patched onto all 5.0.2 disks. If you are using a version earlier than 5.0.2, you must send in for the 5.0.3 update.

SYS0/SYS was changed to allow SYSTEM (FAST) to be sysgened. Also, an OUT to port X'EA' will not be done when booting. Seek error handling in the disk driver was also improved.

SYS6/SYS was changed to fix the NOT operator when processing JCL files, and to prevent rerouting a device that had been routed to a file, from destroying the device name after the file is closed. Also, doing a RESET \*KI will now turn off Type Ahead if it is active.

SYS7/SYS was changed to correct a message used with Type Ahead, and to allow ATTRIB to leave a file's Visibility status untouched if not specified in the command line.

BACKUP/CMD was changed to prevent using the (X) parameter if drive 0 is not the source drive.

CMDFILE/CMD had one small change made involving offsetting of files that partially reside below 4200H.

FORMAT/CMD was changed to correct a problem causing a system crash when the last track is locked out during verify.



PATCH/CMD was changed to allow direct format (DXX,XX=) patches from the command line. Also, both X and D command line patch formats may be entered by using multiple lines separated by a colon. For example:

```
PATCH TEST/CMD (D00,12=55 66:D00,15=FF)
```

REPAIR/CMD was changed to strip the high bit off disk sector 0, byte 3 (directory cylinder byte). This will allow DOSPLUS double density disks to be repaired and read.

RS232/DVR and RS232L/DVR had changes made to correct problems with initialization of the UART parameters and handshake signals. The defaults for all line condition parameters will now be OFF, as stated in the manual. LX-80 owners may now find it necessary to specify DTR, if using certain modems or serial printers.

PDUBL/CMD was changed so an attempt to write to an empty drive is now recoverable with <SHIFT><BREAK>. Seek error handling was also improved.

Use the BUILD command to type in the following JCL and patch files for updating your 5.0.2 disks. When executing the JCL, be sure to specify the source and destination drives. For example:

```
DO BUILD503 (S=0,D=1)
```

would assign the source drive as 0, and the destination drive as 1. It is very important that the two SYS6 and SYS0 patches be applied in order - SYS0G before SYS0H, and SYS6N before SYS6O and SYS6P. In general all the patches shown here should be applied in the order that they appear.

```
. BUILD503/JCL
. Patches an LDOS 5.0.2 to make a 5.0.3
. S=disk containing patches
. D=disk to make a 5.0.3
patch repair/cmd.RS0LT0FF:#D# repaira:#S#
patch sys0/sys .RS0LT0FF:#D# sys0g:#S#
patch pdubl/cmd.RS0LT0FF:#D# pdubla:#S#
patch rs232/dvr.RS0LT0FF:#D# rs232c:#S#
patch rs232l/dvr.RS0LT0FF:#D# rs232ld:#S#
patch sys6/sys.RS0LT0FF:#D# sys6m:#S#
patch sys6/sys.RS0LT0FF:#D# sys6n:#S#
patch sys7/sys.RS0LT0FF:#D# sys7q:#S#
patch patch/cmd.RS0LT0FF:#D# patchd:#S#
patch backup/cmd.RS0LT0FF:#D# backupl:#S#
patch format/cmd.RS0LT0FF:#D# formatg:#S#
patch cmdfile/cmd.RS0LT0FF:#D# cmdfilea:#S#
. The following three patches are all that are required to
. make a 5.0.3 into a 5.0.3-A. These can be applied alone
. with this DO file by "DO BUILD503 (@ MAKEA,S=s,D=d)"
@MAKEA
patch sys0/sys.RS0LT0FF:#D# sys0h:#S#
patch sys6/sys.RS0LT0FF:#D# sys6o:#S#
patch sys6/sys.RS0LT0FF:#D# sys6p:#S#
//EXIT
```

Note that these patches are to be applied to a 5.0.2 LDOS only!! If you already have a 5.0.3 with file dates earlier then 09/15/81 then you need only apply the last three of these patches. If your file dates are 09/15/81 or later you do not need to apply any patches.

. REPAIRA/FIX

D00,41=CD CC 52  
D00,D8=3A 02 54 CB BF C9  
D00,48=FD CB 04 66 C2 FB 52 CD DD 52 CD 7F 53  
D01,8F=FD CB 04 AE 2E CB 7E FE 40 D8 FE 70 D0 2E CD CB 6E C8 FD CB 04 EE C9  
D00,B6=01  
D00,BB=00  
. End of patch  
\*\*\*\*\*

. SYS0G/FIX

D06,A6=33  
D0D,5B=00 00  
D0F,72=57  
D0F,79=1A  
D0F,8E=E6 82 C0 01 FE 01 C3 30 44 21 0F 43  
D04,9F=B7 C8 FE 07 28 3F 30 6F  
D05,16=01 88 05 FE 0A 28 0A 30 0D 06 0A CD A0 46 01 1A  
D05,26=02 CD A0 46 01 1A 0E A8 FE 0E 38 06 0E F4 20 02  
D05,36=0E AB CD A0 46 09 0A 12  
D05,42=E3 7E 23 32 F9 46 7E 23 66 6F 22 D4 46 E1 C5 CB  
D05,52=61 CC 54 46 D5 E5 21 EC 37 CD 46 46 7A 32 ED 37 11 EF 37  
D05,87=C8  
D05,96=B9  
. End of patch  
\*\*\*\*\*

. PDUBLA/FIX

D00,9F=33  
D00,EC=CA  
D00,EE=D3  
D00,F0=DA  
D00,F2=E1  
D00,F4=EF  
D00,FE=E9  
D01,02=09  
D01,13=B7 C8 FE 07 28 4B D2 9D 53  
D01,B6=01 A8 06 20 15 01 88 04  
D01,C5=D0  
D01,CB=D0  
D01,D0=FE 0E 38 06 0E F4 20 02 0E A9 CD D0 53 09 12 0A E3 7E 23 32 60 54 7E 23  
66 6F 22 22 54 E1 C5 CB  
D01,F0=61 CC 62 53 DD E5 D5  
D01,F8=DD 21 23 54 21 ED 37 CD 54 53 72 2B 11 EF 37  
D02,23=76  
D02,60=91  
D02,6F=82  
D02,B8=35 30  
. End of patch  
\*\*\*\*\*

```
. RS232C/FIX
D01,AD=33
D00,6C=CD 12 55
D00,8B=5D 52
D00,91=00 00
D00,B2=7A 52
D00,B8=00 00
D00,CA=3A 2C 54 CD 18 55
D01,27=CD
D01,29=55 B0
D01,35=B7
D02,34=1B
D02,3C=1D
D02,73=00 00 00
D02,84=80 F7 00 00 80 EF 00 00 80 9F 00 00 80 7F 80 00 80
X'5507'=E6 07 EE 07 47 DD 7E 03 E6 F8 C9
X'5512'=23 7E EE 80 C9 00
X'5518'=EE 80 C8 3A 2B 54 2F E6 08 DD B6 03 DD 77 03 C9
. End of patch
*****
```

```
. RS232LD/FIX
D02,44=33
D01,99=82
D03,04=00
D03,10=00
. End of patch
*****
```

```
. SYS6M/FIX
D0C,DB=01 F6 AF 78 C9
D24,6C=D5 28 0B
D24,7A=E5 CD EC 52 E1 D1
D04,FA=59
D05,11=5E 2C 56 C1 EB 3E 00 B7 28 16 5F 04 05 28 01 2B
D05,21=CD A9 4B 4D 54 67 68 7B CD A9 4B 61 B7 28 01 23
. End of patch
*****
```

```
. SYS6N/FIX
L63
X'5256'=C2 46 54
X'5446'=3E 1D BB C2 2D 40 F3 3E 0A CD 13 44 21 1F 44 CB 8E 3E CD 32 9C 43 FB
C3 2D 40
. End of patch
*****
```

```
. SYS7Q/FIX
D15,D4=F8
D0C,3D=EB
D0E,E5=F1
D0E,FA=EE
D11,17=2E 92 01 2E AA 01 2E BE 01 2E D5 26 58 CD 7B 44 C3 2D 40 00
. End of patch
*****
```

```
. PATCHD/FIX
D00,90=3A
. End of patch
*****
```

```
. BACKUPL/FIX
X'5988'=CD E5 60
X'60E5'=F5 B7 28 0C 3A C5 52 B7 28 06 21 8E 6B C3 AA 52 21 68 53 F1 C9
X'6B8E'="Source must be 0 with (X)"
X'6BA7'=0D
. End of patch
*****
```

```
. FORMATG/FIX
D07,3A=20 C2 52 C3
D07,5A=72 E9
D03,F2=7A 14 00 00
. End of patch
*****
```

```
. CMDFILEA/FIX
X'54C3'=CD 76 SA
X'5A76'=D5 CD 8D 59 D1 C9
. End of patch
*****
```

The following three patch are to create a 5.0.3-A from a 5.0.3. They must be installed on either a 5.0.2 (after the above patches) or to a 5.0.3 (unless the dates on the 5.0.3 files are later then 09/05/81).

```
. SYS0H/FIX
.
D0F,72=54
D0F,79=1D
D0F,91=C3 30 44 01 FE 01
. End of patch
*****
```

```
. SYS6O/FIX
.
D24,31=64 54 01 1C 64
D22,EE=6A 54
.
. End of patch
*****
```

```
. SYS6P/FIX
.
L63
X'547A'=21 FA 03 22 C1 4B FB C3 2D 40
.
. End of patch
*****
```

Items of General Interest

-----

Model III LDOS comes with a patch file to allow Scripsit to function properly with LDOS. As stated in the manual, this patch must be applied to a copy of MODEL I Version 1.0 Scripsit, NOT MODEL III. It will NOT work if applied to a copy of Model III Scripsit. Once patched, the Model I Scripsit will work properly with Model III LDOS.

Model III owners wishing to move programs from LDOS to Model III TRSDOS should put the files on a 35 track, single density disk and then use the TRSDOS Convert command to move the files. Be sure that it is a 35 track disk, as the TRSDOS Convert expects the directory to be on track 17.

=====

With LDOS, it is possible to assign a physical drive more than one logical drive number. Although this will work, it does present several problems. Since the DCT contains the last cylinder accessed for each logical drive, accessing the same physical drive using a different logical drive number will cause the drive to do a restore every time you switch drive numbers. This will degrade the overall efficiency of the system. This is a very dangerous practice and is NOT recommended!!!

=====

We have found that many direct connect modems need to have the DTR (Data Terminal Ready) line high before the modem will remain off hook. Therefore when using one of the LDOS RS-232 driver programs, it may be necessary to specify (DTR) as a parameter when setting the RS-232 driver.

=====

Once REPAIR/CMD has been used on a TRSDOS Model I, 2.3 disk, it initially will not be possible to read that disk with a normal TRSDOS. Also, the repaired TRSDOS disk will not boot, as the boot sector will not be able to read the DAM on the directory track. There are several ways to get around this problem.

To allow Model I TRSDOS to read an LDOS or repaired TRSDOS disk, and to allow a repaired TRSDOS disk to boot, apply the following patches TO THE TRSDOS DISK. A full series of patches to allow TRSDOS to fully use the X'F8' DAM are contained in the Model I 5.0.2 Update sheets mentioned below.

PATCH BOOT/SYS.POI (D00,DC=1C)  
PATCH SYS0/SYS.XOBIAAAA (X'46B0'=3C)

If you do not wish to patch your TRSDOS disks, you may put "repaired" disks back to an "unrepaired" state with the following procedure.

To provide cross compatability with Model I TRSDOS you must be able to switch a Model I LDOS disk back to write the OLD directory Data Address Marks (Model III LDOS disks must always have the NEW Data Address Marks, as it is physically impossible for them to read or write the old one). This is a very simple procedure. First make a clean backup of your Master Model I LDOS, then apply the following patches (Radio Shack Interface ONLY!):

PATCH SYS0/SYS.WOLVES (X'4699'=A9)  
PATCH PDUBL/CMD.RRW3 (X'5403'=A9)

Now clearly mark this disk with **\*\* OLD D.A.M SYSTEM \*\***. When you want to convert a disk back to the TRSDOS type of Data Address Mark simply Boot with this disk and do a REPAIR (ALIEN) on the disk you want to switch back. The conversion will be reversed and the disk will now be usable by TRSDOS & Model I LDOS but not Model III LDOS.

Do not run on this patched LDOS as it will continue to write the OLD style Data Address Marks to the directory. This will cause no great problem, but any disks written to by this patched LDOS will not be "DIRECTLY" usable on Model III LDOS. To make the disk again usable by Model III LDOS, use a regular Model I or III LDOS and do a REPAIR on that disk. It will then be back to being a "TRUE" LDOS Disk.

This type of conversion may be done as often as desired without problem. If you are doing this often, it would be better to patch TRSDOS to be able to deal with both types of Address Marks. You can find the information on how to do this on the Model I 5.0.2 upgrade sheets. If you don't have a 5.0.2 upgrade sheets write us a note (include your serial number) we will be glad to send you a set.

B A C K U P - B Y - C L A S S  
-----

The Backup-by-class feature of LDOS is a very powerful feature, but it is also very complex internally. There are a couple of things that are done when SYS0 is moved to the destination disk by a backup by class that are not readily apparent.

SYS0 is the heart of LDOS. It can also be referred to as the resident system. It is like a traffic control system for the entire operation of LDOS. When it is moved, certain other sectors of the source disk are also placed on the destination disk. These are sectors 2, 3, and 4 of track 0.

One reason these sectors are moved is that they contain special information storage used by the system, sectors 0 and 1 are not moved to assure that the proper type of boot will be on the destination disk. Another reason to move these is that the state of the SYSTEM (SYSGEN), On or Off, is stored within them. So if the source disk is sysgened then the destination will also be, and if the source disk is not sysgened, the destination disk will NOT be sysgened, even if it was before the backup begins.

This is a good time to mention the use of JCL to configure your system. Because you have to type in all the setup you desire for your system, why not type it into a JCL file so that it can be reused!! To do this, refer to BUILD in your LDOS manual (a suggested name for the file would be SYSGEN/JCL). After you have entered the statement to turn on or set up the features and options you want, simply execute the JCL file with the LDOS DO command. This will execute all the statements you typed in as though you were entering them as commands from the keyboard. When the DO is completed, type SYSTEM (SYSGEN) and the present setup will be saved on drive 0 as that system disk's configuration. Do not place "SYSTEM (SYSGEN)" in the JCL file!

CUSTOMER SERVICE

-----

The Customer Service department requests you use the following procedure when selling or transferring ownership of your LDOS system.

1) Notify us in writing, including your name and LDOS serial number, the new owners name, address, and phone number, and the date the transfer is to take effect.

2) Remind the new owner that his support is good only for the existing term of the original warranty (1 year). At the end of the original term, the support services (800 line, newsletter, and update service) are renewable for the then current support fee.

This should enable us to accurately keep track of the new owner and assure him of receiving any notices and newsletters he is entitled to.

When sending in your disk for an update, please do not use UPS when sending only a disk mailer. Their sorting machinery tends to mangle the disk mailers, and the disk!

While we're on the subject of diskettes, it seems that some people are getting back diskettes that won't turn in the jacket. These are the Verbatim disks only, as the problem does not occur with the Scotch disks. To repair the disk, follow these instructions:

Examine the edges of the disk, looking for places that appear flattened or crimped. The edges should be nicely rounded to allow the disk to turn freely.

If a flat spot is found, rub firmly along the edge of the diskette with a round object such as a plastic pen barrel perpendicular to the edge. This will reform the outer edge of the jacket and allow the disk to turn freely.

-----  
Since Radio Shack made some major changes to TRSDOS for the Model I and III, we have developed a program to move files off of the new TRSDOS 2.3B Model I disks. This is an LBASIC program called SALVAGE/BAS. It will copy files from a 2.3B or later TRSDOS disk to an LDOS disk.

```
10 REM--SALVAGE/BAS
20 REM--Program to get files from a TRSDOS 2.3B disk
30 REM-- onto LDOS disks, Model I or III.
40 REM--Use only on the Model I, 2.3B or later. Model III owners should
50 REM-- use CONV to get the files from the ThSDOS 1.3 disk.
100 LINE INPUT "Source filespec? ";SF$
110 LINE INPUT "Destination filespec? ";DF$
200 OPEN"RO",1,SF$
210 OPEN"R",2,DF$
220 FIELD 1,1 AS D1$ 'Find source FCB
230 FIELD 2,1 AS D2$ 'Find destination buffer
240 Z1=VARPTR(D1$):Z1=PEEK(Z1+1)+256*PEEK(Z1+2)
250 Z2=VARPTR(D2$):Z2=PEEK(Z2+1)+256*PEEK(Z2+2)
260 POKE Z1-32+3,Z2AND&HFF:POKE Z1-32+4,Z2/&H100 'Use 1 bfr
```

```

270 N=PEEK(Z1-32+12)+256*PEEK(Z1-32+13) 'Get ERN
280 IF PEEK(Z1-32+8)=0 THEN 300 'Adjust if needed
290 N=N+1:POKE Z1-32+12,NAND&HFF:POKE Z1-32+13,N/&H100
300 FOR I=1 TO N 'Copy file
310 GET 1:PUT 2
320 NEXT I
330 POKE Z2-32+8,PEEK(Z1-32+8) 'Transfer offset
340 POKE Z1-32-2,0 'Don't close source file!
350 END

```

-----

The LDOS Quarterly has a rather lengthy patch section this issue. Besides the patches to upgrade your LDOS 5.0.2 to 5.0.3, we also have developed patches for several commercial programs, including the Microsoft and Radio Shack language products.

As we have said in the past, to our knowledge, all Microsoft products run on Model I LDOS without any patches. Unfortunately, Microsoft has not yet come out with Model III versions for their Fortran, Bascom, or Macro-80 packages, so to fill the gap we have developed patches for the Model I versions of these programs allowing them to run under Model III LDOS. All of the patches are designed in such a way that the patched programs will run on EITHER the Model I or Model III under LDOS. The patches are for the latest versions of the Microsoft products at the time of this writing. The specific version numbers are noted in each patch. Contact Microsoft about upgrading to the latest versions if you have an earlier version of any of these programs. Note: LDOS owners should install the BRUN patch even if the program is to be used only on the Model I. The patch corrects the LOC, LOF, and EOF functions to work properly with random files of any record length.

It was also necessary to develop some small patches for Radio Shack's Cobol and RSBASIC compilers. The Cobol package will refuse to run under anything but TRSDOS 2.3B on the Model I or TRSDOS 1.3 on the Model III unless patched, and both packages do some peculiar things in attempting to handle the <Break> key, which may not work under LDOS. Although this may not be clear from the documentation, these programs were designed to run on either the Model I or Model III, so the patches can be applied to a copy taken from either the Model I or Model III disks in the Radio Shack package. The resulting patched programs will run on either Model I or Model III LDOS. Due to the incompatibility between TRSDOS 2.3B and earlier versions of TRSDOS, it was also necessary to write a special program to recover files from 2.3B disks. The REPAIR utility is not capable of repairing 2.3B diskettes (see SALVAGE/BAS, page 15). If you have LDOS 5.1, it is recommended that you use the CONV utility to transfer the Model III package to an LDOS formatted disk.

. F80/FIX

- . This patch for Microsoft Model I FORTRAN-80 (version
- . 3.42) allows it to run under Model III LDOS. After the
- . patch is applied, the same program will run on either
- . the Model I or Model III.

```

X'57A4'=CD 51 7E 20 0B 21 11 44 22 E1 7A 2E 4B 22 58 7E
X'7E51'=3A 25 01 FE 49 C9 C3 73 44
X'7B78'=57 7E
X'7B9B'=57 7E

```



```
X'7BB5'=57 7E
X'7F94'=57 7E
X'7E5D'=00 00 00
X'593D'=" LDOS V3.42a "
. End of patch
```

. FORLIB/FIX

. This patch converts FORLIB/REL from the Microsoft Model I  
. Fortran package (version 3.42) for operation under  
. Model III LDOS. After FORLIB is patched, programs  
. linked with it will run on either the Model I or Model  
. III under LDOS.

```
D5D,A3=5D
D63,CA=81 49 12 53 92 55 25 00 00 13 51 E0 05 78 C8 00
D63,DA=00 19 00 00 02 19 27 FF F8 00 00 22 A1 00 03 A1
D63,EA=28 05 FC 49 15 12 48 02 00 18 A8 22 44 7C 98 0D
D63,FA=2E 98 C8 58 06 24 43 4C 53 46 4C 8C 81 00 32 44
D64,0A=54 38 E8 00 05 24 49 4E 49 54 8C 82 80 62 44 94
D64,1A=F4 64 C4 79 C0 00 00 9E
. End of patch
```

. EDIT80/FIX

. This patch converts Microsoft Model I EDIT-80 (version  
. 1.32) to operate under Model III LDOS. The patched  
. program will run on either Model I or III LDOS.

```
X'79E9'=22 18 53 CD F6 63 20 0B 21 11 44 22 0E 7A 2E 4B 22 75 67
X'63F3'=C3 2D 40 3A 25 01 FE 49 C9 00 00
X'7A7C'="LDOS V1.32a "
. End of patch
```

. LINK80/FIX

. This patch allows Microsoft Model I LINK-80 (version  
. 3.43) to run under Model III LDOS. After the patch is  
. applied, the same program will run under either Model I  
. or Model III LDOS.

```
X'5203'=CD F9 58 20 0B 21 11 44 22 28 52 2E 4B 22 56 60
X'58F8'=C9 3A 25 01 FE 49 C9
X'54CC'="LDOS Ver. 3.43a"
. End of patch
```

. M80/FIX

. This patch allows Microsoft Model I MACRO-8g (version  
. 3.43) to run on the Model III under LDOS. After the  
. patch is applied, the same program will run on either  
. the Model I or the Model III.

```
X'5203'=CD 5D 96 20 0B 21 11 44 22 DF 92 2E 4B 22 64 96
X'965D'=3A 25 01 FE 49 C9 C3 73 44
X'9379'=63 96
X'939C'=63 96
X'93B6'=63 96
X'97A0'=63 96
X'9669'=00 00 00
```

X'6B0D'="LDOS M80 3.43a"  
X'6B1B'=09  
X'6B1C'="14-May-81"  
X'6B25'=09  
X'6B26'="Pg."  
. End of patch

. CREF80/FIX  
. This patch allows the Microsoft Model I CREF-80 program  
. to run on Model III LDOS. After the patch is applied,  
. the same program will run on either Model I or Model  
. III LDOS. CREF-80 carries no internal version number,  
. but the patch was developed for the version that comes  
. with M80 3.43.  
X'5203'=CD 39 5C 20 0B 21 11 44 22 1A 59 2E 4B 22 40 5C  
X'5C39'=3A 25 01 FE 49 C9 C3 73 44  
X'59AF'=3F SC  
X'59C9'=3F SC  
X'SC45'=00 00 00  
. End of patch

. BASCOM/FIX  
. This patch modifies Microsoft Model I BASCOM 5.23 to run  
. on the Model III under LDOS. After the patch is  
. applied, the same program will run on either the Model  
. I or the Model III.  
D79,FF=00  
X'5203'=CD 0C 93 20 0B 21 11 44 22 A7 8F 2E 4B 22 13 93  
X'930B'=C9 3A 25 01 FE 49 C9 C3 73 44  
X'9042'=12 93  
X'9065'=-12 93  
X'907F'=12 93  
X'944E'=12 93  
X'6497'="LDOS BASCOM 5.23a, Copyright 1979,80"  
. End of patch

. BRUN/FIX  
. This patch modifies Microsoft Model I BRUN to run under  
. Model III LDOS. After the patch is applied, the same  
. program will run on either the Model I or the Model  
. III. BRUN does not carry an internal version number,  
. but this patch was developed for use with the version  
. that comes with BASCOM 5.23. The patch also permits  
. /CHN file names to be typed in lower case, and corrects  
. the LOC, LOF, and EOF functions.  
D38,FF=00  
X'5201'=CD 11 53  
X'530C'=CD 1A 53 18 0D CD EB 53 21 11 44 CC 25 54 2A 49 40 C9  
X'53E8'=C3 2D 40 3A 25 01 FE 49 C9 13 13 C3 5A 44  
X'5219'=50 59 CD 1C 44 00 00 00 00  
X'5401'=CD EB 53 28 0B CD 70 44 36 20 23 CD 6D 44 18 09 CD 33 30 36 20 23 CD  
36 30 3E 11 21 EA 52 CD 56  
X'5421'=6C D1 C1 C9 22 1B 53 2E 4B 22 5B 52 2E 2B 22 32 7B 22 42 7B 22 77 7B

```
2E 6D 22 A3 79 C3 59 79
X'7949'=50 59 13 13 CD 4B 44 21 00 00 D1 C1 C8 2B C9 E9 22 F4 53 2E 70 22 BB
79 2E 58 22 4E 79 C9
X'799E'=CD 17 58 7D 21 5A 44 C5 D5 E5 CD 3C 7C E1 CA 97 7D 50 59 13 13 18 0A
00 CD 17 58 7D 21 5D 44 18
X'79BE'=E6 CD 58 79 60 69 D1 C1 C9
X'7A10'=50 59 CD F1 53 50 59
. End of patch
```

```
. LINK80B/FIX
. This patch allows Microsoft Model I LINK-80 (version
. 3.41) to run under Model III LDOS. After the patch is
. applied, the same program will run under either Model I
. or Model III LDOS.
X'5203'=CD EF 58 20 0B 21 11 44 22 28 52 2E 4B 22 2A SF
X'58EE'=C9 3A 25 01 FE 49 C9
X'54BF'="LDOS Ver. 3.41a"
. End of patch
```

```
. RSCOBOL/FIX
. Patch for Radio Shack COBOL compiler version 1.3B
. to run under LDOS Model I and III
X'A196'=4F
X'A1D8'=50
X'9A4F'=D2
X'9ASA'=3E C9 32 0C A0 00
. End of patch
```

```
. RUNCOBOL/FIX
. Patch for Radio Shack COBOL Version 1.3B runtime package
. to run under LDOS Model I and III
X'AE6B'=4F
X'AE7E'=50
X'9B38'=38
X'9B4A'=3E C9 32 B0 A9 00
. End of patch
```

```
. CEDIT/FIX
. Patch to RS Cobol editor version 1.3B to run under LDOS
. model I or III.
X'5832'=3E C0 32 13 5C 00 00 00 00 00 00
X'5C05'=C3
. End of patch
```

```
. RSBASIC/FIX
. This patch corrects the Radio Shack Basic Compiler 2.4
. for proper operation under LDOS Model I and III.
. Either the Model I or Model III version may be patched,
. and the resulting program will run on either model.
X'9971'=C9
X'997B'=00 00
. End of patch
```

```
. BEDIT/FIX
. This patch corrects the Radio Shack Compiler Basic editor
. program for proper operation on Model I/III LDOS.
. Either the Model I or Model III version may be patched,
. and the patched version runs on both models.
X'58DB'=00 3E C9 32 7C
X'58E4'=00 00 00
X'5C6E'=C3
. End of patch
```

U S I N G V I S I C A L C W I T H L D O S by Roy Soltoff  
-----

The difficulty in obtaining directory information from most Model I applications stems from the lack of an operating system facility for doing so. This is not so with LDOS. That deficiency has created convoluted techniques to appear in applications so as to extract such information. A uniform procedure such as opening the DIR/SYS as a file and reading it record by record can not be universally applied across the DOS's since TRSDOS ( tm) restricts that procedure by two methods. First, the hash index table (HIT) entry for DIR/SYS in TRSDOS is not the code that would arise if DIR/SYS is operated on by the system's hashing algorithm. Secondly, the DIR/SYS file's protection level is NO ACCESS.

Programs such as ELECTRIC PENCIL (remember that one?), DISK\*MOD EDTASM, EDAS, and VISICALC (as well as hundreds of others) have used a technique of creating a file control block already in an "open" state referencing the entire disk as a file. Thus by positioning to a record and issuing a READ SECTOR command, any sector on the disk can be read. When a file is opened in the normal manner, its location on a disk is irrelevant to the user - the operating system takes care of positioning the drive to the record that satisfies your read request. If you "fool" the system into thinking a file is already open, the catch is that you must know where you want it to read. The number one problem, therefore, is to calculate the position of the directory on the disk.

For over two years, the only media supported on the Model I was single density - with its 10 sectors per track. DOS gurus have already discovered that the third byte of the BOOT/SYS sector(s) contains the cylinder number of the directory. Aha! A simple manner to locate the relative track number for the directory would be to read the BOOT/SYS sector, grab the third byte (usually a X'11' on SDEN 35 track media) and multiply by ten, since there are 10 sectors per track.

Enter double density! Enter LDOS and 8" drives and double sided drives and multi-headed hard drives and... No longer do we find 10 sectors per track, but different numbers depending on the disk configuration being accessed. All of the applications that were written with directory accessing routines to work under TRSDOS (and most commercial applications MUST be written to be supported under TRSDOS), will not work properly unless the media being accessed is single density, 35 track, 5-1/4" floppy disk media.

Enter the VISICALC patch! VC/FIX cures your ails so you don't need to down all those ales. The patch creates a VC program whose directory scroll feature works under LDOS. Not only that, the patched VC works under Model I LDOS or Model III LDOS. One program serves all! While I was about it, VC now uses LDOS's keyboard driver so the type-ahead is faster. Watch Out - it's fast.

A couple of changes are in order. Visicalc uses two SHIFT functions for cursor control. The keys used to activate them have changed. Use the following:

CONTROL B - switches cursor between blinking and steady

CONTROL L - causes highlight cursor to disappear

The CONTROL key is entered in the usual LDOS manner - SHIFT and DOWNARROW.

While I am discussing documentation, let me note a few of the undocumented features in VC. The VC gurus may have discovered the "/X" command already. For those who haven't, let me elaborate. The "/X" command can be used to perform some sleight of hand. It is quite interesting and potentially powerful. Here are its uses:

- /X> coord - Another GO TO COORDINATE; however this will establish the displayed grid with the entered coordinate in the upper-left hand corner.
- /XVn - Splits the window vertically with window 1 occupying as many character positions as specified by "n". In addition, the column width is set to 3.
- /XHn - Splits the window horizontally with window 1 occupying as many data lines as specified by "n". In addition, the column width is set to 3.
- /X- - Added by LDOS. This command will perform a screen refresh. Why, do you ask, is a screen refresh needed? Well, if you use a minIDOS command, the screen will be rather unpleasing. Thus, you can refresh the display with /X-.
- /X! - A do-nothing command but capable of being easily patched to add your own VC command.

That's it. Enjoy your new Visicalc.

```
. VC/FIX - Copyright (c) 1981 Logical Systems, Inc.
.
. Patch to convert Visicalc Model I version 1.20z to
. compatibility for Model I and Model III LDOS operation.
.
D00,18=CD BE 54
D00,DC=CD BE 54
D00,F3=00
```

```

D02,C4=2B 00
D02,CA=3A 25 01 FE 49 3E 00 2A 11 44 C8 2A 49 40 C9 44 49 52 01 F2 D0 54 2F
53 59 53 3A 30 0D C6 30 32 48 A0 C9
D34,BB=4C 44 4F 53 20 56 31 2E 30 20 28 63 29 20 27 38 31 20 4C 53 49 20 20
56 43 20 28 63 29 20 27 38 30 20 53 41 49
D47,93=08
D47,BC=CD 54
D47,C6=CD D7 54
D47,D2=00 00 00 00 00 00
D48,AC=06 00 21 80 A0 11 40 A0 CD 24 44 C2 A1 97 3A 4C A0 3D 3D 32 9D 98 CD
91 99 CD 36 44 C8 C3 A1 97
D48,D0=07
D4B,0B=C9
D4B,25=C9
. End of patch

```

-----

The following patch was done by Tim Mann to allow ORCHESTRA-80 to run with LDOS.

```

. ORCH80/FIX - 08/18/81 - Tim Mann
. Fixes to make ORCH80 run on LDOS 5.0
.
. Fix the stack problem
X'60AC'=E0 41
X'60C6'=E0 41
X'610E'=E0 41
X'6132'=DE 41
X'6164'=DE 41
X'65DD'=E0 41
X'697F'=E0 41
X'69B0'=E0 41
.
. Pick up HIGH$ instead of doing memory test
X'69E6'=2A 49 40 2E 00 25 2B 22 80 59 C9
.
. Fix the DIR command
X'5200'="DIR/SYS:0"
X'5209'=0D
X'5BAC'=CD FC 59 21 24 52 0E 02 58 54 FB CD 24 44 F3 C9
X'5EEE'=00 00 FE 38 30 E6 32 08 52
X'5F03'=CD AC 59 CD 42 44 11 00 52 CD 36 44 28 07 FE 1C 28 38 C3 DE 5C
X'5F4A'=C3 09 5D CD 28 44
.
. Enable interrupts while opening files
X'69F1'=FB CD 20 44 F3 C9
X'5E6E'=B6 59
X'5EB1'=20 52
X'6853'=F1 67
X'5220'=FB C3 20 44
.
. Allow patch areas to be saved
X'6864'=00 52
X'687A'=65
. End of patch

```

## LDOS 5.1 - Enhancements and New Features

=====

LSI is proud to announce the upcoming release of LDOS 5.1 for the Model I. As was noted in the VIEW FROM THE BOTTOM FLOOR section, LDOS 5.1 will be available towards the end of the year. This section of the Newsletter will outline the enhancements and new features contained in LDOS 5.1.

- 1.> Any LDOS command which requires parentheses around the parameters may be entered without the closing parenthesis.
- 2.> The PROT Library command has been eliminated. ATTRIB has been extended to include the functions of PROT.
- 3.> The XFER command has been eliminated. COPY now contains an (x) parameter to allow the transferring of files when dealing with non-system diskettes.
- 4.> The CONV command has been added to the system. This command will allow you to transfer files created on the Model III under TRSDOS 1.2 or 1.3 to an LDOS diskette. Note that Model I owners who wish to utilize CONV must have two drives, and be capable of reading double density with their system.
- 5.> Some LDOS drivers, filters and Library commands will reuse their original memory allocations if they are established, turned off, and re-established. The list of these is as follows:

- SPOOL library command
- KI/DVR (Keyboard driver - see #7.)
- MINIDOS/FLT (MiniDOS Filter - see #8.)
- PR/FLT (Printer Filter)
- KSM/FLT (KeyStroke Multiply filter)

- 6.> The Job Log (\*JL) now has its own driver, and is now established via the SET command.
- 7.> The Keyboard (\*KI) now has its own driver. This driver activates the use of the <CLEAR> key, and must be established if advanced keyboard features are to be utilized (KSM, MiniDOS, etc). In addition, the type-ahead and screen print options are enabled using the Keyboard driver.
- 8.> The MiniDOS filter has been added. This is an additional keyboard filter which allows the keyboard driver to intercept certain keyboard inputs and immediately act on them. MiniDOS commands are issued by depressing <CLEAR>, <SHIFT> and an alphabetic key. The following functions are available when using the MiniDOS filter.

- Toggle the clock display on or off.
- Enter the system DEBUGger (if activated).
- Display FREE space for a specified drive.
- Kill a file.
- Display a disk's directory.
- Repeat the last DOS command.
- Issue a Top of Form to the lineprinter.

9.> A Supervisory Call (SVC) table can now be loaded into high memory for use by assembly language programmers. It contains most documented system entry points and routines.

10.> Some SYSTEM modules may now be made resident in high memory by using the SYSTEM library command. This will allow you to perform certain functions that normally require these SYSTEM modules to be on a disk. For instance, the user of a two drive system will now be allowed to perform a BACKUP-by-class using non-system diskettes, provided the proper SYSTEM modules have been made resident. Residing SYSTEM modules also increases the speed of the system.

11.> Any Physical drive in your system may now be established as Logical drive 0. This can be accomplished by using the SYSTEM library command.

12.> Because of the installation of the Keyboard driver (KI/DVR), the parameters JKL and LOWER have been removed from the SYSTEM Library command. In addition, the UPDATE parameter is not needed in the Model III and has also been removed.

13.> The TYPE parameter of the SYSTEM library command still exists, but serves a different function. Rather than establishing the type ahead feature, it merely turns type ahead on or off. Type ahead is established using the Keyboard driver (KI/DVR).

14.> Drives may now be software write protected. This is accomplished using the SYSTEM Library command.

15.> With a minor modification to SCRIPSIT, the SPOOLer will now work in conjunction with SCRIPSIT. The despooling speed (output) of the SPOOLer has also been increased.

16.> APPEND now has two additional parameters. The ECHO parameter will now echo characters to the screen if a " devspec to filespec/devspec" APPEND is performed. This can be used when, for instance, you wish to APPEND the keyboard to a file. The STRIP parameter will "strip" off the last character of the file being APPENDED to, before the APPEND is performed. This can be useful if you wish to, for instance, APPEND two SCRIPSIT files together.

17.> Aside from having the (x) parameter added to it (see #3), the COPY Library command has also had the ECHO parameter added to it. This will function in the same manner as the ECHO parameter of APPEND (see #16).

18.> The DEVICE Library command will now display additional information. Included in this is whether or not the diskette is write protected (software or physically), options which are currently active (e.g. KI/DVR, MiniDOS, TYPE, etc.) and SYSTEM modules which have been made resident in high memory.

19.> A "NOT partspec" may be specified for some Library commands and utilities. For example, by entering the following command:

```
DIR -/CMD:0
```

you will be shown a Directory of drive 0 of all files that do NOT have the extension "/CMD". This NOT partspec is available with the DIR and PURGE Library commands and the BACKUP utility.



20.> A specific date, or a range of dates may be specified when utilizing the DIR and PURGE Library commands, as well as the BACKUP utility. The system will use the file's last modification date to determine whether or not that file is to be included in the specified operation.

21.> Along with the above mentioned modifications to PURGE (see #19 & #20), an additional enhancement has been made. Unless specified by use of parameters, PURGE will only deal with VISible, non-SYStem files when performing the PURGE.

22.> Along with the above mentioned modifications to BACKUP (see #19 & #20), two additional parameters have been added. The NEW parameter will BACKUP only those files not already on the destination disk. The OLD parameter will BACKUP only those files already existing on the destination disk.

23.> One enhancement was made to the FREE Library command. If the FREE command is given with a drivespec, a FREE space map will be displayed for that particular drive. This map will show the used, available, locked out, and directory grans assigned, along with the disk's size, density, and number of sides.

24.> The MEMORY command has acquired a new parameter. The (CLEAR) parameter will fill memory locations X'5200' to HIGH\$ with X'00'.

25.> Issuing the TIME Library command without parameters will display the current time on the screen. If the Job Log is active, this time will also be written to the Job Log. (If the DATE command is given with no parameters and the Job Log is active, the date will also be written to the Job Log).

26.> The FORMAT utility has had two parameters added to it. The (QUERY) parameter will query you as to the type of FORMAT you require. If not specified, the (QUERY) parameter will default to OFF. The (SYSTEM) parameter has been added, and deals with formatting a hard drive.

27.> LCOMM has had the following characters added to its set of characters available directly from the keyboard.

X'1B' <escape>	X'5E' <caret>
X'1C' <file separator>	X'5F' <underscore>
X'1D' <group separator>	
X'1E' <record separator>	X'7E' <tilde>
x'1F' <unit separator>	X'7F' <delete>

In addition, several features have been added to LCOMM.

An ACCEPT-LINEFEED function has been added. LCOMM normally ignores the first linefeed received after a carriage return. If this function is turned on, all linefeeds will be accepted. This function replaces the AUTO-LINEFEED function.

An 8-BIT function has been added. The 8-BIT switch is used to indicate that all 8 bits of each character received from the communications line are valid. If not turned on, bit 7 is stripped from each character received. This function replaces the LCOMM function.

A HANDSHAKE function has been added. If turned on, LCOMM will respond to the following four control codes received from the communications line:

```
X'11' DC1   - Resume transmission
X'12' DC2   - *FR ON
X'13' DC3   - Pause transmission
X'14' DC4   - *FR OFF
```

The DC2 and DC4 characters function identically to the \*FR ON and \*FR OFF programmed function keys. DC3 causes transmission through the \*CL device to be halted until a DC1 is received.

Internal changes have also been made to LCOMM to allow a dumb terminal to operate at baud rates higher than 300. Downloading of files must remain at a baud rate of 300

28.> The RS232 driver has an added parameter and has also been had enhanced.

As specified by standard RS232 conventions, a TRUE condition means a logic 0, or a positive voltage. A FALSE condition means a logic 1, or a negative voltage. The RS232 driver will now treat the line condition parameters DSR, CD, CTS and RI in the following manner:

If specified ON, the driver will observe the lead and wait for a TRUE condition before sending each character.

If specified OFF, the driver will observe the lead and wait for a FALSE condition before sending each character.

If not specified, the lead will be ignored.

In addition, a (BREAK) parameter has been added to the RS232 driver. This parameter determines whether the driver can set the system BREAK, PAUSE or ENTER bits.

29.> LBASIC has had several enhancements. It now comes as a program rather than as a patch to Radio Shack Basic. Loads and saves are done at high speed, except when saving in ASCII. An additional overlay has been added. It contains the error handling routine used by LBASIC, and also contains a string array sort routine, referenced by issuing a CMD"O" command.

Because CMD"O" was previously used by the system to turn off the DEBUGger, a change was also made to CMD"D". The CMD"D" command may now be given with an optional switch. This switch may be used to turn the DEBUGger on or off, and hence replaces the function of the former CMD"O" command.

The command CMD"L","filespec" has been added to LBASIC. This command will allow you to load a machine language routine from LBASIC.

The command CMD"B","switch" has been added to LBASIC. This command will allow you to enable or disable the <BREAK> key.

CMD"P" now performs a different function. Rather than performing a screen print, CMD"P" will now return the status of the printer to a variable. A screen print is done with a CMD"\*.

The RESTORE command has had an optional parameter added to it. Entering the command RESTORE nnnn (where nnnn represents a line number in an LBASIC program) will restore all DATA statements in lines whose numbers are greater than or equal to the line number specified in the RESTORE statement.

The RUN command has had two parameters added to it. These parameters primarily come into play when using the RUN command as an LBASIC program statement which will chain programs together. The (V) parameter will allow you to save all variables established in one program, and utilize them in another program. The (line number) parameter will allow you to specify the line number where execution of the program is to begin.

A new command has been introduced into LBASIC. This command is SET EOF# (where # refers to the buffer # associated with an open disk file). This command may be used to reset the End Of File (EOF) marker of a random file. This is a very convenient way of shortening a random file that may contain unwanted records at the end.

=====

#### MANUAL NOTES

-----

Following is a collection of things that were not in the LDOS manual, or were not explained clearly enough for everyone to understand (according to our customer feedback reports).

To exit back to LDOS from the regular (non-extended) DEBUG, use the command G402D<ENTER>. This will bring you back to the LDOS Ready prompt.

The PRINT library command used in some operating systems to list a file to the lineprinter is part of the LDOS command LIST.

Although the SYSTEM Library command describes how to set up most disk drive parameters, no mention is made of setting the number of tracks or the disk density. These two parameters are automatically established during the formatting process. To save a particular disk configuration with the SYSTEM (SYSGEN) command) place diskettes formatted with the desired number of tracks and density in the proper drives and do a DEVICE Library command before doing the sysgen. This will cause the desired Drive Code Tables to be saved in the configuration file. Be aware that LDOS will automatically pick up the density and number of tracks on a diskette the first time it accesses that drive, even if the DEVICE command command was not given.

The Model III manual had a section listing all files on their Master LDOS disk, and describing their functions. This is a modified version for Model I owners. It will also describe how to construct a minimum system disk for running applications packages.

#### FILE GROUP - SYSTEM FILES (/SYS)

-----

The descriptions of the /SYS files can be found in SYSTEM OVERLAYS near the end of the technical section in the existing manual.

FILE GROUP - UTILITIES

-----

- BACKUP - Used to duplicate data from one disk to another.
- CMDFILE - A disk/tape, tape/disk utility for machine language programs.
- FORMAT - Used to put track) sector, and directory information on a disk.
- LCOMM - A communications package for use with the RS-232 hardware.
- LOG - Used to log in a double sided diskette in drive 0. Also updates the drive code table information the same as the DEVICE library command.
- PATCH - Used to make changes to existing disk files.
- PDUBL - Double density disk driver program for use with the Percom Doubler II in the Radio Shack interface. Includes double sided support.
- REPAIR - Used to correct certain information on non-LDOS formatted disks.
- TWOSIDE - A double sided disk driver program for use with the Radio Shack interface) single density operation only.

FILE GROUP - DEVICE DRIVERS

-----

- RS232 - Used to configure and access the RS-232 hardware, Radio Shack interface only.
- RS232L - Used to configure and access the RS-232 hardware, LOBO LX-80 interface only.

FILE GROUP - FILTER PROGRAMS

-----

- KSM - Establishes the KeyStroke Multiply feature, which allows assigning user determined phrases to alphabetic keys.
- PR - Allows the user to format printed output.

FILE GROUP - BASIC AND BASIC OVERLAYS

-----

- LBASIC - Enhanced Disk Basic program created from the LBASICP1 and LBASICP2 patch files.
- BASIC/OVN - Renumber overlay used with LBASIC's CMD"N" feature.
- BASIC/OVX - Cross reference overlay used with LBASIC's CMD"X" feature.

## CREATING A MINIMUM CONFIGURATION DISK

-----

All files except certain /SYS files may be removed from your run time drive 0 disk.

For operation, SYS files 1, 2, 3, 4, 8, and 10 should remain on drive 0. SYS 11 must be present only if any JCL files will be used. Both Libraries (SYS 6 and SYS 7) may be removed if no Library command will be used. SYS 5 and SYS 9 may be removed if the system DEBUGger is not needed. SYS 0 may be removed from any disk not used for BOOTing.

When using LBASIC, BASIC/OVN and OVX may be removed if no renumbering or cross referencing will be done.

The presence of any Utility, Driver, or Filter program is totally dependent upon the user's individual needs. Most of the LDOS features can be saved in a configuration file with the SYSTEM (SYSGEN) command, so the Driver and Filter programs won't be needed in run time applications.

The passwords for LDOS files are as follows:

System files	(/SYS)	Update password = WOLVES
Filter files	(/FLT)	Update password = GSLTD
Driver files	(/DVR)	Update password = GSLTD
Utility files	(/CMD)	Update password = RRW3
LBASIC		Update password = BASIC
BASIC overlays	(/OV\$)	Update password = BASIC
CONFIG/SYS		Update password = CCC

MASTER PASSWORD = RSOLTOFF (allows access to any file)

## The JCL Corner by Roy

-----

A philosophical question is what's the most powerful feature of LDOS? That's almost akin to what's the best car? I will not attempt to answer the latter or the former as my answer is subjective at best. One of the most powerful is perhaps Job Control Language. JCL is a very complicated and little understood function. This corner will attempt to shed some light on various procedures available under JCL. Those users implementing JCL procedures may wish to donate suggestions to this column. You are encouraged! The less time I have to spend writing newsletter articles, the more time I have available to add features into LDOS. Since I have not supported a command to COPY ROY:0 to ROY:1 (CLONE), I can at best work on one thing at a time.

This issue's JCL example was implemented to aid in transferring a group of LDOS source files from the hard drive to another drive in the system for archival purposes. Since our LDOS development system has seven drives on-line, the archival can at times be mini-floppy or floppy. Thus, it was desirable to create a JCL procedure to segment the entire set of source files into groups and allow run-time selections for group name, source drive, and destination drive. It's also desirable to provide defaults for standard parameter values so minimal user entry is needed.

The following JCL is an abbreviation of the one satisfying the need. It is called "@COPYLDOS" and is but one of the many JCL procedures in our PROCLIB/JCL file. The groups are noted as SYS for all source decks generating /SYS files, LIB for all source decks generating modules included in the LDOS library (SYS6 and SYS7), and UTIL for all other source decks. The source hard drive is primarily defined as drive 5. The operation to perform was to copy all files if no group name was entered. If one or more group names were entered as parameters, then only those groups would be copied. Examine the JCL //ASSIGN statements to uncover how these defaults were established. The //IF compound logic macro is the key in defaulting to all groups if no group name was entered. It was this macro statement that revealed a bug in JCL which was corrected in release 5.0.3.

```
@copyldos
. *==**
. procedure to copy LDOS /ASM files
. *==**
//if -s
//assign s=5
//end
//if -d
//assign d=2
//end
//if -sys&-lib&-util
//set sys
//set lib
//set util
//end
//if sys
copy proclib/jcl:#s# :#d#
.
copy sys11/asm:#s# :#d#
//end
//if lib
copy genisam/asm:#s# :#d#
.
copy liba2/asm:#s# :#d#
//end
//if util
copy format1/asm:#s# :#d#
.
copy jl/asm:#s# :#d#
//end
//exit
```

## U S I N G @ P A R A M by Roy Soltoff

-----

A number of users have requested additional technical information on how to use the @PARAM vector in Assembler programs. Since the use of @PARAM can make the job of handling parameter input quite simple, the topic was chosen for this issue's QUARTERLY.

@PARAM is used in practically all LDOS Library commands and Utilities as well as Filters, Drivers, and languages (LBASIC for one). Since you are already familiar with all LDOS commands (aren't you?), you can recognize the wide range of input syntax parsed and interpreted. The @PARAM vector is used to decode TRUE/FALSE parameters (by either entering or not entering a parameter word), YES/NO parameters (by using PARM=Y or PARM=N), ON/OFF parameters (by using PARM=ON or PARM=OFF), decimal values (by entering PARM=dddd), hexadecimal values (by inputting PARM=X'xxxx'), and character string values (by entering PARM="characterstring"). @PARAM functions with input in either upper case or lower case - even with hexadecimal digits (A-F equally acceptable as a-f). This is truly a remarkable and powerful function.

A little bit of history is in order. Since the geneology of LDOS stems from TRSDOS and follows the VTOS branch, let's look at what TRSDOS offers. The scant documentation in the TRSDOS technical section made no mention of @PARAM. Thus few people knew of its existence even in TRSDOS. It's there; however it is not as powerful as what is in LDOS. TRSDOS supports all parameter entry identified except character string input. It also doesn't support lower case entry. VTOS implemented character string entry but omitted lower case support. It is interesting to note that NEWDOS80 (tm) omitted the @PARAM vector entirely from their system so that any application using the system call would not work! I have not examined the other DOSs to ascertain their support of this important vector - but if they don't support it, they are not TRSDOS compatible. One sad note is that Tandy apparently didn't realize the importance of this vector, since they removed decimal interpretation from their Model III DOS, changing the syntax of hexadecimal entry to 0xxxx, with the preceding zero required if the first hexadecimal digit started with A-F. This got them into trouble with their DUMP command. But enough of the other DOSs. Let's discuss LDOS.

The LDOS technical manual is actually rather complete in its discussion of @PARAM. Of course, "complete" is a relative term and if you are not related to me, then it may not be complete enough for you (you see, I wrote it). I'll try to "fatten up" the documentation here.

Let's assume we have an application that offers the user varying options to set up the function of the application. In LBASIC, this may be number of files or protected memory size. In BACKUP, this may be the diskette master password or date range of files to select. In RS-232 drivers, this may be whether CTS is to be honored. How do we get this information to the program? We could prompt the user by a prompt message for each and every parameter that needs to be determined. Experienced users soon get tired of prompts. Inexperienced users get extremely frustrated with requirements for rigid syntax of entry options. How can we satisfy everyone - from novice to expert? Why, by using @PARAM.

We will propose a hypothetical application requiring the determination of five options:

(1) A length field used in ascertaining the number of print columns of output. This should default to 80 to denote an 80 column printer if no entry is made. The range should be limited to 32-255.

(2) A module specification field to indicate whether line feeds are to be added after carriage return, removed after carriage return, or no checking is to be performed.

(3) A title field to be placed on each page of output. In addition, paging is to be suppressed if no titling is desired. Furthermore, the default is to incorporate paging unless otherwise specified by the user.

(4) A prompting specification to note whether prompts for changing paper are to be made at the appropriate time if sheet paper is used or omitted if tractor feed paper is used. The default should be no prompting.

(5) A translation option for converting a character on output. This should default to no translation.

The first thing required by the system designer is to designate "words" for the command line parameters. They should be chosen to be as easily remembered as possible. They should be greatly correlated in definition to the function they are specifying. Additionally, abbreviations should be considered in addition to the full "word". Thought should be given to using words whose first character is different for each parameter so as to provide single character abbreviations. However, if any parameter is omnipotent, care should be exercised in designating an abbreviation.

In the example above, we will choose LENGTH, FEED, TITLE, PROMPT, and XLATE parameter words for the options 1-5. We will also abbreviate these as L, F, T, P, and X. Your application's documentation will fully explain the purpose of the parameters. A typical command line entry could be:

```
URPROG (length=132,title="LDOS is great",xlate=x'0e00')
```

The command line could just as easily have been entered as:

```
URPROG (t="LDOS is great",x=x'0e00',l=132)
```

Note that not only are abbreviations used, but the order of appearance in the command line is irrelevant. Also note the obvious command line parameters are enclosed in parentheses. In LDOS 5.0 and beyond, you can take some liberties with the character string and hexadecimal syntax. The closing double quote is not required by @PARAM. It skips over the string to either the closing double quote, the comma separator, the closing right parenthesis, or the ending carriage return. Hexadecimal entries can drop the closing single quote. This leeway permits entry of such command lines as:

```
URPROG (t="LDOS is great)
```

```
URPROG (t="LDOS is great,x=x'0e00)
```

Effective with LDOS release 5.1, the closing parenthesis is not even required. You're saying there must be a catch. How can this do all that?



Easy - you must follow some rules and implement some coding in your URPROG. Not very much coding is required, though.

One additional point needs to be made. When you execute a command line) the command interpreter is activated (@CMNDI). If a LIBRARY name is specified, the system's library module is activated. If a program name is entered (the system first tries a default extension of /CMD if the user does not supply one) the program will be loaded and transfer will be performed to the program's transfer address which is located at the end of the load module (following the X'0202'). When control is passed to the program) register pair HL contain the address of the first character following the program name entered. The character which is obviously something other than A-z, a-z, or 0-9 is considered a separator character (because if it was one of those it, would be part of the program specification - also it could be an out-of-sequence filespec separator "/.:"). The @PARAM routine will search the command line for the left parenthesis starting from the address pointed to by HL. It will ignore blanks while it looks for the "("; however, if it finds a non-blank character other than "(", it will immediately return. You may note that some LDOS commands use entries after the program name but before the parameter field. It means that these must be parsed first by your program before issuing the @PARAM call.

The prologue of URPROG might go something like this:

```

URPROG   PUSH     HL           ;Hang on to INBUF$ pointer
         LD      HL,HELLO$    ;Point to hello message
         CALL   @DSPLY       ;Display message to screen
         POP    HL           ;Recover INBUF$ pointer
         LD      DE,PRMTBL$   ;Point to parameter table
         CALL   PARAM        ;Go parse all of the parms
         JP     NZ,PRMERR     ;Go to error handler if bad entry
         .
         .                   ;The rest of URPROG
         .
UREXITP  JP      EXIT        ;Exit of your program
HELLO$   DB      31,'Some friendly message',CR
;*****
;       This is the parameter table. Note its entries are
;       all 6-characters in width. The address specified by
;       the parameter vector follows each parameter "word".
;       In addition, the table is ended with a zero byte.
;*****
PRMTBL$  DB      'LENGTH'     ;Length parameter
         DW     LPARM+1
         DB      'L'
         DW     LPARM+1
         DB      'FEED'      ;Line feed parameter
         DW     FPARAM+1
         DB      'F'
         DW     FPARAM+1
         DB      'TITLE'    ;Title parameter
         DW     TPARAM+1
         DB      'T'
         DW     TPARAM+1
         DB      'PROMPT'   ;Prompt parameter
         DW     PPARAM+1

```

```

DB      'P
DW      PPARAM+1
DB      'XLATE '      ;Translate parameter
DW      XPARAM+1
DB      'X      \
DW      XPARAM+1
NOP
END      URPROG      ;This is the ending zero byte

```

The PRMTBL\$ is going to be structured similarly for all tables. The convention I used of specifying the address vector as LABEL+1 will become immediately obvious once you inspect the method of using the result in URPROG. As an aside, let's look at two conventions of referencing the 2nd byte of a 3-byte instruction.

```

METHOD1 LD (LABEL+1),HL      ;Load HL into the "nn" field
.
.
.
LABEL   LD      BC,0      ;P/u the value loaded

METHOD2 LD (LABEL),HL      ;Load HL into the "nn" field
.
.
.
LD      BC,0      ;P/u the value loaded
LABEL   EQU      $-2      ;The "nn" field is 2-bytes back

```

I prefer to use the first method, thus that's the manner in which my parameter table references vector addresses. Use the method you are most comfortable with. My personal suggestion is to choose one technique and stick with it. Otherwise you will find yourself getting into trouble as you forget which you were using.

Now that the @PARAM system function has parsed the entered `command` line, how do we utilize the "values" it interpreted while still supporting our defaults and conditions? Well, bear in mind that if the user has not entered a parameter word, nothing will be entered by @PARAM into the address vector specified by the parameter table. Therefore, an initial condition can be supplied in the coding. Also, the initial value coded will be dependent on just what condition you want the default to be. Let's see how this would work.

```

.
.      ;Some front end code
.
;***==*
; Here is where we pick up the length parameter. Note
; that it is initialized to 80 if there is no user entry
;***==*
LPARM  LD      BC,80      ;Pick up the entry
INC    B      ;Test hi-order for zero
DEC    B      ;It must be zero for range check
JP     NZ,LBAD      ;Bad length if range >255
LD     A,C      ;P/u the lo-order length

```

```

CP          32                ;Must be >= 32
JP          C,LEAD            ;Bad length if range < 32
;*****
;          The length parameter has been tested for proper range.
;          It can be used in URPROG where needed by either stuffing
;          the accumulator where needed or by picking up the value
;          later by a LD A,(LPARM+1) instruction.
;*****
.
.
.
;*****
;          Here is where we pick up the line feed parameter. Based
;          on the conditions specified, we need a three-way te st.
;          What has to be ascertained is whether the user specified
;          FEED=ON, FEED=OFF, or didn't even enter FEED. The ON/OFF
;          entries are the same as TRUE/FALSE specifications and
;          result in a -1/0 value respectively (ON = -1, OFF = 0
;          We therefore must de fine a default value which is
;          neither 0 nor -1.
;*****
FPARM      LD          BC,1          ;We will use a "default" of 1
           LD          A,B          ;Merge the hi and lo orders
           OR          C
           JR          Z,RMVFEED    ;Remove line feed if FEED=OFF
           INC         A            ;If FEED=ON was specified, A=X'FF'
           JR          Z,ADDFEED    ; thus A would be zero after the INC
;*****
;          The line feed parameter has now been handled. It is left
;          up to our readers to provide routines for RMVFEED and
;          ADDFEED. The best forwarded to LDOS Support will be
;          entered into the next QUARTERLY.
;*****
.
.
.
;*****
;          The title parameter needs to default to ON per our
;          conditions. This would mean that if no TITLE was
;          supplied in the command line, the user would be prompted
;          to enter it (userfriendly). On string parameters,
;          @PARAM will load the addr ess of the first character of
;          "string" into the vector address specified in PRMTBL0.
;          URPROG will then have to parse the string until it finds
;          one of the string terminating characters.
;*****
TPARM      LD          BC,-1        ;Force the default to be TITLE=Y
           LD          A,B          ;Check on entry of T=N
           OR          C            ;Merge hi and lo orders
           JR          Z,NOTITLE    ;To user provided routine
           INC         A            ;Check if T=Y or no entry
           LD          HL,PMTITL$   ;Init pointer just in case
           CALL        Z,GETTITLE   ;Go prompt & get title if only T=Y

```

```

;*****
;   The GETITLE routine would have to display the prompt,
;   provide an input means, then place the address of the
;   first character of string into register pair BC.
;   Otherwise, reg BC already has the address of that char.
;*****
.
.           ;Your routine for parsing the title
.           ;character string belongs here.
.
;*****
;   The prompt parameter will be an easy one. Its default is
;   PROMPT=OFF and no other special conditions need be met.
;*****
PPARM      LD          BC,0           ;Zero because the default is OFF
           LD          HL,FLAG$      ;Let' 5 set a flag for t his one
           RES         0,(HL)        ;Init flag to off
           LD          A,C           ;Only the lo-order is needed
           OR          A             ;Test the entry
           JR          Z,$+4         ;Skip the next instruction if P=N
           SET         0,(HL)        ;Set the flag if P=Y
.
.
.
;*****
;   The translation parameter is the last one to retrieve.
;   In order to provide a default of no translate character,
;   the code will use a zero value for this test. It is
;   important to note that since the entry is a 16-bit
;   value, your documentation must clearly note which order
;   is the character to test. If in X'xyy', we denote "xx"
;   for the test character and "yy" its translated value,
;   then "yy" becomes the lo-order byte when loaded while
;   "xx" becomes the hi-order byte.
;*****
XPARM      LD          BC,0           ;Note the zero default
;*****
;   That's all there is to it. We coul d, of course, test
;   for an 0 value and set a flag to indicate no XLATE
;   option entered. Then later test the flag first before
;   checking on a XLATE match. However, it would probably
;   take just as long to test for the option as it would to
;   test for the character so we will not use a flag.
;*****
.
.
.
;*****
;   Here is some code that could use the translate feature:
;*****
LD          BC,(XPARM+1)           ;P/u the test characters
CP          B                     ;Translate this character?
JR          NZ,$+3                 ;Go if no match
LD          A,C                   ;Use the translation character
LD          DE,PRDCB$             ;Point to Device Control Block
CALL       @PUT                   ; and put the character

```

```

;***=***
;      We could have used:
;          CALL      @PRT
;      in lieu of:
;          LD DE,PRDCB$
;          CALL @PUT
;      but not many assembler programmers realize the power of
;      using @PUT calls instead of specific device calls so I
;      thought I would use it here.
;***=***

```

This dissertation should now be sufficient for you to use the @PARAM vector in your application programs. If you still require additional assistance, our hourly rates are available on request. Happy programming!

Using Double Sided Drives With LDOS - by Tim & Roy  
-----

Unlike most other TRS-80 operating systems, LDOS is designed to work with multi-headed disk drives. Since a double sided drive has two heads that both access the same diskette, it would seem sensible to treat both sides of the diskette as a single volume which is twice as large as a single sided disk. This is what LDOS does. In fact, LDOS treats the second side as track and sector numbers identical to the first side but with a side identifier. This is where the term "cylinder" originates. Hard drives may have 2, 4, 6, or 8 heads - with identically numbered tracks on each "platter" constituting the cylinder. Unfortunately, since few (if any) other operating systems are capable of doing this, many disk drive manufacturers have their double sided TRS-80 drives jumpered to behave as two separate drives. The purpose of this article, therefore, is to explain how to set up your floppy drives and cables properly for use with LDOS. We will deal primarily with 5" drives) but a few comments will be made at the end about 8" drives.

If you are using a Model I with the Radio Shack expansion interface, there is an additional problem. The original standard set up for 5" drive cabling did not include a side select lead, and there were only 3 drive select leads provided. The Radio Shack interface was not designed to work with double sided drives, and so no side select output was provided on the drive cable. As luck would have it, though, the lead Radio Shack chose to use as the fourth drive select (pin 32) was later defined as the standard side select lead! That's why a double headed drive can work as two separate drives if cabled as drive 3 and 4. Thus the Radio Shack interface does have a side select line, but if it is used for that purpose) only three different drives can be selected. The Lobo LX-80 and the Radio Shack Model III both have provisions for selecting up to four double sided drives) so this problem does not occur with them.

Most drive cables for the TRS-80 have pins removed from the connectors to control which drive is selected by which drive select line. Since Radio Shack used the side select line as a drive select, this pin will have been removed from all connectors except the last on the connector. For this reason, we recommend that you use a cable with all pins intact if there are any double sided drives in your system. Observe that some cable manufacturers may have disabled drive select leads not by pulling the pins on the connector but rather by removing the pin splice that contacts the ribbon cable within the connector.

Therefore, if your cable has all pins intact, you should still verify drive select continuity on each connector with an ohmmeter or other continuity tester.

If you use a "solid" cable, it then becomes necessary to program your drives internally to determine which one will select on each drive specification. After this is done, any drive can be plugged into any slot in the cable, without changing its physical drive number. All your drives must be programmed, including any single sided drives that may be in your system. Programming a drive is a fairly simple operation. Remove the cover from the drive, and look on the main logic board for a DIP socket containing either several wires or a DIP shunt. Typically there will be seven positions in the shunt, labelled DS1, DS2, DS3, MUX or MX, HL, MH, and one spare. Some drives may have a DS4. First of all, the MX and HL must be open, and the MH shorted for TRS-80 operation. The MX must be open for all systems having more than one drive. The HL and MH arrangement cause the drive's head to be loaded on motor-on. One (and only one) of the DSx positions should be shorted for operation with a solid cable. DS1 sets the drive to be physical #1 as displayed by DEVICE (typically assigned as logical drive 0), DS2 as #2, DS3 as #4, and DS4 (if present) as #8. If you are using a Radio Shack Model I interface, DS4 cannot be used. If you have a Model III, DS1 and DS2 on the internal drive cable are physical positions #1 and #2, while DS1 and DS2 on the external drive cable are #4 and #8. DS3 and DS4 are unused. If you are using an LX-80 with four 5" drives, you should be aware that the LX-80 brings out its DS4 on pin 34. This may or may not agree with the DS4 jumper position in the drive, if there is one.

Double-headed drives may have been previously modified to be used as two separate drives in some systems. If such is the case, they may require additional activity depending on what was previously done to them. Any jumper wire soldered from pin 32 is suspect. Rather than attempt to advise you of any jumper that requires cutting, you are recommended to contact the drive manufacturer or the store where your drive was purchased for specific instructions on arranging your drive for single-drive operation.

Some cautions are necessary for those using double headed drives in mixed hardware systems. What happens if you inadvertently insert a diskette formatted for two-sided operation into a single-headed drive? The answer is different depending on your hardware complement.

On the Model III, the disk controller returns an error status if an I/O attempt is made when the side byte in a sector's identification field does not match the side selected. That's pretty fail-safe. However, you may be able to start writing over an existing file that starts on the first side but continues onto the second and not get the error until an I/O is attempted on the second side. A two-sided FORMAT attempt in a single-headed drive will result in every track locked out so that attempt will be absolutely fail-safe.

In a Model I, the story is not so good. The LX-80 interface disk drivers attempt to perform the side verification because the controller hardware is expected to support side verification. Unfortunately, no error status can be ascertained because the Fuji 1791-compatible disk controller chip does NOT support side verification as does the Western Digital 1791. The Fuji was used in the LX-80 because the WD-1791 has another problem that occasionally causes the system to lockup.

If you are going to FORMAT a diskette as double-sided in a double-headed drive, make sure the diskette is capable of two-sided operation. A mini-floppy diskette has no discernable difference other than the diskette label denoting two-sided. Eight inch floppies are another story. The index hole on a double-sided eight inch diskette is in a different location than is the index hole in a single sided diskette. That was done so the disk drive hardware could know if the diskette was two-sided. The double-sided drive ( Shugart SA-850 for example) will NOT be able to access the second side unless the index hole in the media denotes it as double-sided. You will get NO error if you attempt to format a single-sided eight inch diskette as two-sided in a disk drive capable of two-sided operation; however, the diskette will be completely UNUSABLE.

On the Model I with the Radio Shack Interface, another tale unfolds. The WD-1771 single-density controller chip does not support side verification but the WD-1791 double-density chip does. The PDUBL disk driver was not written to select side verification since it uses both the 1771 and 1791 depending on its current density of operation.

The bottom line recommendation is to readily mark whether a diskette is single sided or double sided. This can be done by obtaining colored Pres-a-Ply dots (or other brand) at a stationary store and affixing a specific color dot to each double-sided diskette. Also identically mark your double-headed drives. This will guard against inserting a double-sided diskette into a single sided drive. Until such time as the hardware permits side verification, no software fix is possible. Incidentally, there is absolutely NO problem with inserting a single-sided diskette formatted under LDOS into a double-headed drive. The operating system is smart enough to detect and use it as single-sided.

As a final note, all double-sided disks used under LDOS must be formatted by LDOS.

#### USER CONTRIBUTED PROGRAMS -----

This section of the newsletter contains programs and documentation files contributed by LDOS owners. Some of these files are also in the data base of the LDOS MicroNET bulletin board. Please remember that these are not "official" LDOS programs - any questions, comments, etc. should be addressed to the person that contributed the program, not to LDOS Support. This Section of the newsletter is specifically meant for interchange between LDOS users. We are always looking for contributions for this section and to our MNET database.

Note that the hex program listings must be put into the proper load module format before they can be used. There is a program included in this section called BINHEX/BAS, which will do the conversion for you. To use the program, follow these steps:

- 1) Use Scripsit, Electric Pencil, or the BUILD command to create an ASCII file containing the hex code. DO NOT leave spaces between the hex characters - the spaces were put in for readability only!. The ASCII files must NOT contain more than 254 characters (127 byte pairs) per line.
- 2) Enter LBasic, and Run BINHEX/BAS, choosing the "Hex to Binary" mode.

BINHEX/BAS

-----

Following is the listing for the LBasic program, contributed by Tim Mann.

```
10 REM -- Hex to binary/Binary to hex file converter
20 REM -- Tim Mann
30 CLS:PRINT:PRINT"Hex to binary/Binary to hex"
35 PRINT"      file converter":PRINT
40 CLEAR 5000
50 GOSUB 58000
100 PRINT "Type 1 to convert a binary file to hex"
110 PRINT "      2 to convert a hex file to binary"
120 PRINT:INPUT D
130 PRINT
140 ON D GOTO 400,200
150 GOTO 100
200 LINE INPUT "Hex file name: ";HF$
210 LINE INPUT "Binary file name: ";BF$
220 OPEN"I",1,HF$
230 OPEN"O",2,BF$
240 IF EOF(1) THEN 320
250 LINE INPUT#1,D$
255 IF D$="" OR D$="OK" THEN 240
260 FOR I=1 TO LEN(D$) STEP 2
270     PRINT#2,CHR$(FND2(MID$(D$,I,2)));
300 NEXT I
310 GOTO 240
320 CLOSE
330 PRINT:PRINT"Done":PRINT
340 GOTO 100
400 LINE INPUT "Binary file name: ";BF$
410 LINE INPUT "Hex file name: ";HF$
420 OPEN"RO",1,BF$,1
430 OPEN"O",2,HF$
440 FIELD 1,1 AS F$
450 FOR I=1 TO 30
455     IF EOF(1) THEN 505
460     GET 1
470     PRINT#2,FNH2$(ASC(F$));
480 NEXT I
490 PRINT#2,
500 GOTO 450
505 PRINT#2,
510 CLOSE
520 PRINT:PRINT"Done":PRINT
530 GOTO 100
58000 DEF FNH1$(X)=MID$("0123456789ABCDEF",(X AND 15)+1,1)
58010 DEF FNH2$(X)=FNH1$(X/16)+FNH1$(X)
58040 DEF FND1(X$)=INSTR("123456789ABCDEF",LEFT$(X$,1))
58050 DEF FND2(X$)=FND1(RIGHT$(X$,1))+16*FND1(RIGHT$(X$,2))
58070 RETURN
60000 END
```



SLASH0/FLT

-----

Those of you who received the first LDOS Quarterly may have noticed that it was difficult to tell the letter 0 from a zero. This was because our daisy wheel printer did not have a print wheel with a slashed zero on it. To correct this problem, we wrote the following filter program (suggested by one of our readers). It will allow printers capable of handling a backspace character (X'08') to print a slashed zero (0).

```
05 06 53 4C 41 53 48 30 01 53 00 52 D5 DD E1 DD 6E 01 DD 66 02 22 37 52 22 41
52 22 44 52 22 49 52 22 4F 52 01 1B 00 2A 49 40 AF ED 42 22 49 40 23 E5 EB 21
36 52 ED B0 E1 DD 75 01 DD 74 02 C3 2D 40 DA 00 00 F5 79 FE 30 28 04 F1 C3 00
00 CD 00 00 0E 08 CD 00 00 0E 2F F1 C3 00 00 02 02 00 52
```

DIRMAP/BAS

-----

This is an LBASIC program that will display a map of a diskette, showing which programs occupy which granules on the disk. It was contributed by Dave McLaughlin, and is also in our MNET database. There are some humorous comments randomly displayed by this program.

```
20 'Disk Mapping Program      07/07/81
40 'Public Domain by Dave  Mc Laughlin - not for sale
60 'for use only under LDOS - single or double density
80 'displays a map of diskette granules or ** Empty **
100 CLEAR20000
120 DEFINT A-Z
140 DIM X$(96,3) ' tracks,grans
160 CLS: PRINT @ 269,"Dave  Mc Laughlin's Disk Mapper"
180 PRINT @456,"Insert Disk in drive 1 and press enter"
200 PRINT@656,"LDOS DISKS ONLY!!!!!!!!!!"
220 FOR I=0 TO 96 : FOR J=1 TO 3
240 X$(I,J)="** Empty **"
260 NEXT J,I
280 PRINT @858,"NOW!"
300 DN$="" : DN$=INKEY$ : IF DN$="" THEN 300
320 IF DN$<>CHR$(13) THEN 300
340 OPEN"R",1,"DIR/SYS:1",32
360 '***** get track count / disk name *****
380 FIELD 1,32 AS F1$ 'track count
400 IF RND(10)=1 PRINT@852,"Wiping disk" : GOTO 440
420 PRINT@851,"Reading Directory"
440 GET 1,7 'track count differential
460 TD=ASC(MID$(F1$,13,1))
480 IF TD>100 THEN TD=TD-256
500 TC=35+TD
520 TX=0
540 FOR I=4 TO 6
560 GET 1,I
580 FOR J=1 TO 32
600 TX=TX+1
620 J1=ASC(MID$(F1$,J,1))
640 IF J1<>255 THEN 740
660 FOR J3=1 TO 3
```

```

680 J4=((I-4)*32+J-1)
700 X$(J4,J3)="*Locked Out"
720 NEXTJ3
740 IF TX=TC THEN 780
760 NEXTJ,I
780 FIELD 1,16 AS F1$,8 AS Y1$,8 AS F2$ 'DISK NAME
800 GET 1,7
820 DN$=Y1$
840 FIELD 1,32 AS Y1$
860 Y1=$
880 POKE 14304,0 'KEEP DISK ALIVE
900 '***** Granule Allocation Table *****
920 TX=$
940 FOR I=1 TO 3 'GAT
960 GET 1,I
980 FOR J = 1 TO 32
1000 TX=TX+1
1020 Y7=ASC(MID$(Y1$,J,1))
1040 IF X$(((I-1)*32+J-1),1)="*Locked Out" THEN 1180
1060 IF NOT Y7 AND 1 THEN Y1=Y1+1 GOTO 1100
1080 X$(((I-1)*32+J-1),1)=" 'NOT EMPTY
1100 IF NOT Y7 AND 2 THEN Y1=Y1+1 : GOTO 1140
1120 X$(((I-1)*32+J-1),2)=" 'NOT EMPTY
1140 IF NOT Y7 AND 4 THEN Y1=Y1+1 : GOTO 1180
1160 X$(((I-1)*32+J-1),3)=" 'NOT EMPTY
1180 POKE 14304,0
1200 IF TC=TX THEN 1=4 : J=33 'done
1220 NEXTJ:NEXTI
1240 '***** Process Directory entries *****
1260 FIELD 1,1 AS Y9$,4 AS F1$,8 AS P1$,3 AS P2$,4 AS F2$,2 AS EF$,10 AS EE$
1280 GET 1,25 'dir/sys entry
1300 IF ASC(MID$(EE$,2,1))=1 THEN ZZ=96 : GC=2 : GOTO 1340
1320 ZZ= 144      GC=3 'double density
1340 IF GC=3 THEN K1=INT((Y1*15/10)+.5) : G1=Y1 :GOTO 1380
1360 K1=INT((Y1*12.5/10)+.5) : G1=Y1
1380 FOR Y8=17 TO ZZ
1400 GET 1,Y8
1420 Y9=ASC(Y9$)
1440 Y7=Y9 AND 16
1460 IF Y7= 0 THEN 1920 'KILLED
1480 Y7=Y9 AND 128
1500 IF Y7 = 128 THEN 1920 'skip - this is fxde
1520 P1=INSTR(P1$," ") 'FIND BLANK
1540 IF P1=0 THEN P1=9
1560 IF P2$<>" " THEN P1$=LEFT$(P1$,P1-1)+"/"+P2$ELSE P1$=P1$
1580 POKE14304,0
1600 FOR J=1 TO 10 STEP 2 'extents
1620 T=ASC(MID$(EE$,J,1)):W=ASC(MID$(EE$,J+1,1))
1640 IF T=255 AND W=255 THEN 1920
1660 IF T<>254 GOSUB 2480 : GOTO 1900 'not fxde pointer
1680 L=W AND 224 'get leftmost 3 bits
1700 R=W AND 7 'get rightmost 3
1720 L=L/32
1740 R=(R+2)*8+L+1
1760 GET 1,R 'get fxde
1780 FOR K=1 TO 10 STEP2 'extents

```

```

1800 T=ASC(MID$(EE$,K,1)):W=ASC(MID$(EE$,K+1,1))
1820 IF T=255 AND W=255 THEN 1880
1840 IF T<>254 GOSUB 2480 : GOTO 1880
1860 GOTO 1680 'another fxde needed
1880 NEXTK
1900 NEXTJ
1920 NEXTY8
1940 CLOSE
1960 SP=0 : TC=TC-1
1980 SF=12 'screen roll factor
2000 '***** Display Results *****
2020 CLS 'print screen
2040 PRINT"Diskette "DN$"      Tracks "TC+1"      Free: "STR$(G1)"g "STR$(K1)"k
";:IF GC=2 PRINT" SDENS" ELSE PRINT" DDENS"
2060 FOR I=SP TO SP+SF 'track
2080 IF I>TC THEN 2180
2100 PRINTUSING "##          ";I;
2120 FORJ =1 TO GC 'gran
2140 PRINTUSING"%          %          ";X$(I,J);
2160 NEXTJ : PRINT : NEXTI
2180 UA=0 : DA=0 'arrows
2200 IF SP<TC-SF THEN UA=-1 : PRINT@960,CHR$(91)" = Roll up";
2220 IF SP>0 THEN DA=-1 : PRINT @976,CHR$(92)" = Roll down";
2240 PRINT@996,"<ENTER> = quit";
2260 IK$=""
2280 IK$=INKEY$ : IF IK$="" THEN 2280
2300 IF IK$=CHR$(91) AND UA THEN SP=SP+SF : GOTO 2020
2320 IF IK$=CHR$(10) AND DA THEN SP=SP-SF : GOTO 2020
2340 IF IK$=CHR$(13) THEN CLS : END
2360 GOT02260
2380 '***** Debug Tool *****
2400 FOR I=0 TO 35 : PRINT I, :FOR J=1 TO 3
2420 PRINTX$(I,J),
2440 NEXTJ : PRINT : NEXT I
2460 END
2480 '***** post extent *****
2500 G=W AND 31 'mask left 3 bits
2520 G=G+1 'actual # of grans
2540 IF W<32 THEN W=1: GOTO 2600
2560 IF W>63 THEN W=3: GOTO 2600
2580 W=2
2600 IF LEFT$(X$(T,W),1)<>"*" THEN 2640
2620 PRINT"Directory error in "PN$,T,W : CLOSE : END
2640 X$(T,W)=PN$
2660 G=G-1
2680 IF G=0 RETURN
2700 W=W+1
2720 IF W=4 AND GC=3 THEN W=1 : T=T+1
2740 IF W=3 AND GC=2 THEN W=1 : T=T+1
2760 GOTO 2600

```

MiniDOS/FLT

-----

This keyboard filter is for Model I LDOS Version 5.\$x users only, as Version 5.1.x comes with a MiniDOS filter. This program was converted from the Model III version for Model I use by Roy Soltoff of LDOS Support.

Once the filter is applied (FILTER \*KI MINIDOS), pressing the <CLEAR><SHIFT> and appropriate key will cause the following:

<D> - enters the system DEBUGger (if previously activated).

<K> - prompts for a filespec and kills the specified file.

<Q> - prompts for a drivespec and displays all visible files.

<R> - repeats the last entered DOS command.

<T> - sends a top of form to the lineprinter.

Use BINHEX/BAS to convert this code to a file called MINIDOS/FLT. If a KSM filter is to be used along with MiniDOS, it must be applied before the MiniDOS/FLT. Additionally, MiniDOS/FLT must be the last filter applied to \*KI if the <R> command is to function.

```
05 06 4D 49 4E 49 44 4F 1F 32 43 6F 70 79 72 69 67 68 74 20 28 43 29 20 31 39
38 31 20 62 79 20 4C 6F 67 69 63 61 6C 20 53 79 73 74 65 6D 73 20 49 6E 63 6F
72 70 6F 72 61 74 65 64 01 02 00 52 D5 21 7B 52 CD 67 44 21 1F 44 CB 6E C2 73
52 CB EB DD E1 ED 5B 16 40 3A 0F 43 CB 6F 28 04 ED 5B BE 43 ED 53 FB 52 01 BC
00 2A 49 40 AF ED 42 22 49 40 23 E5 11 86 00 19 22 29 53 22 45 53 E1 E5 11 A4
00 19 22 84 53 E1 E5 11 B9 00 19 22 A9 53 D1 D5 21 FA 52 ED B0 F3 DD 36 00 01
E1 3A 0F 43 CB 6F 20 08 DD 75 01 DD 74 02 18 03 22 BE 43 FB C3 2D 40 21 E0 52
CD 67 44 18 F5 4D 44 4F 53 20 2D 20 4C 44 4F 53 20 4D 69 6E 69 2D 44 4F 53
20 4B 65 79 62 6F 61 72 64 20 46 69 6C 74 65 72 20 2D 20 56 65 72 73 69 6F 6E
20 35 2E 30 0A 43 6F 70 79 72 69 67 68 74 20 28 43 29 20 31 39 38 31 20 4C 6F
67 69 63 61 6C 20 53 79 73 74 65 6D 73 20 49 6E 63 6F 72 70 6F 72 61 74 65 64
0A 0D 4D 69 6E 69 44 4F 53 20 69 73 20 61 6C 72 65 61 64 79 20 61 63 74 69 76
65 0D CD 00 00 F0 D8 F5 01 B8 00 53 FE E4 28 12 FE F1 28 3C FE EB 28 1C FE F4
28 10 FE F2 28 51 F1 C9 F1 3A 0F 43 E6 80 C8 C3 0F 40 F1 3E 0C CD 3B 00 AF C9
CD 80 53 21 81 44 06 1E CD 40 00 11 80 44 CD 1C 44 3E 13 C0 CD 24 44 C0 CD 2C
44 C9 CD 80 53 21 80 44 06 05 CD 40 00 7E 23 FE 0D 20 02 3E 30 D6 30 4F 06 00
CD B8 44 3E 20 C0 C3 63 44 F1 21 10 00 39 5E 23 56 21 53 4E ED 52 28 02 AF C9
D1 D1 DD E1 E1 C1 E1 E1 C9 E3 D5 C5 11 9E 53 D5 F5 3E 0D CD 33 00 3E 7B CD
33 00 F1 E6 7F CD 33 00 3E 7D CD 33 00 E9 C1 D1 E1 28 05 F6 C0 CD 09 44 CD B3
53 F1 3E 0E CD 33 00 AF C9 3E 83 EF 02 02 00 52
```

TRS-80 and TRSDOS are trademarks of Tandy Corp. The LDOS QUARTERLY is copyrighted in its entirety. No material contained herein may be duplicated for any purpose without the written permission of Logical Systems, Inc.