

# THE MISOSYS QUARTERLY

In this issue:

- ☛ FILLow: Fill low memory, by J. Armstrong
- ☛ Internal HD for 4P, by J. Cerul
- ☛ Roll your own on the XLR8er, by J. F. Slinkman
- ☛ Profile 4+ printer codes, by D. Krebs
- ☛ Boot LDOS 5.3 from HD Model 4, by A. Rubin
- ☛ Memory: How much and why, by K. Strickler
- ☛ 300 Dots: An update, by G. Shanafelt
- ☛ Lair of the Dragon: Hint sheets, by D. Goben



**MISOSYS will be closed  
August 11th through  
August 17th**



## PRICE LIST effective June 1, 1991

### TRS-80 Software

Product Nomenclature	Mod III	Mod 4	Price S&H
AFM: Auto File Manager data base	P-50-310	n/a	\$49.95 D
BackRest for hard drives	P-12-244	P-12-244	\$34.95
BASIC/S Compiler System	P-20-010	n/a	\$29.95 B
BSORT / BSORT4	L-32-200	L-32-210	\$14.95
CON80Z / PRO-CON80Z.	M-30-033	M-31-033	\$19.95
diskDISK / LS-diskDISK	L-35-211	L-35-212	\$29.95
DISK NOTES from TMQ (per issue)			\$10.00
DoubleDuty		M-02-231	\$49.95
DSM51 / DSM4	L-35-204	L-35-205	\$49.95
DSMBLR / PRO-DUCE	M-30-053	M-31-053	\$29.95
EDAS / PRO-CREATE	M-20-082	M-21-082	\$44.95 D
EnhComp / PRO-EnhComp	M-20-072	M-21-072	\$59.95 D
Filters: Combined I & II	L-32-053	n/a	\$19.95
GO:Maintenance	n/a	M-33-100	\$49.95 B
GO:System Enhancement	n/a	M-33-200	\$49.95 B
GO:Utility	n/a	M-33-300	\$49.95 B
Hardware Interface Kit	n/a	M-12-110	\$24.95
HartFORTH/PRO-HartFORTH	M-20-071	M-21-071	\$49.95 B
LDOS 5.1.4 User Manual	L-40-020	n/a	\$15.00 D
LDOS 5.3 Mod3 Upgrade Kit	M-10-033	same	\$34.95
LED / LS-LED	L-30-020	L-30-021	\$19.95
LB Data Manager-M4 (Ver 2.1)	n/a	M-50-510	\$99.00 D
LS-DOS 6.3.1 Upgrade Kit - M4	n/a	M-11-043	\$39.95
LS-DOS 6.3.1 Diskette - M4	n/a	M-11-243	\$15.00
LS-DOS 6.3.1 Upgrade kit - M2/12/16		M-11-002	\$39.95 B
LS-Host/Term	n/a	L-35-281	\$39.95
LS-UTILITY	n/a	L-32-150	\$24.95
MC / PRO-MC	M-20-064	M-21-064	\$79.95 D
Mister ED	n/a	M-51-028	\$39.95 B
MRAS / PRO-MRAS	M-20-083	M-21-083	\$59.95 D
PowerDot (Epson or Tandy)	P-32-217	n/a	\$19.95
PowerDraw	P-32-220	n/a	\$19.95
PowerDriver Plus (Epson).	P-50-200	P-50-200	\$17.95
PowerMail Plus	P-50-003	P-50-004	\$39.95 D
PowerMail Plus TextMerge	P-50-100	P-50-100	\$15.00
PowerScript	P-50-142	P-50-142	\$24.95
PRO-WAM	n/a	M-51-025	\$74.95 D
PRO-WAM Toolkit	n/a	M-51-225	\$29.95
Programmer's Guide DOS 6.	n/a	M-60-060	\$20.00 B
QuizMaster	L-51-500	n/a	\$19.95
RATFOR-M4		M-21-073	\$59.95 D
RSHARD - R/S HD driver	M-12-013	same	\$29.95
ST80-III	P-35-300	n/a	\$39.95
SuperUtilityPlus	P-32-132	P-32-104	\$44.95 D
SuperUtilityPlus CMD file diskette	P-32-832	P-32-804	\$20.00
Supreme HD Driver (PowerSoft-RS)	P-12-113	P-12-113	\$34.95
TBA / LS-TBA	L-21-010	L-21-011	\$19.95 D
THE SOURCE 3-Volume Set	n/a	L-60-020	\$40.00 D
Toolbox/Toolbelt	P-32-203	P-32-245	\$24.95
UNREL-T80	same	M-30-054	\$29.95
UTILITY-I	L-32-070	n/a	\$19.95

### MSDOS Software

LB Data Manager 2.1	M-86-510	\$99.00 D
DED-86 [Disk/Memory sector editor]	M-86-020	\$29.95 D
RATFOR-86	M-86-073	\$59.95 D
HartFORTH-86	M-86-071	\$59.95 D
SAID-86 [Text Editor]	M-86-040	\$29.95
FM-86 (File Manager)	L-86-050	\$29.95

### TRS-80 Game Programs

Bounceoids (M3)	M-55-GCB	\$14.95
Crazy Painter (M3)	M-55-GCP	\$14.95
Frogger (M3)	M-55-GCF	\$14.95
Kim Watt's Hits (M3)	P-55-GKW	\$9.95
Lair of the Dragon (M3/M4)	M-55-021	\$19.95
Lance Miklus' Hits (M3)	P-55-GLM	\$19.95
Leo Cristopherson's (M3)	P-55-GLC	\$14.95
Scarfman (M3)	M-55-GCS	\$14.95
Space Castle (M3)	M-55-GCC	\$14.95
The Gobbling Box (M3/M4)	M-55-020	\$19.95

### MSDOS Game Programs

Lair of the Dragon	M-86-021	\$19.95
--------------------	----------	---------

### Hardware

TeleTrends TT512P modem (M4P)	H-4P-512	\$74.95	E
XLR8er e/w 256K RAM (M4)	R-MB-004	\$182.00	F
Floppy drives (5.25" 360K 1/2 ht)	H-FD-360	\$75.00	D
Floppy drives (3.5" 720K 1/2 ht)	H-FD-720	\$85.00	B
Floppy Drive Case (2-1/2 ht drives)	H-FD-2SV	\$60.00	F
Hard drive kit e/w clock, 20Meg M3/M4	H-HD-020	\$475.00	?
Hard drive kit e/w clock, 40Meg M3/M4	H-HD-040	\$595.00	?
Hard drive joystick port option	H-HD-JSO	\$20.00	
Hard drive: Kalok KL320	R-HD-020	\$200.00	F
Hard drive: Seagate ST251-1	R-HD-040	\$320.00	F
Hard drive: Seagate ST157A (16B IDE)	R-HD-A40	\$320.00	F
Hard drive: Seagate ST-157N (SCSI)	R-HD-S40	\$350.00	F
Cable: dual floppy extender	H-FD-2EX	\$18.00	
Cable: 4Ft floppy (1 34EDC each end)	H-FD-C04	\$12.50	
Cable: 4Ft M3/M4 printer	H-RC-PM4	\$20.00	
Cable: 4Ft Radio Shack hard drive	H-HD-CT4	\$20.00	
Cable: 4Ft MISOSYS hard drive	H-HD-C04	\$22.50	
Cable: 26-1069 internal floppy	H-FD-2NG	\$20.00	
Cable: 26-1069A/26-1080 internal floppy	H-FD-2GA	\$20.00	
Cable: 26-1080/A internal floppy	H-FD-24P	\$20.00	
Cable: drive power Y	H-HD-CPY	\$5.00	
Cable: XT hard drive set	H-HD-CXT	\$5.00	
Cable: Custom IDC ribbon (M3/M4/M2)	?-??-???	varies	
Standby Power System: 200VA	R-PS-200	\$199.00	?
Standby Power System: 450VA	R-PS-450	\$399.00	?
HD Controller: Adaptec 4010A	H-HD-CA4	\$75.00	D
HD Controller: Xebec S1421A	H-HD-CX2	\$75.00	D
T80 to SCSI host adaptor	H-HD-MHA	\$75.00	D
ZOFAX 96/24 Fax/Modem (PC XT/AT)	R-Z1-FAX	\$225.00	F
Infochip Systems Expanz! (PC)	R-IC-EXP	\$179.10	F
DJ10 Tape Backup (PC)	R-TD-D10	\$275.00	D
AB10 Tape Adaptor (PC)	R-TD-A10	\$75.00	
KE10 External tape adaptor/case (PC)	R-TD-K10	\$110.00	D

### The Fine Print

Freight codes: A = \$3.50; B = \$4.00; C = \$4.50; D = \$5.00; E = \$5.50; F = \$6.00; G = \$8.50; H = \$12.00; ? = varies; All unmarked are \$3.00 each; Canada/Mexico add \$1 per order; Foreign use US rates times 3 for air shipment. Virginia residents add 4.5% sales tax. We accept MasterCard and VISA; Checks must be drawn on a US bank. COD's are cash, money order, or certified check; add \$4 for COD.

**MISOSYS, Inc.**

**P.O. Box 239**

**Sterling, VA 22170-0239**

**703-450-4181; Orders only: 800-MISOSYS (800-647-6797)**



The MISOSYS Quarterly is a publication of MISOSYS, Inc., PO Box 239, Sterling, VA 22170-0239, 703-450-4181.

The MISOSYS Quarterly is published four times a year in February, May, August, and November. Published by Roy Soltoff

### THE MISOSYS QUARTERLY subscription rate information

Each issue of TMQ has information on MISOSYS products, programs and utilities, patches, significant messages from our CompuServe forum, and articles on programming. Not only that, TMQ will keep you up to date with information, news, and announcements concerning our entire product line and related machine environments. Subscription cost varies by rate zone as follows:

A = \$25; United States via 3rd class bulk mail  
B = \$30; Canada, Mexico, via 1st Class  
C = \$32; Colombia, Venezuela, Central America via AO Air  
D = \$35; South America, Europe, & North Africa via AO Air  
E = \$40; Asia, Australia, Africa, Middle East via AO Air

### TMQ Toolbox

The MISOSYS Quarterly is published using the following facilities:

The hardware used for development of the "camera ready" copy consists of an AST Premium/386 computer (20 MHz) equipped with 9 Megabytes of RAM, a Seagate ST4096 80-Meg HD, ST251 40M, Expanz! card; a CMS DJ10 tape backup device, a NEC Multisync II color monitor driven by a Video Seven VGA card, an AST TurboScan scanner (Microtek MS300), and a NEC LC-890 PostScript laser printer.

Text is developed, edited, spell-checked, and draft formatted using Microsoft WINWORD Version 1.0; Submissions on paper and letters are scanned and converted to text using Read-Right optical character recognition software by OCR Systems. Final page composition is developed using PageMaker 4.0 by Aldus. Cover art and clip art comes from CLIPPER, a product of Dynamic Graphics.

## Table of Contents

### The Blurb

Points to Ponder	2
TMQ Schedule	4
Out of print TMQ's	4
Vacation reminder	5
TMQ advertising	5
PD Software Librarian	5
MISOSYS Forum	5
DISK NOTES 5.4	6
Infochip's Expanz! data compression	6
Onesies and Twosies	7
Ribbon Cable Assemblies	7

### Letters to MISOSYS

Response to TMQ V.iii	8
Metrics...	13
Lair of the Dragon Maps	13
The saga of DoubleDuty	14
LB Data Manager	16
LS-DOS 6.3 Support	17
Model I display problem	17
MC/QuickC & void	18
TRSCROSS and SuperScripts	18
Help TRSDOS - IBM	19
TRSCROSS & BASIC translation	19
RS Hard Drive Cable	19
SUPERSCRIPSIT PRT DRIVER	19

### Inside The MISOSYS Quarterly

FILLOW: Fill low memory, by Jonathan Armstrong	20
Internal HD for 4P, by John Cerul	21
How to "roll your own" on the XLR8er, by J. F. R. Slinkman	23
Profile 4 Plus printer codes, by Dave Krebs	28
BOOT LDOS 5.3 directly from a hard drive Model 4, by Adam Rubin	29
Memory: How much and why, by Ken Strickler	34
300 Dots: An update, by Gary Shanafelt	39
Lair of the Dragon : Hint Sheets, by David Goblen	40

### List of Advertisors

J.F.R. Slinkman	43
MISOSYS, Inc.	6,38,44-48,IFC,IRC,RC
Pacific Computer Exchange	19
Roy T. Beck	28
TRSTimes magazine	43

### List of Patches in this Issue

XLBOOTB2/FIX, XLSYS0B2/FIX	9
XLBOOTC1/FIX	10

## Points to Ponder

In this issue of *The MISOSYS Quarterly*, I announce what may be the biggest up-graded product in the history of MISOSYS. The two DOS products - LS-DOS for the Model 4 and II, and LDOS for the Model III - used to share that spot in the limelight, but with the release of LB Data Manager version 2.1, that product takes the cake.

LB has been a huge project - one that begins to tax the ability to manage. But with the proper tools, the suggested changes and enhancements posed by its users, excellent and creative C programming design and implementation by Richard Deglin - who did the lion's share of bringing Little Brother version 1.0.0 to LB Data Manager version 2.1.0, and some final programming efforts by yours truly, the product is released.

Before I expand on the features which have been implemented, let me shed some light on how LBDM 2.1 came together. LB is written predominantly in the C language; a smidgeon of code is in assembler - that used to perform some low-level OS interfacing. Both the TRS-80 Model 4 and MS-DOS versions of LB are compiled into executable program modules on my 386 MSDOS machine. The files which collectively make up the data base product during its generation encompass 3.2 megabytes of space; the documentation files are another 0.6 megabytes. I won't even begin to count the number of files.

In order to smooth the program generation process, all of the commands required to go from the source code level to reach an executable program file are managed by NMAKE, a make utility bundled with Microsoft's C compiler version 6.0, which was used to compile the MS-DOS version of LBDM. The TRS-80 version of LBDM was compiled with a version of our MC compiler which runs under MSDOS but still generates Z80 code; this special ver-

sion of MC is considered a *cross-compiler* because it generates code for a system different from the one the compiler is running under. The conversion from Z80 source code files output by MC to binary files is handled by the Uniware cross assembler from Software Development Systems. Finally, some in-house tools developed by Richard convert the binary files to the load module types needed for the TRS-80. NMAKE is also used to manage the TRS-80 data base development process. A make utility is absolutely essential to the process of putting together a complex program package such as LB. Note that TMQ V.ii presented a simple make utility for MC.

The LB User Manual was no small effort, either. The 600K of source files were processed with Microsoft's Word for Windows. The most beneficial aspect of WINWORD was its tremendous ease in putting together the index and table of contents. Although the 200-page LB User Manual was composed of about a dozen files, WINWORD generated the nine-page index in one pass; all of the individual files were linked together.

It hasn't been easy getting all of these tools up to date. My primary system has seen a great deal of activity over the past six months. Starting with the addition of the *Expanz!* board to increase my usable disk capacity by 50%, I have installed an inordinate quantity of software upgrades. I moved up to Windows 3.0. I installed Microsoft's MSC 6.0, its maintenance release 6.0a, as well as the MSC HELP generator and the toolkit. I also installed WINWORD, the new Excel 3.0, the new ReadRight OCR version 3.0, Imagestar for Windows 3 version 1.0, as well as Pagemaker 4.0. I even upgraded to Microsoft Word 5.5 from 5.0 which was a big

mistake! Although I had intended to do most word processing under WINWORD, I still wanted to maintain character word for certain jobs. The primary change introduced in 5.5 was a total change of the user interface. A user interface for a complicated program is learned over the years. With the change to 5.5, I threw out three years of education using WORD. On the hardware side, I added the Zofax 96/24 fax/modem (hey, I sell it) and associated BITCOM and BITFAX software. Another 4 megabytes of memory were also acquired - although I did not see much of a change in performance. Phew! So I'm now left with a half-dozen or so new packages with a shelf of new manuals to learn. Incidentally, I am using Pagemaker 4.0 to prepare this issue of *The MISOSYS Quarterly*; a few of its features can be recognized: the type rotation found on the cover and chapter introductions, and the maze data tables in Goben's Lair of the Dragon hint sheets. Incidentally, you might drop me a line to express your choice of the italicized vertical headline used in *Letters to MISOSYS*, or the normal font vertical headline used in *Inside TMQ*. But back to LB. One final addition was the acquisition of Knowledge Dynamics Corporation's "INSTALL" automated software installation utility which was used to enable the easy installation of LB under MSDOS.

Now then, many LB users have been chomping at the bit for the new release - I hope so, because LB has to be the product to carry MISOSYS forward. Here's the kinds of new features which version 2.1 provides:

In the way of utility adjuncts, The Little Brother Maintenance Utility (LBM1), previously a separate product, has been upgraded for version 2 definitions and

## The Blurb by Roy Soltoff



renamed LBREDEF (for *redefining* a data base structure). A help command has been added to its command bar (the command actually uses the help module displayed by LBDM's File Definition mode). LBREDEF is now bundled with the data manager.

The LBMANAGE utility has been added; this allows you to duplicate a database structure, copy or move records from one to another (identical structure), delete records from a database, or fix a broken deleted record chain (replaces the FIXDEL utility). The functions copy, move, and purge require an index file. LBMANAGE is quite useful for splitting a large data base into two or more subordinate data bases or for combining two smaller into a larger. The duplicate function allows you to easily initiate a new data base of identical structure.

The help files have been updated to reflect all of the changes introduced into version 2. The LBHELP/HLB module has also been altered to include help screens for LBMANAGE and LBREDEF.

The database definition module enhances the use of *Protected Fields* without PASS-WORD entry by allowing the protection to be display/print/edit, or just edit. In this way, a "master" user can protect a field from user alteration but that field could be displayable or printable. The enhancement is implemented by an additional choice: Y = protect all; E = protect against edit; N = not protected.

In terms of the data base field types, a "date-of-last-update" field has been added. This field is automatically maintained by LB if it is defined for the data base record. The field cannot be altered by the user; it contains the date when the record was last edited. The format used is "YYYY/MM/DD"; the field definition character is a "\ " (reverse slash). Here's one more reason to keep your DOS system date accurate.

All fields can now be optioned to be a *must-fill* field, where data must be entered in the field or the program refuses to store

the record (i.e. F3 does not save; <ESC> aborts any changes/additions). This "must-fill" facility is another column entry during data base definition or re-definition.

Most of the changes introduced are associated with report generation - that was the area where users concentrated their requests. For printouts, LB 2.1 introduces *printer initialization control sequences*. These add the capability of sending a control sequence of codes to a printer or file prior to and subsequent to a printout. The (de)initialization control code sequences are part of the PRT PARAMETERS setup for each format. Such codes allow you to initiate your *smart* printer to a particular font or other arrangement - something which usually has to be done once.

LB provides for multiple reports without having to go back to the main menu. When a print is complete, LB now returns to the print sub-menu. When the ESCAPE (QUIT) from the sub-menu is invoked, it returns to the "Enter print format(1-10)" prompt. An ESCAPE from that prompt exits the print modules.

For output flexibility, reports are capable of being directed to a <P>rinter, the <S>creen, or to a disk <F>ile. Screen prints are paged. <F>ile prints do not suppress printer (de)initialization strings so the file could be subsequently printed as you would expect. File reports are written to the TEMP path with the filename of your choice and an extension of RPT. An existing report file can be overwritten or appended to. <S>creen printouts suppress the (de)initialization strings.

For added index file support, the print module function now accepts a print range when printing with an index, just like record range when not using an index. This allows you to use other than the first/last index positions for the printing. You can select First through mth, nth through last, or mth through nth. So when your printer jams up and you have to abort the printout, you can restart the report at some page boundary.

There's a print output option that can suppress printing of records, but still print footer subtotals. This is useful for summaries on screen when the printout is directed to the screen with screen pause.

To extract more information from your data, enhancements add special footer printing values for the "number of records" printed, per page, per ctrl-break, and total. This logically uses ^#p^ for page count, ^#c^ for ctrl-break, and ^#t^ for total count. We've added printing of field averages (for D, F, or C fields). This again logically uses ^as?^ for average of a subtotal, ^ac?^ for average of a ctrl-break subtotal, or ^at?^ for average of a total; "?" specifies the field #.

Listening to the folks who wanted improved form letter capabilities, the following Header/Text enhancement coupled with the print-to-file enhancement makes LB very versatile for generating a mail/merge data file for use with existing word processors such as Allwrite or SuperScript. This enhancement adds an option to suppress the header on all but the first page. The header can then be used to generate the mail/merge header record once while the data records are then generated as the merge data. The procedure is fully documented in the LB User Manual.

At the main menu level, the function of DOS access (SHELL) has been introduced as a menu item. Using the shell facility, you can temporarily escape to the DOS command level for such things as copying files, formatting disks, displaying directories, etc. Use it to obtain the names of your JOB files.

The Update and Delete module was beefed up to include a *Replace* sub-command. This supports the global SEARCH and REPLACE of a field either with or without an index file, with wildcard replacements (i.e. replacement wildcard character maintains the positional source character). A "?" matches anything in the match string while a "\*" substitutes the original destination string's corresponding character instead of the "?" in the

replacement string. Note that the match string is case specific for added security! When using the *records find* option, you are now able to view the field definitions to determine what field to search for.

Navigation has been improved within the ADD/EDIT/UPDATE screens with the ability to navigate forward or reverse using cursor control keys <UPARROW> and <DOWNARROW>.

While adding records, you can enter the same field data from the previously entered record during ADD, using a single keystroke (<CTRL-D> for duplicate).

In the Sort/Select module, a *record number range selection* can be requested. SELECT adds select by record number range as a "field" criteria. Specify a "0" for the field number prompt to be queried for starting and ending record number. The "Ignore case" extraneous query on UPPER case literal fields (B and U) is also suppressed.

All data files of version 1 are upward compatible with version 2 with the exception of the MSDOS database.PFL file. Path strings under MSDOS have been increased to 64 characters in length. A conversion utility is provided for MSDOS users to upgrade their existing PFL files. If you have been a TRS-80 user running LB 1.0.0, you must create a path file for your database(s) using menu option 14 prior to being able to access your existing data files; those who have been using the LB 1.x.x beta version, will require no changes.

LB 1.0 placed three data items as the last six bytes of the data file. These were total record space allocated, records in use, and record length. The data was redundant with the definition file and was originally used by LSI to guard against a user using a backup of the /DEF file which differed from the /DAT file. Of course, when LB found a mismatch, it didn't let you access the data. That problem caused great difficulties with some folks. These six bytes of data are no longer used in version 2. Thus,

once you use version 2 to add/delete data, you will no longer be able to use version 1.

All in all, I feel that the LB 2.1 is a fantastic release. For existing LB 1.0 users, I have tried to make the upgrade cost most attractive and easy to accomplish. The upgrade fee is \$40 plus shipping (\$5 in U.S.; \$6 in Canada; \$7 in Europe; \$9 in Asia, Pacific Rim, and Australia) and all you need to remit is the fee plus the original Table of Contents page from the version 1.0 User Manual - just rip it out and send it in. You will be entitled to a totally new package - complete with the 200+ page *User Manual* which has been re-written. It doesn't matter which version you had - TRS-80 or MS-DOS - you can request either version for the 2.1 release. Thus, if you had the TRS-80 version and have moved on to MS-DOS, you can obtain the MS-DOS version for the upgrade cost. And don't forget that a data base structure and data running under the TRS-80 version is directly usable under the MS-DOS version. Just TRSCROSS the files using binary transfer or download the database files via a null modem transfer and use them under the MSDOS version. I'll be looking for your upgrade request.

If you are not already an LB Data Manager user, but have any other commercial data base product - no matter how primitive, you can trade in your old package and deduct \$49.50 from the cost of acquiring a new LB 2.1.0 package. Just send in your old data base package (disk and manual) with \$49.50 + \$5S&H to obtain a complete LB 2.1.0. It doesn't matter whether what machine or DOS your old data manager was designed for; you can pick either the TRS-80 Model 4 version or MS-DOS version of LB.

As a postscript, if you would like to make public your existing data base structure(s), for other LB users to utilize, just duplicate your data base (using LBMANAGE), and send the blank data base file, DEF file, and screen/prINTER files to me. I will bundle them into an archive and make them available.

## TMQ Schedule

I target mailing *THE MISOSYS QUARTERLY* the last week of the respective month as follows: Spring issue in February, Summer issue in May, Fall issue in August, and Winter issue in November.

Note that your mailing label usually has the expiration date of your subscription. For instance, those with "91/08" complete their subscription with this issue. If you want to save me the cost of mailing a renewal notice, send in your renewal fee quickly. I usually wait about a month after TMQ is mailed before sending out renewal notices. Please note that there is no longer a US first class rate - US is "A" rate only!

As I write this sentence, it's Monday the 20th of May. This issue will be off to the printers early next week. It looks like this is almost on track.

## Out of print TMQ's

I have discontinued the practice of copier reprints for out of print issues (Volume I and Volume II); they are no longer available. The price for back issues still in print is \$4 + S&H (minimum order of \$15). S&H for a single issue is \$2.75 in the U.S. and CANADA; \$5.50 zone D; zone E is \$6.50. S&H for four issues is \$5 (US), \$6 (CAN), \$14 (ZoneD), \$20 (ZoneE). Here's a synopsis of past issues:

**IIIi** Reading NEWDOS/80 disks; An LB archival utility; Popup Application Window; XMODEM in C; Getting into computer math, part I; TMQ Volume I index.

**IIIii** Getting into computer math Part



2; Writing interactive RATFOR/FORTRAN programs; PRO-EnhComp: a review; Desktop publishing and the Model 4; A better TERM/APP; adding floppy drives; and a new XLR8er interface.

**III.iii** The CRC program; PG: a page display program; Locating high memory routines; FIXMA3; Jumbo tape backup for PC clones; New style for TMQ using Pagemaker; and an Index to Volume II.

**III.iv** Checking for a file from Model 4 BASIC; Surviving the Hard Disk crash; An "interview" with Niklaus Wirth; Keep your printer clean and oiled; On-line HELP with PRO-WAM; MISOSYS announces availability of Hard Drives; Logic in the C language.

**IV.i** Cataloging files with a word processor; Page display PRO-WAM application; File undating with FUNDATE; Array load routine for BASIC; XLR8er and the GT-180 graphics board.

**IV.ii** Printing from BASIC without cutting words; LOAD100 for Model 100; Generating date/time stamp; Favorite recipes; Some BASIC routines.

**IV.iii** Fast in-memory sort using XLR8er RAM; Using XLR8er RAM as graphics video RAM; Upgrade your 4P with external floppy drives; Doubling of files solved; SuperScript document file format; FELSWOOP PRO-WAM export utility.

**IV.iv** Five Twelve K: A better way; Multi-Command; Touch/c/c; Fixes for LS-DOS 6.3.1; DoubleDuty Version 2.6.0 released.

**V.i** 300 Dots on the TRS-80; Tandy 16/6000 Hard Disk Drives; NXWAM PRO-WAM application; A review of M.A.D.'s XROM; a MIDI interface for your TRS-80.

**V.ii** Image processing on the TRS-80 Model 4; A MAKE utility for MC; New XLR8er patches for LS-DOS 6.3.1; FORTH: A language for every application.

**V.iii** It's rude not to interrupt; A Model 4 mouse driver; Profile 4+ to filepro 16/dBASE III; and a complete map to *Lair of the Dragon*.

## Vacation reminder

MISOSYS closes up for a week during the summer. This year, Brenda's parents will be taking a long-awaited cruise to Alaska for two weeks during late July to early August. Brenda will be down in Miami during that time taking care of her dad's print shop. It is not known as I write this whether Stacey, Stefanie, and Benjamin will stay here with me or go to Miami with Brenda. So for now, I'll have to take a guess at the week I'll take off and close up MISOSYS. For now, the best guess is the week of August 11th through the 17th. So if you call and get the answering machine, you'll know why.

## TMQ advertising

If you are interested in reaching a dedicated TRS-80 audience, consider *THE MISOSYS QUARTERLY*. If you have a TRS-80 Model III or 4 related product to sell, you can reach these buyers by placing your advertisement in our publication. Current space rates are as follows (reduced from previous rates):

Full page	\$100
Half page	\$60
Quarter page	\$35
Ninth page	\$15

I compose the ninth-page ad layout so you have no artwork charge. Just submit your text. Ads for our inside covers are printed in the same color as the cover (TMQ

alternates between PMS colors: green 354, purple 266, blue 293, and red 199). If you would like to place your ad in *THE MISOSYS QUARTERLY*, send it in.

## PD Software Librarian

Vic McClung has volunteered to be the librarian for the collection of TRS-80 public domain diskettes. Henceforth all requests and contributions be directed directly to him at:

Vic McClung  
914 Crescent  
Sikeston, MO 63801  
USA

Note that if you upload a "public domain" file to our CompuServe forum [PCS-49], and want it to receive general distribution, please also mail a copy on disk to Vic. There is no legal provision for downloading files from Compuserve and re-distributing them, unless you were the uploader. Some of our readers who do not have access to our forum have an interest in those submissions. So if you want to help out the most numbers of fellow users, don't limit your submissions to just one source.

## MISOSYS Forum

MISOSYS sponsors a forum on CompuServe. You can reach many "experts" on TRS-80 and MS-DOS subjects by dialing in. The forum is reached via GO PCS49, or GO LDOS. If you have any questions concerning access, get on and leave a message to SYSOP. Joe Kyle-DiPietropaolo will get to you. Please don't call me here at MISOSYS because I cannot answer any questions as to its operation.

The forum contains a great deal of programs which you can download, as well as enter into the lively discussions which thread through the message system. If you do programming on a PC, the forum also contains the listings from *Programmer's Journal*. If you want to direct a message to me, my user ID is 70140,310. Post a message in private if you don't want it "broadcast"; some folks even send me orders via a PRIVATE message.

Make note of a recent upload: TRS80.ZIP. This package is a Model III emulator for fast PCs. It deserves a look.

## DISK NOTES 5.4

Each issue of *THE MISOSYS QUARTERLY* usually contains program listings, patch listings, and other references to files we have placed onto a disk. DISK NOTES 5.4 corresponds to this issue of TMQ. If you want to obtain all of the patches and all of the listings, you may conveniently purchase a copy DISK NOTES priced at \$10 Plus S&H. The S&H charges are \$2 for US, Canada, and Mexico, \$3 elsewhere.

## Infochip's Expanz! data compression

Here's a rundown on the Expanz! board for PC compatibles. I have had this board installed in my AST 386 machine for over a six months now. The board is based on a dedicated data compression/decompression chip, the IC-105; both the chip and the board are products of Infochip Systems. Running at a maximum speed of 40 Mhz, the chip can accept input data for

compression at up to 2 Mbytes/s; decompression is rated at 5 Mbytes/s. The underlying function of the board is to increase effective disk capacity transparently by compressing files as they are written to the disk drive and subsequently decompressing the files as they are read from the drive. The operation is transparent because once a drive has been *expanzed*, the board and its attendant interfacing software operate without user intervention.

Most compression schemes operate on data sequentially; that is, any attempt to provide compression for a data file which is accessed randomly would prove fruitless since there is no way to predict the actual location of a data segment in the compressed file. On the other hand, Infochip Systems appears to have accomplished the miracle of totally accessible compressed data on a random accessed basis.

Adding an Expanz! card to your system is relatively painless. You are cautioned to make backups of your files on the drive partitions you are going to compress. **Heed this warning.** The procedure to install an Expanz! card is straightforward. With backups in hand, you de-install any expanded memory managers. Then you have to run CHKDSK on the drive partitions you wish to compress to ensure that no cross links are found. You then run the install program. This process creates for you a SAFETY disk and an ICONVERT disk; Infochip Systems even provides pre-printed labels to apply to your disks. The SAFETY disk is used if you ever develop a problem with the Expanz! card; it emulates the decompression in software to provide a means for data recovery. The ICONVERT disk is used when you want to compress a disk partition; you can also run ICONVERT from your hard disk.

If you are going to compress your boot partition (typically your C: drive), you must ensure that you are using at least version 1.1 of the driver software.

The Expanz! card driver took up 44384 bytes of low memory - memory in the

640K region. It's memory needs are based on the number of DOS buffers specified in your CONFIG.SYS file multiplied by the largest sector size in use on your drive. I couldn't afford to lose that much space in addition to that taken by MS-DOS 3.3, DOS buffers, mouse driver, and scanner driver. I solved my memory problem by acquiring QEMM386 5.1 release. This memory manager provides the ability for 386 systems to load device drivers, buffers, and a few other odds and ends into the high memory space between 640K and 1,024K. I heartily recommend Quarterdeck's Expanded Memory Manager 386 (QEMM386) to any 386 owner.

Among the three partitions of my 80 megabyte drive, the data compression afforded by the Expanz! card turned the drive into almost a 150 megabyte drive. This has provided probably a year or two delay in having to acquire a much larger capacity hard drive.

For those without a 386 machine - or the ability for some 286 machines to load drivers into high memory (certain 286 chipsets can do just that with appropriate software like QEMM 386), you would need to make do with about 44K less low memory to consider an installation of the Expanz! board. That's about the only drawback I've seen. The board and its software have performed flawlessly for me since its installation over six months ago. In short, if you are running out of disk capacity, and are constantly shuffling files around between floppies and hard drive to make runtime room, and have been put off by the cost of upgrading to a larger disk drive, it's time to consider an Expanz! board. The board normally retails for \$199. I have a handful in stock that I want to place into the hands of some folks desperate for disk capacity. **You can pick up one while my stock lasts for just \$125 plus \$5 S&H**



## Onesies and Twosies

I have dredged up the following items from our archives. All items are new and unused except where indicated. Numbers in parentheses indicate the quantities available; numbers in square brackets indicate UPS shipping charge. If you don't already know what an item is or can be used for, then you don't want it.

(3) *Microsoft BASIC Decoded and Other Mysteries*, by James Farvour: \$5 [\$3];

(3) *TRS-80 Disk and Other Mysteries*, by H. C. Pennington: \$5 [\$3];

(1) *BASIC Faster and Better & Other Mysteries*, by Lewis Rosenfelder: \$5 [\$3];

(1) BFBDEM Disk Software for the TRS-80: \$5 [\$3];

(1) *Games and Graphics for the TRS-80*, by Tom Dempsey: \$4 [\$3];

(1) *Getting Started on the Radio Shack PC-2*, by H. C. Pennington: \$4 [\$3];

(1) *The Custom Apple & Other Mysteries*, by Winfried Hofacker: \$2 [\$3];

(1) Electric Pencil V 2.0 for Model I: \$10 [\$3];

(2) *The Custom TRS-80 & Other Mysteries*, by Dennis Kitsz: \$5 [\$3];

(2) A-B switchbox - DB-25 serial ports: \$10 [\$5];

(15) bare XLR8er board with 256K: \$100 [\$5];

(1) American Power Conversion AP200 UPS: \$150 [\$10];

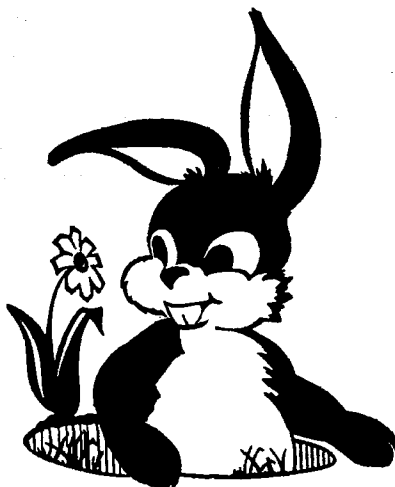
(39) Used Xebec S1410 controllers: \$30 [\$5]

## Ribbon Cable Assemblies

MISOSYS uses a Cirris Systems cable tester for testing custom manufactured ribbon cables. This equipment performs a 100% test for shorts and opens on cables. It handles many different kinds of connectors based on switchable test assemblies.

I have an assortment of test assemblies to support the kinds of connectors typically associated with the TRS-80 microcomputer. Thus, I'm custom fabricating low-volumes of cables according to your specifications, as well as providing standard replacement cables for your needs. These are all using unshielded ribbon cable. I can provide cables using DB-25 M/F, 20-pin header; 34-pin edgcard M/F, 36-pin printer, 50-pin edgcard F (not 50-pin male), 50-pin SCSI M/F, 34-pin Header M/F, as well as DB9 Male. See my latest price list for the cost of *standard* TRS-80 cables. I also have Kel-Am male and female connectors available - while they last; these will definitely not be re-ordered. Kel-AM 34-pin male edgcard @ \$8; Kel-AM 34-pin female edgcard @ \$5. You usually need at least a pair of them. S&H is extra.

**Eh! Look at those bargains!**



## WANTED Scripts or SuperScripts Users

Now you can use that fabulous word processor from Radio Shack with your non-Tandy printer! No point in missing out! Your printer CAN work great with SuperScripts+!

## PowerDRIVER

*These printers are fully supported*

EPSON MX-80/100 Series  
EPSON FX/RX 80/100 Series  
Most all EPSON "compatibles."  
C.Itoh 8510 Prowriter  
C.Itoh Starwriter Daisy Wheel  
C.Itoh A10-20 Daisy Wheel  
Okidata 92/93 Dot Matrix

*Model 4 Now Supported!*

All features of SuperScripts are supported to the fullest capabilities of the printer involved. Drivers are easy to install, easy to use, and no extra commands to learn!

**PowerDRIVER** Only \$17.95 each

**PLEASE SPECIFY PRINTER TYPE**

## PowerSCRIPT

*A HIGH-POWER modification for Scripts from Radio Shack.*

Give Scripts Full Power! PowerSCRIPT is a modification for the original Scripts/LC which allows it to work on Model I or III and gives it the power of many of the newer expensive word processors. Define printer codes for ANY printer. Embed printer codes in the middle of a line! Alphabetized directories! User definable printer filters! User-definable HELP-FILE! Chain files together at print time from any part of the text. Logical EXIT to DOS, optional automatic linefeed after C/R, FETCH, CHAIN, and KILL command and more! Compatible with most major DOS's. Simply use the enclosed INSTALL program, and you will be up and running in NO TIME! Includes full documentation. Get lots of new features for a very small price!

**PowerSCRIPT** Only \$24.95 each

US/Canada Please add \$3 Shipping/Handling Foreign \$6

**POWERJOFF**  
PRODUCTS FROM MISOSYS, INC.

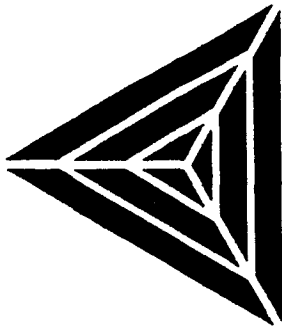
**MISOSYS, Inc**

P. O. Box 239

Sterling, VA 22170-0239

703-450-4181 or 800-MISOSYS

# Letters to MISOSYS



## Response to TMQ V.iii

**Fm MISOSYS, Inc:** Apparently some of what appeared in my last issue caused a little bit of ruckus. Due to an oversight on my part caused by not carefully reading what was sent to me, a portion of the "Recovering Superscripts Documents" article which appeared on pages 36-37 attributed to Dave Kelton were, in fact, derived by him from other sources and referred to me as an internal compendium. I made the mistaken assumption that anything sent in to me not marked PRIVATE is usable for TMQ. A minor flame concerning the piece appears in David Goblen's letter. I offer my apologies for the oversight, and I am publishing Dave's letter to David.

There were some words stated relative to Frank Slinkman's update to David Goblen's XLR8er patches from the previous issue. This is covered in David's letter and I publish Frank's response to balance the claim.

Lastly, on this subject, David also took Matthew Reed to task for burdening his time writing another mouse driver which I published in TMQ. Matt's response is also published.

I hesitate to initiate any me-toos, but sometimes folks write duplicative programs for the sheer joy of doing the task. I am sure David Goblen had no real need for another Z80 assembler; although David states in CN80, "I wrote DEA so that it would do everything I expected an assembler to do and then some", no one has any Z80 program development which could not be satisfied with either my absolute code generating EDAS or my MRAS relocatable macro assembler. I am sure that developing such a complex program as DEA provided a great amount of personal satisfaction to David. Thus, no one should be taken to task for writing what another may claim to be is a "duplicative" program.

To tell the truth, in this day and age where the TRS-80 is virtually an orphan and the few remaining businesses attempting to support Model I, III, and 4 machines with products will soon become extinct, I cannot understand bickering over what little information appears in print. Nobody's intentions were dishonorable. There are few sources of products and information. Aside from the few remaining computer clubs, my small circulation of *The MISOSYS Quarterly* and our CompuServe forum (PCS49), Stan Slater's *Computer News 80* publication, Lance Wolstrup's *TRSTimes*, and *TRSLINK* available from certain BBS's, none of which has substantial circulation, we need every word - every article. Let's remember to focus our energies.

**Fm David Goblen, Willimantic, CT:** Dear Roy, First off, I received TMQ on the 22nd of March. Just as a benchmark, or rather the lack of one, on the average, either Stan at CN80 in Wyoming gets his issue 3 1/2 weeks after I do in Connecticut, or 3 1/2 weeks before me. This time there was only a week difference.

Before I dive into the meat of this letter, I wish to express my apologies for the seeming futility of starting something which might evolve into a "round" of some sort, such as the XLR8er subject, but I feel in this case that my personal defense justifies it. After all, I should have the right to my defense.

**NICE NOTES ON YOUR HARD DISK DRIVES:** Thank you so much for the prompt delivery of the "bare" 3 1/2" 20 Meg hard disk drive and connector cables. I really appreciate the pin-1 arrows on the cable connectors, as these were missing from the cables I had obtained from elsewhere, which left you playing "let's try it this way now" until I had things running properly. With your cables, one quick glance assured me I had things right the first time.

It is incredible how much faster this 3 1/2"



drive is as opposed to my ST-225 5 1/4" 20 Meg Hard drive. Additionally, anyone wishing to add a second hard disk to their system should be sure that they have their reference sheets for their hard disk controller card handy, because on most cards you have to add or change a simple jumper setting for it to accept 2 drives. Also be sure (like I had at first forgot) that the drive select jumper on the second drive should be set to DS-1 rather than DS-0, as the first drive is. Rule # 3 is to be sure that pin-1 orientation on the cables is matched to the pin-1 posts on the controller and the hard disk. With these simple rules, it should be a fast and painless operation to add a second hard disk drive to your system. BTW, a second drive is like a breath of fresh air. No more fudging it by running some applications from a 720K disk, just to ensure enough free space is left on the hard disk.

ON XLR8ER PATCHES: I almost felt like the "star" of Volume V.iii. Though under protest, I can appreciate Frank Slinkman's desire to work on my XLR8er patches to "correct" them to work with unruly applications. Considering how easy it is to use the various functions available when using the @BANK SVC, it is sad that even distributors like Tandy, who have in the past proclaimed that programs which do not abide by the rules set down in the technical reference manual will not be approved or supported, have in fact apparently broken their own rules. With all the diverse add-ons for the TRS-80, it is imperative that the rules are indeed followed, especially now that cooperation is vital just to keep the limited support among ourselves on a sane level. It is important that we bring attention to such wandering to each others attention, especially when one does not realize they are indeed breaking those rules. It is a pity that my correctly operating interface should have to be patched to allow the renegade code to operate. AARRGGHH!

By the way, both of Mr. Slinkman's patches for BOOT/SYS, as published, will not operate! There is a serious error in XLBOOTB3/FIX. Notice the patch lines

starting with D0E,00= and F0E,00= are 17 bytes long, but the next set of patches start at offset 10H, 16 bytes up, which will overwrite one byte from the previous patch line! This patch will also crash the system! This is made all the more dangerous because the O=N parameter is used, which causes PATCH!! not to verify the FIND lines. Had Mr. Slinkman verified his patches manually by directly patching the disk before issuing these patches? The O=N parameter is very powerful, and as such patch authors should make excruciatingly sure that such patches are indeed correct before telling people to use such a parameter (which was actually not even needed because my patches had already extended BOOT/SYS for another record). The error in his XLBOOTB4/FIX is much tamer. The D03,26= line should be changed to D0E,26=.

Now comes the "however" part: Had Mr. Slinkman considered that he should check with the author of a system enhancement before he goes about trying to "correct" it? Even I had tried to contact Mr. Houde before I submitted my own XLR8er patches, but my letter was returned. Had Mr. Slinkman done so, he would have found out that all those changes he applied were not necessary. Indeed, only -2- lines in my original patches need to have - simple- changes to them in order to create the separate BUR\$ he spoke of: In XLSYS0B1/FIX, the patch line beginning with D0D,60= should change the sequence 21 00 02 to 21 10 04. Also in that line, change the 7E to AF. In XLBOOTB1/FIX, in the line beginning with D06,80=, the sequence 2E 02 should be changed to 2E 12, and the sequence 26 02 should be changed to 26 04. What I have done with these minor changes is place a new BUR\$ for use by the XLR8er patches in an unused portion of low memory, in the same place the SuperMEM patches you wrote would hold their extended BUR\$ and BAR\$ registers. As a side note, you may also notice that even with the (I guess) unnecessary OR A instruction still in the VDCTL patch, this causes my patch to be 1 byte shorter than his.

To be complete, here in patch form are the only changes needed:

```
.XLBOOTB2/FIX -- Move BUR$
to un-used RAM area. (24-
Mar-91 David Goblen)
. Must also PATCH sys0/
sys.system6 USING
xlsys0b2/fix
. Apply to disks which
have had XLBOOTB1/FIX and
XLSYS0B1/FIX applied via:
. PATCH boot/sys.system6
USING xlbootb2/fix
d06,82=12; f06,82=02
d06,8d=04; f06,8d=02
.eop
```

```
.XLSYS0B2/FIX -- Move BUR$
to un-used RAM area. (24-
Mar-91 David Goblen)
. Must also PATCH boot/
sys.system6 USING
xlbootb2/fix
. Apply to disks which
have had XLBOOTB1/FIX and
XLSYS0B1/FIX applied via:
. PATCH sys0/sys.system6
USING xlsys0b2/fix
d0f,61=10 04 36 F8 23 AF
f0f,61=00 02 36 F8 23 7E
.eop
```

I would also like to protest Mr. Slinkman's coarse phrasing which would lead one to assume that there was a MAJOR problem with my concept. My premise was sound, so how can there be a major problem with it? Just because an application program's author did not follow the rules does not mean that there is something wrong with my code. Based upon that phrase alone, I have already received one "hate" letter (and expect more) condemning me for submitting faulty patches. Why not vent anger at those who purposefully break the rules so that properly operating code will not work? I would never dream of changing anything in Mr. Slinkman's programs without checking with him. I know that MISOSYS follows the same policy. Every patch to 6.3.0 and 6.3.1 I have published in CN80 has been first submitted to you for checking, since those systems are your babies. As a capper, if a program

developer were to have to research every program "out there" regardless of whether they followed the rules or not, no new programs would ever be developed, because renegade code would always stop them cold, as someone would always find a "major" problem.

Speaking of MAJOR problems in concepts: In Mr. Slinkman's enhancement patches, referenced in his letter as the "B4" patches, which slow the XLR8er clock down prior to keyboard accessing, and then speeding it up again afterward: since the patch forces the clock rate to a setting of 2,1,80 before scanning, and up to 0,1,80 after scanning, it may have been a good idea to have included a warning that this patch may cause problems with some XLR8ers. It has been a commonly reported problem among many Model 4D users who have to use the 6" connector cable that memory faults can occur when M is set to 0 (such as on mine), due to additional noise from the extended cable. And what if someone is purposefully trying to run their clock at a slower rate? They may quickly begin to wonder why their SETX/CMD or SET180/CMD programs no longer seem to be working when they try to slow the clock down, as the patch will force it back up after the next keyboard scan. This is just one more thing a programmer has to consider when writing software: will what I do pose problems for the user? When I had first developed my XLR8er patches, I had originally included a keyboard I/O speed shift feature due to all the room suddenly gained by the shorter patches. My patch solved this high gear problem by saving the current M and I values, slowing the clock down, stroking the keyboard, and then restoring the saved clock speed. This was accomplished in a single 24 byte patch. Rather than intercepting both the KISCAN and CTL255 routines as Frank's patches did (which would require two different patch routines), I had instead intercepted the call to the TYPAMD routine at x'0929' in the \$KI driver to call my patch. This method automatically takes care of both KISCAN and CTL255. This patch ran wonderfully. However, the single justification for me dropping it as a general

public use patch was that for some reason, after entering SCRIPSIT Pro, the clock is left at 2,1,80 (how this happens I'm not yet sure), though all the many other applications I tried running with it ran fine, and did not cause any alteration in the clock setting. Even though I could personally live with this inconvenience, I decided to drop it entirely because I felt that other applications not yet tried may do the same trick which PRO apparently did, thus causing aggravation on the part of the user.

For technical thoroughness, following is the patch lines I had dropped, which anyone can add if they choose, with the caveat that at least SCRIPSIT Pro will "fiddle" with the clock, and can live with this inconvenience. I have constructed it as a stand-alone patch, which should be applied after my previously published patches, modified as outlined above or not:

```
. XLBOOTC1/FIX -- Allow
clock slowdown during kbd
access (18-Sep-90)
. Made standalone 24-Mar-
91. David Goban.
. Apply to disks which
have had XLBOOTB1/FIX and
XLSYS0B1/FIX patches
. added (see TMQ V.ii)
with optional changes
outlined in XLBOOTB2/FIX
. and XLSYS0B2/FIX
. Apply via PATCH boot/
sys system6 USING
xlbootc1/fix
. fix lomem pointer for
added patches
d00,06=2D;f00,06=15
. intercept call to
TYPAMD at x'0929'
d07,2a=15 10;f07,2a=D5 0A
. fix modend ptr for $FD
d0c,3f=2C;f0c,3f=14
. add patch to slow clock
during keyboard access
d0e,15=F5 ED 38 32 32 27
10 3E 80 ED 39
f0e,15=B6 6D B6 6D B6 6D
B6 6D B6 6D B6
d0e,20=32 F1 CD D5 0A F5
3E 00 ED 39 32 F1 C9
f0e,20=6D B6 6D B6 6D B6
6D B6 6D B6 6D B6 6D
.eop
```

By the way, does SuperSCRIPSIT in fact use the @BANK SVC's BUR\$ and BAR\$? When I use SS for anything, I always use it from a MemDISK which uses all of extended memory, and have never noticed any two-fisted bank management being performed on my XLR8er patches. Even SCRIPSIT Pro runs fine as long as I reserve the first 2 banks for its use, because it expects to use only those banks. PRO uses the @BANK SVC to do all its bank switching. The only problem is that it will use these 2 banks even if something is already assigned to them. That's why I reserve the 2 banks, so that my own extended memory applications will not interfere or be crashed. Also, I was not aware that Deskmate used alternate banks, as I had used it for quite some time on my Model 4p before I had acquired extended memory. Hm. At least I must have missed that documentation in the manuals, because I cannot find it even now. Some initial tests have shown that it did not for some reason work properly, though with my above-mentioned alterations to my patches, it will.

ON THE MOUSE INTERFACE: Matthew Kent Reed's mouse interface was interesting. However, he could have saved himself a lot of time and trouble if he had only reported the problem to either CN80 or me. If he had, he would have received a free update in a matter of a couple of days which corrected this problem. Up until I read Mr. Reed's treatise, I had known of only one other person who was using a 2-button mouse. I worked closely with this individual to resolve the problem, and version 1.1.1 of my driver, which has been out for quite some time now, does in fact work with the Microsoft mouse. The problem was that 2-button and 3-button mice communicate in wholly different manners with its host, such as when one uses a positive value to indicate one direction, the other uses it for the opposite direction. There are other more intricate differences, and were outlined in part II of my article on mice in the December 1990 issue of CN80. Of course, Matthew's driver does have the advantage that you do not have to tell the driver that you are using a



two button mouse, as mine does with the B2 parameter, because my driver defaults to the more common, and usually much less pricey 3-button mouse. And so even though I will complain to users that they should always go back to their distributors to report problems if the purchased product does not perform as documented in its manual, and the fact that such problems can in most cases be quickly resolved by the distributor, I must also compliment Matthew on his creative solution. But again, such effort would not have been necessary if he had only brought his problem to my attention via CN80. I by no means have a desire to distribute buggy software, and I think most people know by now that if a bug is found, I will fix it immediately and provide a free update or the patches to fix it (this policy is also standard with CN80 and MISOSYS). The fact is, this bug was fixed sometime in November, 1990. But since most people who bought the driver were using it with a 3-button mouse, CN80 had deemed it not necessary to send those folks an update, as it would not have any affect on them. If I found a bug in Little Brother, should I report it or write my own version to correct it? Personally, I think the writing part would be stupid. I know you would agree, as you service what you sell with great devotion, and would either correct the bug immediately or tell me what I am doing wrong. This is no slam against you. I just used LB as an example.

Going further on this note, anyone who develops software which uses the mouse interface SVC as outlined in both my own and Matthew's article are free to use my mouse driver with any commercial (non-PD and non-Shareware) package they develop without having to pay royalties to me, as long as they had purchased at least one copy (\$5) from CN80, they request and receive permission from me to use it (which won't be much of a problem), and they provide in their documentation the source credit and author of the driver, and they do not alter the driver in any way, except as allowed by written permission from me. I have been wanting to get some mouse software out, myself, but time has

been at a premium lately. So for the time being that job must be left to other and perhaps more creative individuals.

**ON SUPERSCRIPSIT RECOVERY:** In regard to David Kelton's article on recovering SuperSCRIPSIT documents, I suppose imitation is the best form of flattery, but I think that there is a big difference between referencing another's material and just using it out-right. I certainly do not appreciate the fact that he references my 80 *Microcomputing* and *Computer News 80* articles as simply references for his own feature, when better than 80 percent of his article is a word for word transcription of my own text from those two referenced items. Try reading my 80 Micro and CN80 articles to see what I mean. Deja vu. I went through a lot of work to compile that information, deriving most of it through long hours using a disk utility and a disassembler, and tried hard to ensure that it was accurate, and so I don't take too kindly to some guy copying it verbatim and claiming it as his own original writing. The least he could have done was rearrange it and change the words around. But no, he chose to copy it just as he had read it. I don't blame you for this oversight, as a publisher has trouble enough trying to paste up the magazine in a readable format. After all, you do have to exercise a certain amount of trust in your authors without having to police each submission for pirating.

**ON LAIR OF THE DRAGON MAPS:** I would like for your readers to know that Roy Soltoff created those wonderful maps of my "Lair of the Dragon" adventure in TMQ V.iii. I have already received a couple of notes telling me what a fine job I did. Well, I just want to pass those pats on the back over to Roy, who had taken my own original and crude charts, along with the exit tables inside the adventure, and constructed a beautiful set of helpful, informative maps. Looking at the adventure in this light, people can certainly see the involvement and vastness of the adventure, and the temptation to explore all those many places. I think anyone who already has "Lair" will find out two bits of information in the maps

which some were probably prepared to pay a fortune for: the location of the keys and the flashlight. Those were the most often asked questions users have put to me when they had requested the hint sheets. Also, I think the idea of you publishing the hint sheets for the adventure in a future issue is a really good idea.

With that I'll close this novelette, and let you get back to more important things. Best regards, TMQ is great, and thanks for the support!

**To David Goblen fm David J. Kelton:** David, No one, including you, was more surprised that I was to see the article "Recovering Superscripts Documents" appear in TMQ V.iii under my name. I was more than a little embarrassed because I had not submitted the material as an article for publication.

On page 11, you will note that I said in my cover letter, "I'm sending along a copy of my files regarding all the information I've accumulated from 80 Micro, CN80, and TMQ regarding the structure of Superscripts files." I had intended it for Roy's use in answering questions about Superscripts as I thought it was a pretty good compendium of information. Never once did I indicate that I wrote the material, but unfortunately, I also never clearly stated that I didn't write it. Attached you will find the first page from the material I submitted. I think this page clearly shows that the material was extracted from articles by yourself and Tom Price. No where does my name appear as an author.

The only reason that the material is not 100% as you wrote it is because it is a compendium of the three articles mentioned with a little of my own experience thrown in.

I am truly sorry this incident occurred. I am an avid reader of yours (I suppose that's obvious), and I wouldn't dream of ever pirating your (or anyone else's) software or articles. Please accept my apologies.

**Fm Frank Slinkman:** Since writing my patches to David Goblen's new @BANK routine (complete with my editing errors, for which errors I apologize, and for which I thank Mr. Goblen for both alerting your readers and for providing working alternatives to), it has been reported to me that some functions of Allwrite do not work using the original Goblen patches.

I certainly agree the problems with this and other programs are not Mr. Goblen's fault, but the fault of other programmers who broke clearly established rules.

We should not have to take their errors into account when writing our programs. But, fairly or unfairly, given the situation in the "real world" out there, the fact is that we must, especially when writing code of this type. We are now seeing the consequences of what happens when you don't.

Any "hate letters" Mr. Goblen may have received as a result of all this are extremely unfair to him for this reason, and are clearly overreactions. But I think it's equally unfair to try to blame me for engendering such letters simply because I used the adjective "major" to describe a very real problem.

As far as my not contacting Mr. Goblen before altering his code, I point out that Mr. Goblen's code (except for the added function of motherboard RAM bank switching) is virtually byte-for-byte identical to the code I wrote for Richard King's 512K hardware mod more than a year ago (see "All the Way to 512K," TMQ IV.ii and "512K a Better Way," TMQ IV.iv). I felt I knew the code well enough to "wing" it.

It was never my intention to "declare war" on Mr. Goblen, but merely to try to help by pointing out some facts, albeit unpleasant ones, and propose some solutions. Nor do I consider myself in any kind of "anything you can do, I can do better" competition with Mr. Goblen. I consider him an excellent programmer who has made some solid contributions toward helping keep the TRS-80 alive. I am merely trying to do the same thing.

**Fm Matthew Kent Reed to David Goblen:** David, I just received my copy of your March 26, 1991 letter to Roy Soltoff which mentioned my Model 4 mouse driver. I assume you printed the letter using one of your soft font laser printer drivers for SuperSCRIPT and SCRIPT PRO, as you mentioned in The MISOSYS Quarterly, volume V.ii, pp. 35-6. If so, I compliment you. The finished product looks very striking.

I received a disk a few days ago from Computer News 80 containing the updated version of your mouse driver, and I have examined it with great interest. I see that you completely revised the two-button portion of your mouse driver to work with Microsoft compatible mice.

My mouse driver seems to have stirred up a bit of controversy. Both you and Stan Slater said that I could have saved myself a lot of trouble by requesting your updated mouse driver from *Computer News 80*. However, I never thought of doing that because I had come to the conclusion that your driver was not designed to work with Microsoft mice. I came to this conclusion in the following way. After I read your first article on Model 4 mouse use, I was intrigued and decided to buy a serial mouse for my Model 4. To gather information, I studied the two "mouse round-ups" in *PC Magazine* (February 14, 1989, and August 1990) to determine which mouse model would best serve my needs. The articles gave me the impression that there are two main hardware standards for mice: Logitech and Microsoft (see *PC Magazine*, February 14, 1989, pp. 256-7). Since I have had previous mouse experience on an IBM PC compatible, I wanted a mouse that was hardware compatible with Microsoft's mouse, the industry standard. For that reason I chose the Omnimouse II, a two-button serial mouse. *PC Magazine* had even tested the Omnimouse with Microsoft's own mouse driver, so I was assured that the Omni would work reliably and predictably on my Model 4. Also, I preferred the smooth, contoured feel of the two-button Omni over the design of other, slightly less expensive 3-button models.

As I stated in my article, the first version of your mouse driver (which I purchased from *Computer News 80*) did not work with the Omnimouse II, even though I had used the "B2" parameter. Furthermore, the results of my test programs indicated to me that the problems did not result from "buggy software," but that my mouse was simply not compatible with your driver. I looked closer at your mouse article and found that you mentioned using a Logitech mouse, a Genius mouse, and a Tandy serial mouse. I had been told that Genius mice adhere to the Logitech standard (although I now suspect that to be false), and I had reason to believe that Tandy's serial mice emulated Logitech mice as well. My conclusion was that your mouse driver was written to support Logitech compatible mice, and that your article never mentioned the differences between the Logitech and Microsoft standards because your mouse driver was written primarily for three-button mice, most of which use the Logitech standard.

I never reported my conclusions to you or *Computer News 80* because I didn't feel you were responsible for the problem. In your letter to Roy, you wrote, "If I found a bug in Little Brother, should I report it or write my own version to correct it?" My difference with that statement is that I didn't feel there was a bug in your mouse driver. To amend your example, should I report it to Roy as a bug that Little Brother doesn't run on the Model III, or should I realize it was never designed for that purpose and do something to fill the gap myself?

I enjoy programming challenges. Planning and writing a two-button Microsoft mouse driver seemed like an interesting and useful project to undertake, so I reviewed my notes and literature about RS-232 interfacing and spent an enjoyable two weeks in October 1990 writing MMOUSE/CMD.

Since I appreciate your contributions to the TRS-80 world, I felt it was essential for my mouse driver to follow the SVC standard you had already established. As

I stated in my article, any software written for one driver should work without change on the other. I have always intended my driver to complement yours, not to replace it. I realize my driver will have very limited distribution, since few people will be using two-button mice with their Model 4's. However, I wanted to present the mouse driver and its hardware interface information publicly, just in case someone else was having the same problems I had experienced.

Our mouse drivers now seem to operate similarly, and this leads to the question of which driver future two-button mouse owners should use. Again, I don't think there is a conflict. Your updated driver appears to have been written for mice with fairly high resolution, such as 350 ppi (points per inch) or above. My driver was written for low resolution mice, such as my 200 ppi Omnimouse. Therefore, I would suggest that two-button mouse owners test their mice with each driver to see which one gives the best response.

I want to thank you for your revised XLR8er patches from The MISOSYS Quarterly, volume V.ii. I have not yet had the opportunity to install them on my Model 4, since I have been busy on several fairly involved programming projects, but I am looking forward to freeing up more low memory. I especially appreciate the keyboard slow-down patches supplied in your letter to Roy, since my Model 4D needs to decrease speed before accessing the keyboard. I applied a similar patch to Michel Houde's XLR8er module several months ago, and that has eliminated nearly every problem I have had with the XLR8er. The only problem that persists is that ever since the XLR8er has been installed, RAM banks 1 and 2 (the original extra memory on my 128K machine) have been unable to hold their contents without corruption. It doesn't matter what speed the XLR8er is running at, and it didn't help when I installed 120 nanosecond RAM chips. This problem was reported to Roy (see The MISOSYS Quarterly, volume III.iii, pp. 24-5), but as far as I know, no one has been able to explain it. The addition of

eight extra banks far outweighs the removal of two, but that extra 64K could sure come in handy sometimes!

Thank you for sending me a copy of your letter, and I look forward to seeing more of your articles and programs in the future.

---

## Metrics...

---

**Fm Lawrence Rossiter, Victoria, BC Canada:** Please mention in the next issue of TMQ what progress is being made with the rewrite of Little Brother. I note your ad still lists Version 1.0 which I have had for many years. There has been no mention of the new version for quite some time, prior to that it was reported on frequently.

I felt quite honored to have my letter about metric printed in your Blurb! One small error crept in there somehow, the year in the line "by July 1, 1892, and by the nation at large...." should have been 1898, Makes more sense as the article was published in 1896! Best wishes to you and the family.

**Fm MISOSYS, Inc:** I usually add a gotcha or two to keep folks on their toes. And you will note from this issue that LB Data Manager Version 2 is released. Got it out by April 30th.

---

## Lair of the Dragon Maps

---

**Fm Will I. Ramsey, Jr., Gainesville, FL:** Dear Roy and David, I write this letter to the both of you since I do not know which of you will have the answers to the specific questions I have. Perhaps one of you can answer everything; perhaps it will

take a reply from both of you. As usual, I try to cover all bases!

I decided to use the maps you printed in TMQ to make one large map of the entire area covered by the adventure. I wanted to do this so that directions would be constant (i.e., north is up, south is down, west is left, east is right) in all regions and areas of the adventure. In the course of making my map, however, I discovered that some directions apparently were omitted from the maps as printed in TMQ. I do not know if this is a typesetting omission or what. But without the missing directions, I do not know how to complete my map because I do not know how to "tie in" one portion of the map with another. Here are my "problem" areas:

1. Sheet 6: There is no direction given "above" # 125 (long tunnel) so I do not know how (i.e., in which direction) to tie this in to #124 (inner anteroom). Similarly, #124 (inner anteroom) on Sheet 5 does not show a direction to connect to #125 (long tunnel).

2. Sheet 7: ## 139, 140, 141, 142, 164, and 143 are all connected together but they do not appear to connect to any other portion of the map. Obviously, there has to be a connection somewhere!

3. Sheet 7: At the bottom right-hand corner is a "jump" to #121 (mountain trail, sheet 5). However, #121 (mountain trail) on Sheet 5 does not show a "jump" to Sheet 7, as one would expect.

4. Sheet 7: No directions are given between #145 [ridge (rail)] and #146 (platform). I suspect that you go down to the platform from the ridge/rail and up to the ridge/rail from the platform, but I'm not sure. Obviously, I haven't gotten this far in actual play, so I can't test my theory in practice!

After I finish my map, if you guys would like a copy I'll be happy to send you one. (Rest assured! I do not intend to go into distribution of this map and make tons of money and retire at an early age! It is



strictly for my own use, but I figure I could share it with you guys if you were interested. This assumes, of course, that you don't already have your own, which, as I think about it, is actually quite likely.)

By the way, David, thank you for giving me the hint about feeling the floor in the basement of the inn to find the flashlight. From there, it was smooth going for a while. The giant black widow spider gave me fits, though, because none of my weapons had any effect on her. I finally learned that I could (usually) get away from her by just running like hell through the black forest maze (using the maps provided in the hint sheets was a great help). But now I'm stuck again at #123 (cave anteroom). I've seen my share of spy movies and thief movies, but getting that padlock open has me completely stymied. I have found — and have in my possession — every object there was to find up to this point, and the maps bear me out on this. And I have tried every single object on that padlock. Several things will destroy it but none will unlock it so it can be removed. Feel like dropping any more hints, or am I really left on my own this time?

Thanks, guys, for your assistance in this all-important inquiry. Obviously, all work here has come to a grinding standstill as I dedicate all my working and waking hours - well, working hours anyway!

**Fm MISOSYS, Inc:** Will, from what you may have read in David's letter which precedes this, you should know that I prepared the Lair of the Dragon maps which appeared in TMQ V.iii. Adventurers should also be aware that in any complex adventure, not every connection is a two way street! Also, some ingress/egress routes do not materialize until some future event occurs. I had to acknowledge all of these situations when I prepared the map. Some errors did creep in; however, not all of your *alleged* inaccuracies - to use a legal term - are, in fact, errors. Let's go over the discrepancies.

If you get to room 125 (sheet 6), the room's text provides the direction to travel; it is North. Yes, it was a minor omission. I did try to arrange the rooms on each of the map pages so that the directions NESW were typically URDL on the page; but, with a three-dimensional 160+ room adventure spread onto seven pages, that was an impossible task for every room.

Due to a transcription error, room 141 of sheet 7 omitted its connection to sheet 6, room 138. On the other hand, sheet 6's room 138 (west hall) is positioned not more than two inches on the map with a clear direction EAST to 141: sheet 7. I consider that error to be minor.

Did you ever hear the expression, "You can't get there from here?" Just because you can jump from 160 to 121, that doesn't mean you can jump back! You can't. I can jump from my garage roof to the ground (hopefully not break a leg in the process), but I certainly can't jump from the ground to the roof. Got it?

And the dashed lines used in the map between two rooms generally indicate that no connection exists by means of a direction movement between the two locations. For instance, no directions are shown between locations 51-52-22 on sheet one. That's because you can't travel in a direction to move from one of those locations to another. But you can get from one to another via some other means. You have done that, haven't you?

Since I'm on that subject, perhaps others need the hint. If you start out Lair and go to the South Bank from the East-West road, you can't go across the stream. But if you get to the North Bank from the North-South trail, you find a ferry boat. Enter the ferry then pull the rope. That moves the ferry to the South Bank. But you don't move N, E, S, W, U, or D.

Now personally, I don't need a single-page map - it's too big to use. David did have a map drawn on about 8 pages taped together; that was more directionally correct from a geographic standpoint than

mine printed in TMQ. But I was under a handicap. I had to fit mine on individual 8.5" by 11" pages. The symmetry of the locations was a result of using Pagemaker to draw each oval and add each text piece as a separate text block. I found it convenient to use a grid line consisting typically of nine by three blocks.

---

## The saga of DoubleDuty

---

**Fm Richard Politowski, Houston, TX:** Thank you for your continued support of the Model 4 world and your continued publishing of the TMQ to which I am a subscriber.

Your batting average is fabulous but somehow you missed the mark when you questioned reader Richard R. King's remarks (Vol. V.iii pp. 7-8, 14-15) concerning DDUTY/CMD and the time/date functions in each partition. Double-Duty has ALWAYS kept time independently in each partition, including partition #3 (library commands only), at least on a standard 128-K Model 4 machine of any vintage.

Time stands still in each background partition, naturally, but continues where it left off when the partition is switched to the foreground or active position. This is, of course, much easier to prove by setting, as reader King did, a distinctly different date and time in each of the three partitions. I would suggest trying this with an unpatched, unconfigured boot-up of the DOS on a "virgin" machine if one is available.

This was always a problem with Double-Duty even though I found the program very useful when writing and testing BASIC programs and later when writing PRO- ENHCOMP programs. The problem DISAPPEARED on my computer when I purchased an external time clock from Storage Power and patched BOOT/SYS as instructed.

Since you now sell a harddrive system which can include an external clock, I suspect reader King's problem disappeared because he eventually booted to (or from) the harddrive after perhaps "testing" DDUTY in a configuration which did not access his external clock.

The patch as supplied by STORAGE POWER does NOT allow one to effectively reset the TIME or DATE function without accessing the external clock to do so. I assume your implementation does something very similar, beginning at either port 32 or port 64 (decimal). My setup is for #64.

Please do not consider answering this letter personally. I'm sending it to contribute to the discussion concerning DDUTY and am certain some further explanation in a future TMQ will be sufficient.

**Fm MISOSYS, Inc:** Richard, your letter caused me to spend about eight hours of my time to get at the root of the *alleged* problem. Part of my time was spent in evaluating the source code; that took about three hours of scrutiny to confirm that DoubleDuty did not explicitly save either the date or time for each partition. Those system variables, as well as most of the system low memory, are common to all partitions. Specifically stored as global variables are items such as the interrupt pointers, Job Control Language FCB, video screen memory, keyboard speed buffer, some system flags, and a few other entities. My testing was previously done using the latest version of DoubleDuty, release 2.6.0. Although my system has a hardware real time clock, the date is only recorded on bootup. Thus, changing the date in one partition after booting and installing DoubleDuty would have been affected if DoubleDuty performed as Richard and you proposed. On my version 2.6.0, it didn't. The date was uniform through all partitions.

But I revisited the query while assembling

this issue of TMQ. Rather than continue to use my *normal* system, I booted up with a stock 6.3.1 LS-DOS disk. I then had to pull a copy of DoubleDuty from an archive file. But it turns out that I pulled out a 2.5.1 version. Using that version, I duplicated the problem both you and Richard raised. Date, time, and indeed other low memory values were unique across partitions. The difficulty in ascertaining the cause of that is typically immense as the memory shuffling performed by DoubleDuty makes it a difficult animal to debug. Since the source code provided no clue, I had to perform some intense debugging. I was able to simplify the process by re-booting my original system environment - 4D with XLR8er and 40M hard drive with hardware clock - by installing PRO-WAM in bank 3 then installing version 2.5.1 of DoubleDuty using banks 1 and 2. I was able to ignore the accurate time being retained by my hardware clock by evaluating the date. I was also able to store a unique string of characters in the USTOR\$ memory area (13H-17H). By switching between partitions, I was able to note using DEBUG that changing the value of the string in one partition kept it unique from the value in the other partition - when it should have been identical across partitions.

At this point, I invoked PRO-WAM and activated the Memory Editor (MED) application from the MisterEd package. Having PRO-WAM and DoubleDuty both in memory at the same time provided the ability to easily examine banked memory. Knowing how DoubleDuty worked, if the string was maintained unique across all partitions, it had to be swapped into banked memory. I used MED to search for the string of characters. It showed up in bank 2 at address 8F:65 (that's page 8F, relative byte 65). It also appeared that some piece of page 0 system memory was stored in that bank page. I then examined the source code to see what should be at that location. Well, to tell the truth, nothing should have been, if you were using DOS 6.2 or greater. It took a while of digging, but I found the cause - turned out to be an unknown bug which was inadvertently eliminated when

I changed some code while implementing version 2.6. Here's the scoop:

TRSDOS versions 6.0 and 6.1 used system memory immediately following the keyboard driver to store the type-ahead speed buffer. One of the changes LSI made in the 6.2 release in order to add additional features in 6.2, was to relocate the speed buffer to the video RAM region physically above the highest screen address. Remember that your machine has 2K of video RAM and the video screen is but 80 columns of 24 rows (and vice versa). Multiplying rows by columns reveals that 1920 bytes are used to store the screen; that leaves 128 bytes remaining in the video RAM unused. That is now used by the speed buffer.

DoubleDuty has various sections of code to capture and swap data. One of the tables it uses is a pointer table having a word length followed by a word address. Turns out that the speed buffer length and address values are the last entry in the table. You see, under 6.0 and 6.1, the speed buffer is in system low memory and can be swapped just like any other data area in low memory. Under 6.2, a special routine has to swap in video memory, capture the speed buffer, then swap video memory out. DoubleDuty's initialization code inserted the length of the speed buffer into the table (the length is different for 6.1 versus 6.2). Then, if the DOS version was 6.1 or earlier, it inserted the address of the speed buffer into the table. The address would be left as a zero value under 6.2 and greater.

Another routine used the length value to determine if it should capture the 6.2 speed buffer or not. But that routine also had a bug in it. Actually, under DoubleDuty 2.5, the 6.2 speed buffer was never captured. Thus, the single buffer was global across all partitions. But that was most likely never noticed. What was noticed was the fact that the routine which swapped system memory described in the pointer table found an entry of length 52H (the length of the 6.2 and beyond speed buffer) but an address value of zero. So it wound

up swapping memory starting at byte 0000H up through 0051H. Most of those address locations were already captured explicitly (interrupt vectors, RST vectors, etc.) for the global system data, but that region - which had data elements which weren't supposed to be captured - included the DOS time and date storage regions. That's why the "time stands still in each background partition" as you stated in your letter. The bugs were twofold: (1) If the DOS version was 6.2 or greater, then the length field of the speed buffer should remain zero inhibiting the system data capture routine from capturing any speed buffer. (2) The routine which was supposed to capture the speed buffer had a conditional call of CALL Z,nnnn when it should have been CALL NZ,nnnn and the tested speed buffer length value should have been stored in a different location from the pointer table if the DOS version was 6.2 or greater. The way the code worked, the tested value was always non-zero; thus, no call would have been made. It would have been crash city if the value was zero and the call was made because the routine being called would have LDIR'd the zero length which would have zapped all of memory. On the other hand, that bug would have been evident right from the start and fixed before the first version of DoubleDuty ever got released!

The solution to the problem is to upgrade to release 2.6. I had to revise the methods by which the speed buffer was captured to overcome the way the Alpha-Tech memory board worked physically. In the re-write, I didn't need to store the length of 6.2+'s speed buffer; thus the swap routine had a zero-length entry in the pointer table. A zero length terminates the table.

Thus solves the confusion over DoubleDuty.

As far as what I do in my clock patch for the MSCSI hard drive clock, I felt some unique solution was needed. I never liked to utilize additional memory to read a hardware clock; memory is too precious to waste in that regard. The system RTC

timer keeps approximate time. But it can get degraded the longer interrupts are disabled. My solution was to utilize the computer's normal RTC for time keeping, but correct it periodically. The big question is how does one periodically update the clock time without user intervention?

I chose to tack on a routine to the DOS close function which reads the hardware clock time and updates the system time. It doesn't read the date, because that would take more code. The system module was a little tight. Besides, the date could only get to an erroneous value under an extremely unlikely circumstance. And if your machine is left on overnight, the system rolls the date at midnight. So my clock solution keeps good time and takes zero user memory to accomplish that.

---

## LB Data Manager

---

**From Donald M Simpson, Queensland Australia:** I have recently purchased LB86 Data Base Manager and LBMU86 Maintenance Utility from yourselves and have been very pleased with the results I have so far achieved. My primary use for the Data Base Manager is to record relevant information about my various Philatelic Collections.

The purpose of this letter is to ask for advise on the use of the LBMU86 programme. Simply I have not been able to proceed beyond the message "Cannot locate data and/or def files." Initially I wrote to Logical systems Inc but yesterday my letter was returned as "Not Deliverable."

To explain the problem as best as I am able, on receipt of the two disks from you I loaded them on to the hard disk of my computer (C Drive). My computer has a hard disk and one floppy drive and is IBM MS-DOS 4.01. The reason that I wish to

use the LBMU86 is that in the future I may wish to add additional fields to existing files.

My procedure for establishing files is in accordance with Option 10 and after the SAVE Command I press <RET> for each of Enter Data Path, Enter Screen Path, Enter Index Path, and Enter Temp Path. I then press <F3>. I then proceed with Option 8, Option 9, Option 2, and finally Option 4.

After establishing my first few files I decided to try the LBMU86 to add an additional field to these files and this is when I discovered my problem. At the prompt "Enter old data base name" I simply entered say "DB18" and all the time the answer comes back "Cannot locate data and/or def file". I did try using Option 14 but as I am something of a novice with computers I didn't have any luck there. I would be very much obliged if you could give me a simple step by step procedure to overcome my problem.

**From MISOSYS, Inc:** To begin with, I have to assume that you have an understanding of the concept of *paths* under MS-DOS, and the concept of *current directory*. When you create a database under LB86, you are prompted for the path location to put each of four classes of files: data, screens, index, and temporary. Pressing just the [ENTER] in response to the four prompts means that you have selected your current drive as the path for each of these file classes (see LB 1.0 page 74, last paragraph). But that means in order to access the data at some future point in time, the current working directory on that drive must be the one which was current when you created the database. I suspect that what you really should have done is to reference the complete path to the files. LB86 version 1.0 doesn't provide a large character string for the path selection, but it should be sufficient for most cases (version 2 now released provides 64 character path strings). Since I don't know what sub-directory you have the database

stored on, I can't be specific as to how you go about correcting it. But here's an approach.

Assume that the directory path where you, in fact, stored the database was C:\DATA. You should invoke LB86, do not select a database, select option 14 (change path), reference DB18 as the database name, then change the data, screen, and index paths to "C:\DATA\". If you have a RAMDRIVE, that is useful for the temporary path; otherwise, use whatever path you generally use for temporary files (your AUTOEXEC.BAT file may even contain a SET TEMP=drive specification which you can use as the path for LB86). If you have no normal temporary drive, use the C:\DATA\ in that case.

LB86 will be able to access the database regardless of the directory which is current - provided the path file is accessible. Also, LBMU86 will be able to correctly find the directory to that data if the pathfile for that database is accessible.

---

## LS-DOS 6.3 Support

---

**Fm Paul D. Moore, Birmingham, England:** I understand that you have taken over, and are still supporting LS DOS 6.3L+ for the mod4/p. I enclose my registration card. I've only just obtained 6.3. Tandy gave it to me because 6.1 didn't support the date past 87, and they have taken 5 years (that's right: FIVE YEARS) to repair my mod4p.

Can you give me some idea of how long you are going to support LS DOS for? Only I notice that the date prompt is set to run out in 1999. I can't afford to upgrade to another machine; and therefore, will be stuck with mod 4p for the foreseeable future. I use a lot of programs that use the machine date. Is there a patch that will allow the date to be input past 1999. I am

not concerned about the conversion to text format.

I am having problems with the clock speed on 6.3. The Hertz 50 patch from 6.1 doesn't work on 6.3. Its very annoying; I usually set the clock while I'm working so I can keep my eye on the time. Any patches available for it?

I notice that some of the sys files are longer than on 6.1. Some programs don't work properly now. Eg: pfs FILE SYSGEN's sysfiles 1, 2, 3, 10, & 12 into highmem, to allow pfs File to be removed from drive 0 for copying/splitting data files. Under 6.3 you now get a '*Not enough memory to install pfs File*' message. The only way round this is to install the sysfiles into Memdisk, and that takes too long, and doesn't allow a memory split to allow 2 programs in memory. And it appears that sysfiles 1, 2, 3, 10, & 12 are no longer enough to support pfs File anyway. You now get ERROR 7H. Yes, I know there are better databases around and I could change to another. But that would mean re-typing countless records. And pfs File has exactly the features I need for this particular database. Any ideas?

Hope you can help. Tandy can't. In England they don't support their current machines and software (and never have). If you still are supporting the mod4, can you send me a software list. Thanks in advance. SAE enclosed for reply.

**Fm MISOSYS, Inc:** Paul, I really cannot answer the question as to how long I will be supporting the DOS; a lot depends on how long MISOSYS stays in business. On the other hand, our most recent release, 6.3.1, has date support through 2011, so the DOS should be supporting dates for quite a few more years.

You don't need to patch the DOS for 50Hz power; just invoke SYSTEM (HERTZ5) followed by a SYSGEN command. This adjusts the clock timer reload value for a 50Hz system. That change was first intro-

duced in TRSDOS 6.2.0 and was documented with the 6.2.0 update pages.

I have no solution to the Pfs problem with 6.3; that release does have more features necessitating larger /SYS files. But 1, 2, 3, 10, and 12 should be all that are needed unless you are running JCL. You may be able to squeeze in with 6.3.1 as the protection code for the user ID number has been removed. And yes, there are better database programs - our LB to mention one. I suggest you run Pfs form a slimmed down system disk.

Incidentally, you may want to get in touch with the user group in Great Britain. That's NATGUG, 1 Barrowsfield, Sanderstead, Surrey CR2 9EA.

---

## Model I display problem

---

**Fm Jerry Sumrall:** I recently recovered two TRS-80 Model I's. One of the two only displays odd ASCII values. When it tries to display an even value it rounds up to the nearest odd (i.e. r comes out s). After writing a program (10 print" huh?" goto 10) I have determined that the computer is interpreting keystrokes correctly, just not displaying them correctly. Is there anyone who knows how to remedy this problem?

**Fm Fred Oberding:** Jerry, if in addition to "r" appearing as "s", b - c, p - q, @ - a, d - e, etc, then bit 0 is stuck high. You need to replace Z-48, which is a 2102 video RAM chip. It is best to cut the chip out of the board by clipping the leads as close to the chip as possible, leaving the 16 legs still in the board. Then heat each one individually with a low wattage soldering iron, and pull them out with a pair of needle nose pliers. Replace with a socket and new 2102.



## MC/QuickC & void

Fm Dave Krebs to MISOSYS, Inc: Roy, I finally got back to MC (and QuickC) to learn the language. Actually, I'm taking 'Intro to C' at the local community college.

The emphasis is writing code that is ANSI compatible.. To that end, the instructor prefers that we make function prototype declarations, including the function type and type variables passed.

I'm having trouble with the VOID type. When I code MC:

```
void main(void)          /* no good
- preferred by instructor */
void main()              /* OK */
main()                   /* OK */
```

All three of these are OK in QuickC (the compiler of choice in the computer lab in school.

Am I missing an option in MC ... I tried 'ansi.h' and can't find reference to the 'void' type in your 1.6 update docs.

Fm MISOSYS, Inc: The easy answer is that you are confusing the term *void* with the term *prototype*. MC supports the use of the VOID type, but MC does not support prototyping. As an aside, the "function prototype" is in the draft ANSI standard; it is used to pre-declare not only the *type* of a function (i.e. int, char \*, etc.) but also the type of its arguments. Prior to the introduction of prototypes, you were required to predeclare the type of the function prior to its use if it returned other than an int; in fact, it was a good idea to predeclare every function. Compilers which utilize function prototypes can then ensure that all references to the prototyped

function specify the same argument types as the prototype declaration. Your first example, although it used void, was a prototype. You can specify the type of a function as "void" with MC, but you cannot specify the type of its arguments. I suggest that you utilize a conditional for prototyping of the following form. Some header file can have:

```
#ifdef LDOS
#undef NEEDPROTOS
#endif
#ifdef MSDOS
#define NEEDPROTOS
#endif
```

Then your module header files, or wherever you are entering your prototypes could use a:

```
#ifdef NEEDPROTOS
type func(type, type, ...); /*
for MSDOS */
#else
type func(); /* for LDOS */
#endif
```

## TRSCROSS and SuperScripsit

Fm Joyce V. Bateman, Roselle Park, NJ: Gentlemen, I have purchased a 1000 SL/2 and was advised to purchase TRSCROSS also since I had considerable copy from my Model 4, in Superscripts that needed to be converted to the new computer.

Being new to the machine and system I was very nervous. In the beginning I converted some of my files to ASCII, for when I transferred without it seemed to contain a lot of junk. But when some of my files were too large to convert, I studied a little further and found TRSCROSS has the ability to choose any of five methods. Well, I tried SUPERS since I have Varsity scripsit, and find no matter what I tell the computer to use, it transfers it in ASCII and I still have to convert. I don't mind if

there is no other way, but I bought TRSCROSS just to make conversions easier and wonder what is the problem?

Any information or advice you can give me will be appreciated.

Fm MISOSYS, Inc: Joyce, There are two distinct methods of dealing with the conversion of SuperScripsit document files from a TRS-80 disk to the PC using TRSCROSS. First, assuming that your SS document is in SS format (i.e. not saved in ASCII), you can directly convert it to ASCII by specifying the *mode* as SUPRES. Note that it is absolutely essential that you specify the mode as SUPRES; if you don't, you will get gobbledygook as a Superscripts document file is not plain ASCII text! Second, you can convert the SS document to a Document Content Architecture - Revisable Form Text (DCA-RFT) format which will retain most formatting specifications. You can use the DCA-RFT conversion if you have an MSDOS utility which can convert from DCA-RFT to Varsity Scripsit. Most *major* word processors have such a utility; some directly import DCA-RFT. To convert to DCA-RFT, invoke TRSCROSS with the command:

### TRSCROSS /RF

which is covered on page 6 of the TRSCROSS manual, as well as specify the mode as SUPRES.

If you have no means of handling DCA-RFT files, then you can only use ASCII conversion. In either way, you must identify SuperScripsit document files to TRSCROSS by explicitly changing the mode to SUPRES.

On the slim chance that Varsity Scripsit uses the identical document format of SuperScripsit (which I doubt), then specify the document file as binary.

## Help TRSDOS - IBM

**Fm Bill Fishman to MISOSYS, Inc:** I have to move Scripsit files from a Model 4 to a PC that takes 3-1/2 floppies, so I can't just read the TRS-DOS disks. I figure I should be able to use a null-modem cable to connect the machines. Can anyone with experience let me in on what happens, and what command to use to get TRS-DOS to write out to the port (and the PC to read in - I'm a Mac person). Is there some version number I should look for on Scripsit to see if TRSCROSS will work?

**Fm MISOSYS, Inc:** If you are referring to SuperScripsit, then TRSCROSS will directly read SS document files and can convert them to either ASCII files or DCA-RFT files (importable to most major PC word processors). If you are referring to plain Scripsit, those are essentially ASCII files with high-bit set on certain control codes. TRSCROSS will mask the high bit and convert the controls to standard control codes. You should be able to easily use the result in any PC word processor.

You can connect a 3.5" disk drive to a TRS-80. TRSCROSS can handle the TRS-80 720K format 3.5" disk on a PC's 720K 3.5" disk drive but not a 1.44M disk drive.

## TRSCROSS & BASIC translation

**Fm Joe Biagiotti:** I just bought TRSCROSS for a specific program that was on a Model 3. All went well in the transferring and translating of the code, the only thing are seven statements that I need to know what they specifically do:

```
SYSTEM"B", "OFF"
SYSTEM"B", "ON"
SYSTEM"I"
PEEK (16425)
PEEK (14312)
POKE 16425, 2
POKE 16916, 0  <- and ",3" and
               ",5"
```

Beside some minor PRINT @ errors and line Truncated problems everything else is great. Any help would be appreciated. Thanks for any future help I receive!

**Fm MISOSYS, Inc:** The 16425 address references the printer line counter maintained by the DOS. The 14312 the line printer port address (37E8H); some programs peeked that to check on the printer readiness. The 16916(4214H) is a TRSDOS 1.3 poke to set the number of video lines to restrict from scrolling. You'll have to make some adaptation to MS-DOS BASIC operation. The "SYSTEM" commands seem to be a translation of the Model III "CMD" command. TRSDOS 1.3 used the "B",xxx to enable or disable the BREAK key. The SYSTEM"I" was used to exit and return a command to the DOS.

## RS Hard Drive Cable

**Fm Robert Hengstebeck:** Roy, I just got a surplus Navy 5 Meg HD for the Model IV, that I have. It didn't have the cable to hook it up to the computer, so I dropped it off at the local Radio Shack. I am having the repair center check it out for me. If it checks out alright, would I be able to order a ribbon cable for the HD from you. If it flunks, would I be able to scrouge up, the parts, or a new and bigger bubble or what ever else would be needed. I am sort of hoping that my \$30 + (repair look-see) gamble will pay off for me.

**Fm MISOSYS, Inc:** My price list shows that I sell a RS HD Model III/4 host cable for \$20 +\$3 S&H. Order part # H-HD-CT4.

## SUPERSCRIPSIT PRT DRIVER

**Fm Robert W. Via to MISOSYS, Inc:** Where can I find a print driver for Superscripsit that will work with an IBM printer? I think it's similar to an Epson MX-80. I have a Model III manual but there are differences. Do you know where I can get a copy of the Model 4 manual?

**Fm MISOSYS, Inc:** MISOSYS has a driver called PowerDriver+(P-50-200). Price is \$17.95 + \$3S&H. You should be able to order a used Model 4 manual from Pacific Computer Exchange.

USED  
TRSDOS

USED  
XENIX

## RADIO SHACK TANDY OWNERS!

Find the computer equipment that TANDY no longer sells.

**PACIFIC COMPUTER EXCHANGE**  
buys and sells *used* TANDY

TRSDOS  
XENIX  
MSDOS  
COMPUTERS &  
PERIPHERALS

We sell everything from Model 3's and 4's to Tandy 6000's, 1000's to 5000's, Laptops, and all the printers and hard disks to go with them. If we don't have it in stock, we will do our best to find it for you. We have the largest data base of *used* Radio Shack equipment to draw from. All equipment comes with warranty.

## PACIFIC COMPUTER EXCHANGE

The One Source For  
*Used* Tandy Computers

1031 S.E. Mill, Suite B  
Portland, Oregon 97214  
(503) 236-2949

# Inside TMQ

## FILLOW: Fill System Low Memory

by Jonathan Armstrong  
208, 2973 Pembina  
Winnipeg, MB R3T 2H5  
CANADA  
205-261-0142

FILLOW is a small assembler program (here assembled using MRAS) using SuperVisor Calls (SVC's) to fool the operating system into thinking that LOMEM is full. This forces subsequent filters, I/O routines, etc. to be placed in High Memory, so that later, when LOMEM is restored, space is available for a utility (eg. DDUTY) which must occupy LOMEM, but must be installed after the other routines (eg. after a SYSGEN).

In order to provide correct functioning of MemDir and other utilities, a header must be placed in LOMEM to complete the chain of address pointers. To simplify the routine, this header is hard-coded into FILLOW, and is simply copied into LOMEM. Part of it (the name) is first used to determine whether FILLOW is already present, in which case it is removed rather than installed - this avoids the need to scan the command line for a parameter. The result is that if FILLOW is called repeatedly, it alternately installs and removes itself (actually, only the header), occupying or releasing whatever amount of LOMEM is not occupied by other routines.

As described in *The Programmer's Guide*, the LOMEM pointer (DVRHI\$) is located immediately below the Keyboard Input Device Control Block. This is the first DCB in the system, without which it would be pretty hard to get it to do anything, so

I have not bothered to check the error return from @GTDCB after searching for \*KI - if it is not found, something is seriously wrong somewhere! I have however set the error code before returning from FILLOW, though this will only be reported if it is invoked from JCL. The only error is due to insufficient space, and since the header only requires 15 bytes, if there is not enough space to install it, there is little point in doing so.

Note that the location TEMPLO initially contains 0, as is required by @GTMOD to terminate the name string. This is modified during the install procedure, but only in the executing copy in the User Programming Region (it is then copied to LOMEM as part of the header, but is never used there - it is the location normally used to contain a module's DCB pointer). The next time FILLOW is invoked, a fresh copy will be loaded from disk, so the value will again be 0. The reason for doing it this way, rather than using the stack to hold the temporary value (as is done in the 'remove' procedure) is the conditional RETURN statement at X'303B' - the stack must be empty (except for the return address, of course) at this point.



```

;FILL - LOMEM fill routine for LS-DOS 6.3.1 (all 6.x.x?)
;Alternately installs and removes itself (header only)
@GTD CB EQU 82 ;SVC #
@GTM OD EQU 83 ;SVC #
SYSRES EQU 1300H ;First byte above LOMEM
SVC MACRO #SVC ;SuperVisor Call code:
LD A,#SVC
RST 40
ENDM
ORG 3000H ;User Programming Region
ENTRY: JR BEGIN ;Module Header:
DW SYSRES-1 ;Last byte of LOMEM
DB TEMPLO-FILLNM;Length of name
FILLNM: DB 'FILL' ;Name
TEMPLO: DW $-$ ;Temporary pointer storage
DW 0 ;System variable (not used)
BEGIN: LD DE,FILLNM ;Pointer to name
SVC @GTM OD ;Find module by name
JR NZ,INSTAL ;FILL exists?
REMOVE: PUSH HL ;Yes - save address
LD DE,'IK' ; (Search for *KI):
SVC @GTD CB ; Locate LOMEM pointer
DEC HL ; HL <= DVRHI$.HI
POP DE ; DE <= FILL address
LD (HL),D ; Update pointer
DEC HL ; (remove FILL)
LD (HL),E
LD HL,0 ; Zero error code
RET ; EXIT
INSTAL: LD DE,'IK' ;No - locate LOMEM
pointer:
SVC @GTD CB ; (Search for *KI)
DEC HL ; HL <= DVRHI$.HI
LD D,(HL) ;DE <= DVRHI$ value
DEC HL
LD E,(HL)
EX DE,HL ;DE <= pointer to value
LD (TEMPLO),HL ;Save DVRHI$ value
XOR A ;Clear carry flag
LD BC,ENTRY-BEGIN+SYSRES ;Length of LOMEM
space
SBC HL,BC ;Room enough?
RET NC ;No - ABORT, HL<>0 (error)
LD HL,SYSRES ;Yes - value to fill LOMEM
EX DE,HL ;HL <= DVRHI$.LO
LD (HL),E ;Update DVRHI$ value
INC HL
LD (HL),D
LD DE,(TEMPLO) ;retrieve old DVRHI$
LD HL,ENTRY ;Header current address
LD BC,BEGIN-ENTRY ;Header length
LDIR ;Copy header
LD HL,0 ;Zero error code
RET ;EXIT
END ENTRY

```

## Internal HD for 4P

John Cerul  
2551 Caves Highway  
Cave Junction, OR 97523-9779

Well, I'm now the proud owner of a Hard Drive Model 4P. I got all the parts for the MISOSYS hard drive installed inside the box. It wasn't too difficult, just took a little studying and trial fitting. I'll give a listing of the major steps I followed. I've intentionally left out some detail since on a job like this each individual will have his personal preferences as to how to proceed. This is not a job for the novice but should be fairly easy for a reasonably qualified hardware hacker. The best part was that the thing worked on the first try, after I got through "shoe-horning" it into the case.

I really like the MSCSI software package. So far no surprises and has operated "seamlessly". I especially like the clock interface. I had (still have on two other machines) been using the Smartwatch and was satisfied with it, but like yours even better. Enjoy not having to use the 303 bytes of high memory to update the system date and time.

Noted one mistake in the MSCSI Operator manual though - page 44, lower part of the page. - "On the last Sunday in April the time increments.....". Should read first Sunday. I wouldn't have noticed it myself except I was having trouble sleeping at 3:00 AM on April 7th so was reading the manual and remembered that I needed to advance my wristwatch. After advancing my watch I then had to get up and check the clock in the 4P, it had rolled over. [editor's note: the documentation on the DS1287 clock chip provided by Dallas Semiconductor states, "The Daylight Savings Enable bit is a read/write bit which enables two special updates when DSE is set to one. On the last Sunday in April the time increments from 1:59:59 AM to



3:00:00 AM. On the last Sunday in October when the time first reaches 1:59:59 AM it changes to 1:00:00 AM. These special updates do not occur when the DSE bit is a zero. This bit is not affected by internal functions or RESET."

Also noted that if you don't like being scared - don't power up with the Boot disk in the machine if you've parked the hard drive. It takes too long for the hard drive to step out and you'll get a string of "Error 11H"s. It will however eventually make it to the LS-DOS Ready prompt but it is disconcerting to say the least. (ERROR6/CMD when run after the above occurrence says "Error 04H Device not ready or not selected").

[editor's note: M.A.D. software has a replacement ROM for the Model 4P which allows booting directly off of the hard drive. Contact them at PO Box 331323, Ft Worth, TX 76163]

1. Remove the case; then the pan with the mother board and the keyboard.

2. Remove the Floppy drive "cage" and the drive cable. Also remove the power supply and the rear panel.

3. Remove the modem power connector, cut the wires off the connector, toss the connector.

4. Remove the modem guides and the angle bracket that bisects the compartment that the modem is normally mounted in. I found that the easiest way to get the angle bracket out was to use a stout but sharp knife blade between the bracket and the pan. By carefully but forcefully prying/pushing the blade through, the three spot welds will "pop" and will just take a little cleaning and smoothing.

5. Fit the Xebec driver in the modem/high-res board opening, so the component side will face down towards the motherboard with the power connector (P1) and J1 and J2 on the drive side. The 50 pin header (P2) will be on the right if you are facing the rear of the computer. You'll

probably have to "shave" the sides of the board to get it to fit into the opening. (It didn't take much though, I got by with a few swipes with fine grade sandpaper).

If the Adaptec board is the same size as the Xebec it won't fit though because of the card-edge drive connector. [editor's note: the adaptec and Xebec controller boards are the same size. All "S1410"-type controllers have an identical form factor.]

6. Trial fit/locate the host adaptor on the backplate inside the power supply/video compartment with the power connector (P3) toward the blower and the 50 pin header (P4) up, with the component side into the power supply compartment.

Carefully now, it has to be positioned to clear the power supply on top, the blower on the left, and yet not be so close to the bottom pan that the two 50 conductor flat cables are cramped were they have to pass over the pan edge. Then too, the board has to be spaced away from the back plate enough that when the machine is put back together the handle mounting bolts don't pierce the board.

7. I used 6-32 nuts and bolts to mount both boards, but did install acetate insulating sheeting under the boards, just to be sure, in case my spacing was not quite enough.

8. Because of the way that the power connector on the Xebec board faces I had to solder the power wires to the base of the socket (P1) on the board. I then routed the 3 wires out through the hole left by removal of the modem power socket. After splicing these wires and the pigtail for the host adaptor to the wires that originally went to the modem power connector, I realized that I could have put a connector on the harness so that if I have to remove the Xebec board later I would not have to unsolder the power leads. Oh well, next mod. By the way the WHITE wire that went to the modem power connector is -12 volts and is not needed for the hard drive installation.

9. Mount up the hard drive and the floppy

that you're going to use in the cage and reinstall them. I used a Tandy 3 1/2" and the Kalok 320 from Misosys and mounted them with the hard drive in the physical drive 0 slot and the 3 1/2" in the physical drive 1 slot. The power harness fit OK this way so I didn't have to use extenders.

10. Manufactured a new cable for the floppy drive with only one 34 pin edgcard connector to the floppy drive and extended past the header connector terminating in a KEL-AM male connector that is mounted on the back panel so that I can access a standard 40 track disk.

11. Shortened the 34 conductor and the 20 conductor hard-drive cables and routed them so they ran up inside the floppy cable. I had to use the old "nibbler" to deepen the opening in the pan for the floppy cable.

12. Manufactured/routed the two 50 conductor cables. One from P2 on the Xebec board to P4 on the host interface. This one needs a 90 degree fold in it since the headers are perpendicular to each other. The second cable runs from P2 of the host adaptor to a 50 pin cardedge connector to the expansion port on the mother board. This one needs a "double foldover bend" so that the cable is reversed because the two connectors are exactly reversed. ie; pin one on the motherboard is on the right (from the rear) while pin one on the host adaptor after mounting is on the left.

13. Loosely fasten everything together after connecting all connectors and take it for a test drive. Everything ran fine, and as documented.

14. Put everything back together in a proper manner. Still works. Hooray.



By J.F.R. "Frank" Slinkman  
 1511 Old Compton Road  
 Richmond, Va. 23233  
 Phone: 804/741-0205  
 CIS ID# 72411,650

## How to "Roll Your Own" on the XLR8er

The Z180/HD64180 microprocessor in the XLR8er board for the Model 4 has several very powerful features which have not yet been well supported by software. For brevity, I will henceforth refer to the chip as the "Z180."

Let me give you a quick rundown of these powerful on-chip features:

**ASCII** — Asynchronous Serial Communications Interface. According to Hitachi's HD64180 Hardware Manual, there are two ASCII channels capable of talking to UARTs, which means interrupt driven communications under direct CPU control are easily implemented. However, this feature is not supported on the basic XLR8er board, but can be as an option by plugging an extra chip in the board, if I interpret my XLR8er manual correctly.

**CSI/O** — Clocked Serial I/O Port. This enables the Z180 to easily communicate with other microprocessors, or to external devices. I believe this function also requires the extra chip.

**DMA** — Direct Memory Access. There are two DMA channels, 0 and 1, with channel 0 able to manage both in-memory and memory-to-and-from-I/O port transfers, and channel 1 limited to memory-to-and-from-I/O. Channel 1 transfers have been utilized in a couple of graphics applications as a fast way to move data from RAM to the Radio Shack and/or Micro-Labs hi-res graphics boards (see "Using XLR8er RAM as Graphics Video RAM" TMQ IV.iii).

**Internal Interrupts.** My article, "It's Rude NOT to Interrupt," in the last TMQ (V.iii) discussed how to use these.

**MMU** — the Memory Management Unit.

```

; NEWINSTR/ASM - 10-Jan-91
;
; by J.F.R. "Frank" Slinkman, 1511 Old Compton Road,
; Richmond, Va. 23233. 804/741-0205. CIS ID# 72411,650
;
; Routine to demonstrate use of Z180/HD64180 TRAP
; interrupt and DMA Channel 0 memory-to-memory transfers.
; Creates two new instructions; LDIRF to replace LDIR;
; and LDDRF to replace LDDR.
; This code is released to the public domain.
;
; SVCs used in this program
@EXITREQ 16H
@HIGH$ EQU 64H
; Z180/HD64180 internal ports used in this program
SAROLEQU 20H
SAROEQU 21H
SAROEQU 22H
DAROLEQU 23H
DAROEQU 24H
DAROEQU 25H
BCROLEQU 26H
BCROEQU 27H
DSTATEQU 30H
DMODEEQU 31H
DCNTLEQU 32H
ITC EQU 34H
;
*LIST OFF
; Michel Houde's macros to define IN0 & OUT0 instructions
; OPTION NC ;don't list unused conditionals
; in0 g,(m) ;for this instruction, use
; in0 port,reg ;this form (no parentheses)
IN0 MACRO #G,#O
DB 0EDH
IFEQ$ #G,B
DB 00H
DB #0
EXITM
ENDIF
IFEQ$ #G,C
DB 08H
DB #0
EXITM
ENDIF
IFEQ$ #G,D
DB 10H
DB #0
EXITM
ENDIF
IFEQ$ #G,E
DB 18H
DB #0
EXITM
ENDIF

```

```

IFEQ$ #G,H
DB 20H
DB #0
EXITM
ENDIF
IFEQ$ #G,L
DB 28H
DB #0
EXITM
ENDIF
IFEQ$ #G,A
DB 38H
DB #0
EXITM
ENDIF
ERR 'Bad reg'
ENDM

;
; out0 (m),g ;for this instruction, use
; out0 port,reg ;this form (no parentheses)
OUT0 MACRO #O,#G
DB 0EDH
IFEQ$ #G,B
DB 01H
DB #0
EXITM
ENDIF
IFEQ$ #G,C
DB 09H
DB #0
EXITM
ENDIF
IFEQ$ #G,D
DB 11H
DB #0
EXITM
ENDIF
IFEQ$ #G,E
DB 19H
DB #0
EXITM
ENDIF
IFEQ$ #G,H
DB 21H
DB #0
EXITM
ENDIF
IFEQ$ #G,L
DB 29H
DB #0
EXITM
ENDIF
IFEQ$ #G,A
DB 39H
DB #0
EXITM
ENDIF
ERR 'Bad reg'
ENDM

;
SVC MACRO #O
LD A,#O
RST 28H
ENDM

```

This is about the only feature well supported in software, as it MUST be used to implement @BANK modifications in such applications as Michel Houde's XLR8 module and its progeny. It's use was also demonstrated in my article, "Fast In Memory Sort Using All XLR8er RAM" (TMQ IV.iii).

**PRT** — Programmable Reloadable Timer. There are two PRT channels, which were discussed in my article, "It's Rude NOT to Interrupt" (TMQ V.iii).

In this article I will cover the use of the one internal interrupt not covered in "It's Rude NOT to Interrupt," and the use of DMA channel 0 transfers as a superior substitute for the standard LDIR and LDDR instructions. I'm referring to the internal TRAP interrupt, which is generated anytime the Z180 encounters an illegal opcode. The TRAP interrupt vectors to x'0000', the same address as RESET.

The TRAP interrupt has two important potential uses. One is to improve program reliability. It's possible, for example, to write a routine to display or print out an error report detailing the location and nature of any illegal opcodes encountered before terminating the program. It's other use is to create your own custom set of opcodes.

Some operating systems will not run with the Z180 because they make use of undocumented Z80 instructions which access 8-bit halves of the 16-bit IX and IY registers. Using the techniques in the accompanying program listing, it would be a fairly simple matter to TRAP these instructions, duplicate their functions, and make them "legal" again.

The accompanying program, NEWINSTR/ASM, uses the TRAP interrupt and memory-to-memory DMA channel 0 transfers to create two new instructions: LDIRF to duplicate the function of LDIR; and LDDRF, to duplicate the function of LDDR. Please note that the new commands can only be used in the base 128K, and not to/from XLR8er memory?

These two instructions are superior because they execute RAM transfers much faster than the standard instructions. Because of the insertion of refresh cycles, even running the Z180 in it's fastest mode (0,1,80; REFW=0), LDIR and LDDR require an average of 14.35 T-states per byte transferred. LDIRF and LDDRF only require 6T! At the standard Houde' settings (1,1,40; REFW=1), LDIR and LDDR take an average of 15.05T per byte versus 7T for DMA.

However, because LDIRF and LDDRF involve about 400T of program overhead, it works out that the cutoff is at about 50 bytes. If you need to transfer fewer than 50 bytes, it'll be as fast or faster to use LDIR or LDDR. If you need to transfer 50 or more bytes, it's faster to use LDIRF or LDDRF.

For example, to transfer 256 bytes, the new instructions are about 70% faster. To transfer a full video ram screen of 1,920 bytes, the new instructions are over twice as fast.

The new instructions are also superior because they can directly address the 256K XLR8er RAM (or all 512K if you have Richard King's hardware mod) without bank switching.

I chose the opcodes ED C0 for LDIRF and ED C8 for LDDRF, because they are only one byte different from their Z80 counterparts (ED B0 for LDIR and ED B8 for LDDR), and there are no legal ED Cx opcodes.

Now lets "walk through" the program listing. First are a series of equates defining the SVCs used in the program, and the Z180 internal registers the program uses.

Next, for those who did not get DiskNotes 3.2 with Michel's HITACHI/ASM on it, are two macros to facilitate coding the Z180 instructions IN0 g,(m) and OUT0 (m),g. Next is a short macro to facilitate coding LS-DOS SVCs.

The last two macros define the two in-

```
*LIST ON
;
; Macros to define the two new opcodes
LDIRF MACRO
    DB    0EDH
    DB    0C0H
    ENDM
;
LDDRF MACRO
    DB    0EDH
    DB    0C8H
    ENDM
;
;
    ORG    0000H
    JP     TRAP ;vector for RESET and TRAP
;
    ORG    3000H
;
TRAP PUSH    AF ;save AF reg
    IN0      A,ITC ;get trap status
    AND      0C0H ;isolate TRAP and IFO bits
    CP       80H ;is this a TRAPPED 2nd opcode byte?
    JR       Z,TRP10 ;go if so. falls thru if RESET or
                    ; TRAPPED 3rd opcode byte
;
RESETDI      ;duplicate orig code @ x'0000'
    LD       A,1
    JP       3 ; and go to instruction after it
;
TRP10 LD      A,1 ;here if TRAPPED 2nd opcode byte
    OUT0     ITC,A ;reset TRAP bit (turns off interrupt)
    LD       (SAVHL),HL ;save HL entry value
    POP      HL ;HL = AF entry value
    LD       (SAVAF),HL ;store orig AF
    LD       HL,$-$ ;restore HL entry value
SAVHLEQU     $-2
    EX       (SP),HL ;save HL, get ptr to illegal byte
    DEC      HL
    LD       A,(HL) ;p/u opcode byte B4 "illegal" one
    CP       0EDH ;could it be LDIRF or LDDRF?
    JR       NZ,RESET ;go if not
    INC      HL
    LD       A,(HL) ;p/u "illegal" opcode byte
    INC      HL ;calculate RET address
    EX       (SP),HL ;restore HL, RET address on stack
    CP       0C8H ;is "illegal" byte code for LDDRF?
    JR       Z,@LDDR ;go if so
    CP       0C0H ;is it our code for LDIRF?
    JR       NZ,RESET ;if not, it's no good
;
@LDIRLD      A,2 ;increment source and destination
    JR       TRP20 ; addresses, set burst mode
;
@LDDRLD      A,16H ;decrement source and destination
                    ; addresses, set burst mode
;
TRP20 OUT0    D,MODE,A ;set DMA mode of operation
;
    OUT0     SAR0L,L ;load DMA channel 0 source
    OUT0     SAR0H,H ; address registers w/HL
;
    OUT0     DAR0L,E ;load DMA ch 0 destination
    OUT0     DAR0H,D ; address registers w/DE
;
```



```

OUTO BCR0L,C ;load DMA channel 0 byte
OUTO BCR0H,B ; counter registers w/DE
;
LD A,40H ;start DMA channel 0 burst
OUTO DSTAT,A ; mode transfer
;
LD HL,$-$ ;HL = entry value of AF
SAVAF EQU $-2
LD A,0E9H
AND L ;reset H, P/V and N flags
LD L,A
PUSH HL
POP AF ;AF now has same value as if an
; LDIR or LDDR had been done
;
INO L,SAR0L ;give HL, DE & BC same values
INO H,SAR0H ;as if this was an LDIR or
INO E,DAR0L ;LDDR instruction
INO D,DAR0H
LD BC,0
;
OUTO SAR0B,B ;reset address line bits
OUTO DAR0B,B ; A16-A18 for next use
RET
;
; Make sure DMA channel 0 references only base 64K.
INIT_HI_LINES
XOR A
OUTO SAR0B,A
OUTO DAR0B,A
RET
;
BEGINCALL INIT_HI_LINES
LD B,0
LD HL,0
SVC @HIGH$ ;get HIGH$
PUSH HL ;and save it
LD DE,3100H
OR A
SBC HL,DE ;calc # of bytes for xfer
PUSH HL ;and save it
LD B,H
LD C,L ;# of bytes to BC
EX DE,HL ;HL = 3100H
LD DE,3101H
LD (HL),0AAH
LDIRF ;fill 3100H->HIGH$ w/0AAH
;
POP BC ;# of bytes for xfer
POP HL ;HIGH$
LD D,H
LD E,L
DEC DE ;DE = HIGH$-1
LD (HL),0
LDDRf ;fill 3100H<-HIGH$ w/00H
;
LD HL,RESET ;restore original code
LD DE,0 ; @ 0000H
LD BC,3
LDIR
LD HL,0 ;and exit
SVC @EXIT
;
END BEGIN

```

structions being created, LDIRF and LDDRf.

Now we get to some actual program code. The first thing we do is stuff our RESET/TRAP vector into x'0000', which is simply an absolute jump to TRAP.

At TRAP, after PUSHing the AF register, we read the ITC internal register. This register contains bits which flag and describe a TRAP interrupt. If bit 7 = 1, it indicates a TRAP interrupt. If bit 7 = 0, then we must deal with a RESET.

If bit 7 is set, then bit 6 tells us which byte of the opcode was illegal. If bit 6 = 0, it was the second byte. If bit 6 = 1, then it was the third byte.

The only other bit in this register of concern to us is bit 0, which selects the type of interrupts used by the Model 4. This bit is set to 1 on RESET, and should always remain set.

Since both our created instructions have "illegal" 2nd opcode bytes, the values in bits 7 and 6 will be 1 and 0 if one of our instructions triggered a TRAP interrupt. Any other value indicates either a RESET or another kind of opcode error.

Thus we AND the contents of ITC with 0C0H. If the result is not 80H, then we duplicate the two instructions which we over wrote at x'0000' and proceed with RESET processing.

If the value was 80H, the first thing we do is write 01H to ITC to reset the TRAP interrupt flag while keeping bit 0 set.

The next instructions clear the stack while saving the original contents of the AF register in RAM for later retrieval.

The EX (SP),HL instruction right after SAVHL does two things: it saves the entry value of HL on the stack while putting the pointer to the illegal opcode byte in HL.

Since we know the error was in the second byte, we decrement HL to point to the first

byte of the instruction and test to see if it matches the first byte of our created opcodes (0EDH).

If it's not 0EDH, then we know the TRAP interrupt was triggered by some other illegal opcode; so we jump to our code for RESET.

If the value is 0EDH, then we increment HL to point to the "illegal" byte which triggered the interrupt and load it into the accumulator. Since ours are two-byte opcodes, incrementing HL once more causes HL to point to the instruction immediately following ours. This becomes the RETurn address for our routine, assuming we actually are dealing with one of the instructions we are creating. If not, it doesn't matter, because any other value will cause a RESET.

The next EX (SP),HL instruction puts our tentative RETurn address on the stack and restores the original value in HL.

Next we test the "illegal" opcode byte in A. If it's 0C8H, then we know we have a LDDRF instruction. If not, we test for 0C0H. If it's not 0C0H, then we must RESET.

For LDIRF, we load 02H into A, and go to TRP20. For LDDRF, we load 16H into A, and fall through to TRP20.

These are values for the DMODE internal register. This register determines the mode of operation of the DMA channels. Bits 5 and 4 of this register determines how DMA will handle the DMA destination address registers, and bits 3 and 2 determine the manipulation of the source address registers, each according to table I.

Bit 1 determines whether DMA will be done in "burst" or "cycle steal" mode. Burst, as the name implies, means the DMA channel seizes control of the bus, and does not release it until the transfer is complete. No other operations, except NMI processing, can be performed until completion.

Table I

0	0	RAM	increment once for each byte transferred
0	1	RAM	decrement once for each byte transferred
1	0	RAM	keep fixed, no increment or decrement
1	1	I/O	keep fixed — this is a port number.

Cycle steal alternates between DMA control and CPU control. One CPU cycle is performed, then one byte transferred by DMA, alternating until the transfer is complete. Our use of DMA requires burst mode; so bit 1 of DMODE must be set.

To emulate LDIR, we need both the source and destination address to be incremented once for each byte transferred. Thus the only bit we want to set is bit 1, hence the value 02H.

To emulate LDDR, we need both the source and destination addresses to be decremented once for each byte transferred; so we need to set bits 4, 2 and 1, hence the value 16H.

The next three pairs of instructions copy HL to the internal DMA channel 0 source address registers; DE to the destination address registers; and BC to the byte count registers.

The only registers we haven't loaded are the source address and destination address "B" registers (SAR0B and DAR0B). This is intentional. These registers should be set to 00H upon boot up, just as the MMU registers are initiated in the Houde' utilities.

If either instruction is to be used to transfer data to or from XLR8er RAM (physical addresses 40000H-7FFFFH, or any RAM outside the bottom 64K using the Richard King mod), the most significant three bits of the 19-bit address(es) must be written to these registers before the instruction is executed.

Now we initiate the transfer by writing 40H to DSTAT, bit 6 of that register being the enabling bit for DMA channel 0. It is automatically reset upon transfer completion.

Because we want to be able to replace the LDIRs and LDDRs in any existing code with our new LDIRF and LDDRF instructions, and we can't predict what use that code might make of the flags or other registers after the completion of the transfer, we must EXACTLY emulate the LDIR and LDDR instructions. This means we must now give the AF, BC, DE and HL registers the exact values they would have upon completion of an LDIR or LDDR.

First, we tackle the AF register. We do this by loading it into HL, so that H contains the original value of A, and L contains the original flags from F.

We AND the value in L with 0E9H to reset the H, P/V and N flags, while preserving the original state of all other flags, which duplicates the flag actions of LDIR and LDDR. The result is written back to L. PUSHing HL and POPping AF puts the original value in A and the corrected flags in F.

The next four instructions put the correct values in HL and DE by copying the contents of the DMA channel 0 source and destination address registers to them, one byte at a time. The next instruction gives BC the value of 0000H, consistent with either LDIR or LDDR.

Now, to conform to the protocol I have established above, namely that the top three bits of the 19-bit source and destination address registers must reference the base 64K unless altered for a specific block transfer outside that region, 00H is written to SAR0B and DAR0B.

Finally, the RET instruction causes a jump to the return address we calculated previously.

The next code in the listing, at

INIT\_HI\_LINES, is a subroutine to initiate the top three address bits of both the source and destination address registers for DMA channel 0 to zero, to compensate for the fact that they were not initialized upon boot up.

This is done in a subroutine since DEBUG "loses it" if it encounters an IN0 or OUT0 instruction. This way, DEBUG can keep control while stepping through the program starting at BEGIN.

At BEGIN, we call INIT\_HI\_LINES, and then get the HIGH\$ pointer. This is used to set the values for code sequences to first fill all RAM from 3100H to HIGH\$ with 0AAH using LDIRF, and then fill it with 00H using LDDRF.

Finally, we restore the original code at x'0000', and return to LS-DOS.

You may find it interesting to use DEBUG to step through the program starting at BEGIN, then repeat the operation after changing the LDIRF to LDIR and LDDRF to LDDR (by changing ED C0 to ED B0 and ED C8 to ED B8), and note how much faster the new instructions are.

Another interesting test is to step through the program using DEBUG, but stop at x'309F'. Then use DEBUG to change the 0B0H bytes at x'0CEF', x'0D34', x'0D9C' and x'0DAB' to 0C0H. These are the second bytes of LDIR instructions in the \*do driver, which control scroll, clear to end-of-frame, display line and video screen move, respectively.

Then return to LS-DOS and LIST (N) a long ASCII file. The speed increase is quite noticeable, especially if the file is on an ERAM disk. You'll need to hit the reset button after this test, as the easiest way to restore the altered code.

The last article in this series about the Z180, including a new and MUCH faster PEXMEM-type module, will appear in a future issue of *The MISOSYS Quarterly*.



## Profile 4 Plus Printer Codes

by Dave Krebs CIS: 73125,105

FilePro 16, the MS-DOS and UNIX big brother to Profile 4 Plus has the ability to send up to 99 print codes to the printer, up to 99 are defined. After using them for awhile I thought they were pretty handy and even though not available in the Model IV version, maybe I could implement some variation that would work. This procedure will also work on Profile III Plus.

You'll need a disk editor, such as FED or Super Utility for the following procedure.

Using the editor, view sector 00 of your printer format, filename/PRn. You'll probably see your headings in the first sector. If the first line of your heading is centered, the first few characters will be spaces (20H). All you have to do is to replace the required number of spaces with the proper print code for the pitch you want. So, enter the HEX modification mode and change the first two bytes to 1B 14 to change to 16.7 CPI on your Tandy DMPxxx printer.

1B 0F is the code for your Epson compatible printer.

There is one small side effect to this procedure. Your title used to be centered, now it's two spaces to the left of center. To correct this, return to the DEFINE REPORTS creation program of Profile and retrieve your print format. You'll see a graphic representation of your printer codes in the first two bytes. Insert two spaces between the codes and your text. Your title will now be printed centered.

If the first line of your title is left-justified, you can reverse the above procedure. First, use DEFINE REPORTS to move your title the necessary spaces to the right on the first line. Now, use FED to insert the printer codes.

A portion of the first record of LW000000/PR1 with print codes for an Epson FX and elite type is shown below ...

LW000000/PR1										Drive 1	Record	X'0000'									
0123456789ABCDEF	BYTE	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F				
.M	<00>	1B	4D	20	20	20	20	20	20	20	20	20	20	20	20	20	20				
	<10>	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20				
LORAIN WH	<20>	20	20	20	20	20	20	20	4C	4F	52	41	49	4E	20	57	48				
EELMEN	<30>	45	45	4C	4D	45	4E	20	20	20	20	20	20	20	20	20	20				
	<40>	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20				

## GENUINE RADIO SHACK HARD DRIVES

For Models 3 and 4

5 Meg \$175  
10 Meg \$225

15 Meg \$275  
20 Meg \$325

35 Meg \$475

Formatted under LS-DOS 6.x; New MISOSYS RSHARD5/6 driver included.

Do it yourself special, (No bubble) \$125. Includes controller, P/S, fan, cables, driver. Install your own ST-506 bubble, half or full height. Instructions included.

All cables included.  
Add S&H to all  
prices.

Roy T. Beck 213-664-5059  
2153 Cedarhurst Drive  
Los Angeles, CA 90027

All hardware used,  
tested, and  
warranted 90 days.

## BOOT LDOS 5.x directly from a hard drive Model 4 running LS-DOS

**Program and documentation  
by Adam Rubin  
CIS:71320.1052**

Copyright (C) 1990 Adam Rubin. All rights reserved. This file may be redistributed if (1) it is offered only in its original unmodified form, and (2) no charge is made (beyond any actual cost of redistribution).

When running TRSDOS/LS-DOS 6.x, BOOT5 allows you to boot LDOS 5.x directly from your hard disk, without a boot diskette.

The following three files will be available on DiskNotes 5.4:

1. **BOOT5/ASM** - Assembler source code for the program
2. **BOOT5/CMD** - The program itself
3. **BOOT5/DOC** - A documentation file, formatted for printing at 80 columns, 66 lines per page

The three files are contained in BOOT5/ARC which is available on the PCS-49 forum on CompuServe.

### System Requirements:

1. TRS-80 Model 4, 4P, or 4D
2. TRSDOS/LS-DOS 6.2.0 or later
3. LDOS 5.3.0
4. Hard disk with drivers for both operating systems

```
;Program: BOOT5
;Version: 1.0
;Date: December xx, 1990
;Author: Adam Rubin (Internet:
71320.1052@CompuServe.COM)
;
;Hardware: TRS-80 Model 4, 4P, or 4D
;DOS: TRSDOS/LS-DOS 6.2.0 or later, and LDOS 5.3.0
(both required)
;Language: Z80 assembler
;Assembler: Pro-MRAS (Micosys, Inc.)
; Use "-GC" switch to assemble directly to /CMD file
;
;Description: When running DOS 6.x, this program will
boot DOS 5.x directly
; from a hard disk without a boot diskette.
;
;Usage:
; BOOT5 [imagenam] [:]drive
; imagenam (4P only) is A to G, for MODELA/III through
MODELG/III
; drive 1 to 7, drive containing 5.3.0 system disk
;
;Before using this program, please read the documentation
in BOOT5/DOC.
;
;Copyright (C) 1990 Adam Rubin. All rights reserved. This
file may be
;redistributed if (1) it is offered only in its original
unmodified form,
;and (2) no charge is made (beyond any actual cost of re-
distribution).
;
;


---


;
TITLE <BOOT5/ASM>
COM '<Copyright (C) 1990 by Adam Rubin. All rights
reserved.>'
OPTION +NE +NM +SL +WE
;
;Define SVC and ASCII EQUates, SVC macro
*LIST OFF
IF1 ;Define on 1st pass only, to suppress
from symbol list
@CKBRKC EQU 6AH
@CKDRV EQU 21H
@CLS EQU 69H
@DATE EQU 12H
@DSP EQU 02H
@DSPLY EQU 0AH
@EXITEQU 16H
@ERROR EQU 1AH
@FLAGS EQU 65H
@GTDCT EQU 51H
@HIGH$ EQU 64H
@KEY EQU 01H
@LOGOT EQU 0CH
@OPENEQU 3BH
```



```

@RDSEC EQU 31H
@RDSSC EQU 55H
@READ EQU 43H
@TIME EQU 13H
CR EQU 0DH
ETX EQU 03H
LF EQU 0AH
SI EQU 0FH
ENDIF
;
SVC MACRO #NUM
LD A,#NUM
RST 28H
ENDM
*LIST ON
;
ASEG
ORG 3000H
;
;Defaults (X'3000' to X'3004'), can change with command
line patch
DRIVEDB 0 ;Default 5.3 drive, 0=no default or '1'-
'7' (31H-37H)
IMAGEDB 'A' ;Default ROM image 'A'-'G', 'a'-'g' (41H-
47H, 61H-67H)
TIMEFLG DB 0 ;If non-zero, program will not set
5.3 date and time
CRTCF LG DB 0 ;If non-zero, program will init
CRTC for MIII mode
PRESSDB 0 ;If non-zero, program will skip 'Press
any key...'
;
;Program entry point: Set up for switch to Model III mode
STARTLD SP,STACK
PUSH HL ;Save pointer to command line
LD L,C ;Store file name used to invoke
program...
LD H,B ; in case someone renames /CMD
file
LD B,8
LD DE,$PNAME$ ;Where to store file name
$ST1 LD A,(HL)
INC HL
CP '!'
JR Z,$ST1 ;Skip leading '!'
CP '0'
JR C,$ST3 ;Anything below '0' means end of
filename
CP ':'
JR Z,$ST3 ;Also a filename terminator
CP 'a'
JR C,$ST2
AND 0DFH ;Force uppercase
$ST2 LD (DE),A ;Store up to 8 characters
INC DE
DJNZ $ST1
;
;Check current DOS version, machine type, <BREAK> key
$ST3 SVC @FLAGS
LD A,(IY+27) ;DOS version must be at least 6.2
CP 62H
LD HL,$OLDOS$
JP C,ABEND
LD A,(IY+'T'-'A') ;Machine type flag, 04H=4/

```

One logical drive on this hard disk must be configured as drive 0 for LDOS. It must also be accessible under TRSDOS/LS-DOS 6.x, as any drive number. This logical drive will be called "LDOS :0" in this documentation. (If your hard drive is not set up for both operating systems, see your hard disk driver documentation.)

This drive, LDOS :0, must have LDOS SYS0/SYS and CONFIG/SYS files. SYS0/SYS can be moved from your LDOS boot diskette with LDOS's BACKUP utility. CONFIG/SYS can be created by booting with your LDOS boot diskette, and at "LDOS Ready" typing:

SYSTEM (SYSGEN)

**Model 4P only:** This drive must also have a ROM image file, usually MODEL A/III. You can COPY this file from your LDOS boot diskette.

## Using BOOT5

Usage: BOOT5 [imagename] [:]drive

imagename: (4P only) is A to G, for MODEL A/III through MODEL G/III

drive: contains your 5.3.0 system disk, 1 through 7

BOOT5 is run at the 6.x "TRSDOS Ready" or "LS-DOS Ready" prompt. Type

BOOT5:d

where "d" (the colon is optional) is the logical drive that becomes drive 0 under LDOS. Example:

BOOT5:3

BOOT5 loads the necessary files into memory from that drive, and prompts: Press <BREAK> to return to 6.x, or any other key to boot DOS 5.3. If you press the <BREAK> key, BOOT5 will return to DOS 6.x. Press any other key and LDOS will start up, with the date and time copied from 6.x.

If you enter the command line in an incorrect format, or type:

### BOOT5 ?

the program will display the short usage summary above. If BOOT5 encounters any problems, it will display an appropriate error message and return to DOS 6.x.

### Advanced Options

Some of BOOT5's defaults can be changed with the DOS 6.x PATCH utility. Initially, BOOT5 requires a drive number on its command line. However, you can tell BOOT5 what your "usual" drive is with:

PATCH BOOT5 (X'3000'="d")

where "d" (the quotes are required) is the drive number. For example,

PATCH BOOT5 (X'3000'="3")

will change the program to use :3 if no drive is specified. Once this is set, typing

BOOT5

may be sufficient to start up LDOS. You can override your default by specifying a drive on the command line.

Model 4P users can load a ROM image other than MODEL A/III with

BOOT5 i:d

where "i" is "A" to "G" (uppercase or lowercase), corresponding to MODEL A/III through MODEL G/III. (The default is initially MODEL A/III.) The space after "i" and the ":" are optional, so all of the following would be equivalent:

BOOT5 C :6  
BOOT5 c:6  
BOOT5 C6

If you usually load a different ROM image, you can :

PATCH BOOT5 (X'3001'="i")

```

4D, 05H=4P
SUB    04H          ;Any other value is NG
CP     01H+1
LD     HL,$NOTM4$
JP     NC,ABEND
SVC    @CKBRKC      ;Abort if <BREAK> was
pressed
JP     NZ,ABORT
LD     HL,$NAME$    ;Display actual program name
SVC    @DSPLY
LD     HL,$TITLE$   ;Rest of title string
SVC    @DSPLY

;
;Parse command line
POP     HL          ;Pointer to command line
LD      A,(HL)      ;First character can be ...
CP      CR          ;      CR, '0'-'7', ':', ...
JR      Z,$EOPRM    ;      'A'-'Z', 'a'-'z'
CP      '0'
JR      C,$BADPRM
CP      '7'+1
JR      C,$DPRM
CP      ':'
JR      Z,$CPRM
CP      'A'
JR      C,$BADPRM
CP      'Z'+1
JR      C,$IPRM
CP      'a'
JR      C,$BADPRM
CP      'z'+1
JR      NC,$BADPRM  ;If none of these, "bad parameter"

$IPRMLD (IMAGE),A    ;Save ROM image parameter
$IPRM2  INC     HL
LD      A,(HL)
CP      ':'
JR      Z,$IPRM2    ;Next could be some spaces...
CP      CR          ;      then CR, ':', or drive
number
JR      Z,$EOPRM
CP      ':'
JR      NZ,$DPRM
$CPRMINC HL          ;Skip over ':'
LD      A,(HL)
$DPRMLD (DRIVE),A    ;Save drive number
$DPRM2  INC     HL    ;Can only be followed by
spaces and CR
LD      A,(HL)
CP      ':'
JR      Z,$DPRM2
CP      CR
JR      Z,$EOPRM
$BADPRM LD      HL,$USE1$ ;Bad parameter, so display
usage summary
SVC    @DSPLY
LD      HL,$NAME$    ;Use /CMD file name
SVC    @DSPLY
LD      HL,$USE2$
SVC    @DSPLY
JP     ABORT

;
;Check command line entries or defaults
$EOPRM  LD      A,(IMAGE) ;We now have all parameters

```

```

AND    0DFH          ;Force ROM image ID to uppercase
LD      (IMAGE),A
LD      ($IMGLTR),A  ;Insert into filename
SUB     'A'
CP      'G'-'A'+1    ;If not 'A'-'G', then 'Bad Image'
LD      HL,$BADIM$
JP      NC,ABEND
LD      A,(DRIVE)    ;Drive must be '1'-'7'
LD      ($NOT5D$),A  ;Put into error message
AND     A
JR      Z,$BADPRM    ;If unpatched default, show usage
SUB     '1'
CP      '7'-'1'+1
LD      C,A          ;Drive number minus one into C
LD      A,20H        ;DOS: 'Illegal Drive Number'
JP      NC,ERROR
INC     C            ;C now has drive number
SVC     @CKDRV       ;Drive enabled and contains disk?
LD      A,20H
JP      NZ,ERROR
;
;Read 5.3 sectors and files into memory
LD      DE,0002H     ;Cylinder 0, sector 2
LD      HL,SECTBUF
SVC     @RDSEC       ;Read sector into our buffer
JP      NZ,ERROR
LD      A,(HL)       ;DOS version on this disk, set by
FORMAT, ...
CP      53H          ;      updated when SYS0 moved to
this disk
$BADVER LD      HL,$NOT53$ ;Abort if not DOS 5.3
JP      NZ,ABEND
LD      A,(TIMEFLG)  ;If flag is zero, set 5.3 date and
time
AND     A
JR      NZ,$RD1
LD      (_DT1),A     ;Enable MIII routine to insert
date and time
CPL                     ;Set SYSTEM (TIME=NO) so 5.3 won't
ask user
LD      (SECTBUF+0C3H),A
$RD1 SVC     @GTDCT
LD      D,(IY+9)     ;Directory cylinder
LD      E,0          ;Sector 0
LD      HL,SECTBUF+100H
SVC     @RDSSC       ;Read sector into our buffer
JP      NZ,ERROR
LD      HL,0         ;Get current HIGH$
LD      B,L
SVC     @HIGH$
LD      A,H
LD      ($RDHI+1),A  ;Make sure buffer doesn't go past
HIGH$
LD      HL,FILEBUF
SVC     @FLAGS
LD      A,(IY+'T'-'A') ;Machine type flag
SUB     5
JR      NZ,$RD2      ;Jump if not 4P
LD      (_LDIMG+1),A ;Enable "load ROM image" routine
LD      (IMGBGN),HL  ;Store buffer address of ROM image
LD      DE,$IMGNAM
CALL    RDFILE       ;Read MODEL?/III into buffer
$RD2 LD      (SYSBGN),HL ;Start (in buffer) of 5.3 SYS0/SYS
LD      DE,$SYSNAM

```

where "i" (including the quotes) is "A" to "G", either uppercase or lowercase. For example, to have MODEL C/III be your default,

PATCH BOOT5 (X'3001'="C")

and BOOT5 will then look for MODEL C/III. You can still load another image by specifying it on the command line.

Any ROM image used must be on LDOS :0. The "official" image has always been called MODEL A/III, so you will only have others if you created them yourself.

If you do NOT want BOOT5 to start LDOS with the date and time (if any) from DOS 6.x,

PATCH BOOT5 (X'3002'=FF)

and LDOS will follow whatever was set with the LDOS commands "SYSTEM (DATE)" and "SYSTEM (TIME)". There is no corresponding command line option, but you can undo the effect of the preceding patch with

PATCH BOOT5 (X'3002'=00).

If you have replaced your Model 4/4P/4D's preprogrammed 68045 CRT controller with a programmable 6845, it will need to be initialized for the Model III display configuration. BOOT5 can handle that with two patches, described in BOOT5/ASM just before the lines labelled CRTCTFLG (around line 70) and \$CRTCTB (near the end). Use PATCH, or any other suitable utility, to make the necessary changes to BOOT5/CMD.

## Notes

If there is an AUTO command on your LDOS boot diskette, you will need to add it to your hard disk. Under LDOS, use

AUTO ? :d

(where "d" is the drive with your boot diskette) to display your AUTO command. If the response is not a blank line,



use:

AUTO command

(where "command" is the response to the previous step) to store it on your hard disk. Make sure that all files used by this AUTO command are on your hard disk.

Your LDOS AUTO command may be changed or removed at any time. To fix a runaway AUTO command, reboot with your LDOS boot diskette.

You may change your LDOS configuration at any time. Save your new configuration file on both your hard disk and boot diskette with:

```
SYSTEM (SYSGEN)
SYSTEM (SYSGEN,DRIVE=d)
```

where "d" is the drive with your LDOS boot diskette.

When BOOT5 starts up LDOS, it does not allow you to use the <D> key to enter DEBUG, or <CLEAR> to prevent CONFIG/SYS from being loaded. To use either option, boot with your LDOS boot diskette. (Both of these suppress CONFIG/SYS, which LDOS needs to access the hard disk.)

While booting, the <ENTER> and <RIGHT ARROW> keys may be held down, as documented in your LDOS manual. Once you are at "LDOS Ready", of course, DEBUG may be used as desired.

Notice: BOOT5 appears to operate as documented, but I accept no responsibility for its use. Use this program at your own risk.

TRS-80 and TRSDOS are registered trademarks of Tandy Corporation. LDOS and LS-DOS are trademarks of MISOSYS, Inc.



```
CALL RDFILE ;Read SYS0/SYS into buffer
LD (CFGBCN),HL ;Start (in buffer) of 5.3 CONFIG/
SYS
LD DE,$CFGNAM
CALL RDFILE ;Read CONFIG/SYS into buffer
;
;All sectors and files have been read into memory
LD HL,(SYSGN) ;Start of SYS0 in buffer
LD DE,0662H ;Offset to version number in 5.3.0
SYS0
ADD HL,DE
LD DE,$VER$ ;Version required by this program
LD B,5 ;Length of string
$RD3 LD A,(DE) ;Compare strings
CP (HL)
JR NZ,$BADVER ;Mismatch means error
INC DE
INC HL
DJNZ $RD3
;
LD A,(CRTCFLG) ;If non-zero, we'll initialize
CRTC
AND A
JR Z,$RD4
LD HL,OFF3EH ;"LD A,-1"
LD (CRTC1),HL ;Enable MIII CRTC init routine
$RD4 LD A,(PRESS) ;If non-zero, skip 'Press any
key...'
AND A
JR NZ,$RD5
LD HL,$PRESS$
SVC @DSPLY ;Display 'Press any key...'
SVC @KEY ;Wait for a keystroke, don't care
which one
$RD5 SVC @CKBRKC ;See if <BREAK> was pressed
lately
JR Z,$RD6
LD C,CR ;If it was, display CR and abort
back to 6.x
SVC @DSP
JP ABORT
;
;Before leaving 6.x, get and save date and time
$RD6 LD HL,FCB ;Temp buffer for unneeded "mm/dd/
yy" string
SVC @DATE
EX DE,HL ;HL now points to actual values in
lowcore
LD DE,TIMEBUF
LD BC,3
LDIR ;Copy values into our storage
LD HL,FCB ;Repeat for time
SVC @TIME
EX DE,HL
LD DE,TIMEBUF+3
LD BC,3
LDIR
LD C,SI ;Cursor off
SVC @DSP
SVC @CLS
JP BOOT ;Jump to "Model III mode" section
;
;
;6.x subroutine: Read a file into our buffer
```

```

;Entry:  DE points to filename, HL points to buffer, IX
points to system flags
;Exit:   HL points to next free address in buffer (al-
ways a page boundary)
RDFILE  LD      C,L          ;Save pointer to file buffer
        LD      B,H
        LD      HL,FCB      ;Copy filename into FCB
$RDFIL1 LD      A,(DE)
        INC     DE
        CP      '+1
        JR      C,$RDFIL3   ;Space or ctrl-char means end of
filename
        CP      'a'
        JR      C,$RDFIL2   ;Force uppercase
        AND     0DFH
$RDFIL2 LD      (HL),A
        INC     HL
        JR      $RDFIL1
;
$RDFIL3 LD      (HL),':'     ;Add ':d',CR to filename in
FCB
        INC     HL
        LD      A,(DRIVE)
        LD      (HL),A
        INC     HL
        LD      (HL),CR
        SET     0,(IX+'S'-'A') ;Set "force-to-read" flag
        LD      DE,FCB
        LD      L,C          ;Original HL (buffer address)
        LD      B,B
        LD      B,0          ;LRL for unblocked file I/O
        SVC     @OPEN
        JR      Z,$RDFIL4   ;Jump if file successfully opened
        CP      18H         ;'File not in directory' error
        JR      NZ,ERROR    ;Other errors get standard DOS
message
        LD      HL,$NOFIL$
        JR      ABEND
;
$RDFIL4 LD      HL,4         ;Offset to MSB of file
buffer address
        ADD     HL,DE
$RDFIL5 SVC     @READ
        JR      NZ,$RDFIL6   ;Could be EOF (OK) or other error
(NG)
        INC     (HL)         ;Move file I/O buffer to next page
        LD      A,(HL)
$RDHICP $-$              ;Make sure we're below HIGH$
        JR      C,$RDFIL5   ;If all OK, read next record from
file
        LD      HL,$NOMEM$
        JR      ABEND
;
$RDFIL6 LD      H,(HL)       ;Address of next page of
buffer into HL
        LD      L,0
        CP      1CH         ;'End of file encountered' is OK
here
        RET     Z
        CP      1DH         ;'Record number out of range' is
OK here
        RET     Z
;
        JP      ERROR       ;Else it's a DOS error (fall
through to ERROR)

```

## MEMORY: HOW MUCH AND WHY

Ken Strickler  
P.O. Box 773  
Stanwood, WA 89292-0773

With the announcement by Peter Ray of ANITEK, of the recent MEMORY BREAKTHROUGH, now the model 4 owners can have as much memory as they "CAN AFFORD". Up to 8 meg (or more by special request) and easily installed too! I haven't tried it yet myself, but I have installed ANITEK's 768K memory in one of my computers, along with a speedup kit, and the IBM graphic set. Except for having to re-boot 2 or 3 times on initial startup, or until the screen quits displaying "GARBAGE", the additions work as advertised. I bought the LeScript sampler to try it out! (I currently use ALLWRITE, linked to Electric Webster and Dotwriter, and since I have had it for "YEARS" and have 3 licensed copies, probably won't change now.) Except for LeScript, I know of no other program which utilizes the super-expanded memory. Maybe there will be some in the future - BUT WAIT - IF I WRITE A PROGRAM WHICH REQUIRES THE ADDITIONAL MEMORY and offer it FOR SALE, or FOR FREE for that matter, and the only requirement is that the user invests in \$800 - \$1200 worth of memory, how many of you readers are ready to sign up for that plan ? (Talk about a SMALL GROUP OF USERS!)

Let's take a look at the possibilities any way. What would "LIVE" in this extra memory? Well, PROWAM can use a couple of BANKS, the SPOOLER can use 1 and then some disk space (read that RAM-DISK), if bank 1 and 2 are reserved, then Visi-Calc, TK Solver, or Allwrite use them IF THEY ARE NOT



ALREADY IN USE. That old OVER-DRIVE package in the MARK IV collection can do some DISK buffering and system module storing in either bank 1 or 2, and of course the re-vamped DoubleDUTY turns the model 4 into 2+ computers by switching resources between banks. Pretty neat and useful programs, to say the least, but not up to a meg yet! How about a RAM-DISK. YOU BET! Now here's an application that I can GET INTO! Multi-meg, online rapid access to everything! Just load 'er up 'n go! Did you ever think how long it takes to load 8 meg, even from a harddisk? How about floppies! Let's see, that's about 6-180K floppies per meg, times 8 - THAT'S 48 FLOPPIES TO CHANGE ON STARTUP! Better re-think this again! By the time this 'sucker' is loaded, it'll be breacktime or lunchtime. Maybe just leave it on all of the time! Yea, that's the ticket! And to protect it against a power failure, another \$400 or so for a SBS power system!

Let's "BLOCK" it out and see how the DREAM fits:

**Block \*0:** This is actually the upper 32K of system memory. We'll save that for the SYSTEM!

**Blocks \*1 - 2:** Reserved for Visi-Calc, TK Solver, Allwrite or another transient application program designed for extended memory use.

**Block \*3:** PROWAM - Window Applications Manager; Allows access to "HANDY-DANDY" programs from within another program. (calculator, calendar, dialer, etc.)

**Block \*4:** SPOOLER+128K RAMDISK SPACE (4-32K blocks); Print spooler allows printing while using the computer to do something else.

**Block \*5:** OVERDRIVE - provides disk buffering and system module storage (sys1-5,9-13).

**Blocks \*6 - 7:** A couple of blocks for

```

;
;
;Model 4 mode abnormal exits
ERRORLD    L,A          ;Entry: Error number in A
LD         H,0
OR         OC0H          ;Return here, short error message
LD         C,A
SVC        @ERROR
JR         $ABORT2
;
;
ABENDSVC    @LOGOT       ;Entry: HL points to our message
ABORTLD     HL,-1
$ABORT2     SVC          @EXIT          ;Abort back to 6.x
;
;
;Model 4 mode data: filenames and messages
$IMGNAM     DB          'MODEL'
$IMGLTR     DB          'x/III',CR
$SYSNAM     DB          'SYS0/SYS.SYSTEM',CR
$CFGNAM     DB          'CONFIG/SYS.CCC',CR
;
$PNAME$     DC          9,ETX          ;Holds name of /CMD file
$TITLE$     DB          'v1.0 - Copyright (C) 1990 by Adam
Rubin. '
DB          'All rights reserved.',LF,CR
$USE1$      DB          'Usage: ',ETX
$USE2$      DB          ' [imagenam] [:]drive',LF,LF
DB          'imagenam (4P only) is A to G, for '
DB          'MODEL4/III through MODELG/III',LF
DB          'drive contains your 5.3.0 system disk, 1
through 7',CR
$PRESS$     DB          'Press <BREAK> to return to 6.x, or '
DB          'any other key to boot DOS 5.3 ',ETX
$NOTM4$     DB          'Model 4/4P/4D required',CR
$OLDOS$     DB          'DOS 6.2 or later required',CR
$RADIM$     DB          'ROM image name must be 'A' through
'G',CR
$NOT53$     DB          'Drive : '
$NOT5D$     DB          'd is not a '
$VER$DB     '5.3.0 disk',CR
$NOMEM$     DB          'Too much high memory in use',CR
$NOFIL$     DB          'Cannot find ' ;Message continues
with contents of FCB
FCB DS      32
;
;
;
ORG         5200H
STACK EQU   $-2
;
;Start of "Model III mode" section
;First, set up hardware
;Ports are all above 80H, should be no problem for 64180
CPU
BOOT DI
XOR         A
OUT         (0E4H),A          ;No NMI
OUT         (84H),A           ;Model III memory map
LD          A,0D0H
OUT         (0F0H),A          ;FDC: terminate w/o interrupt
LD          A,78H
OUT         (0ECH),A          ;4 MHz, enable I/O bus & alt chars
;
;If CRTCLFG was non-zero, initialize 6845 CRTC registers 0-

```

```

15
CRTC1      JR      _LDIMG      ;Changed to "LD A,-1" to
Init CRTC
LD         BC,1089H
LD         HL,$CRTC2B
$CRTC2     INC      A
OUT        (88H),A           ;CRTC address register
OUTI       ;(HL) to CRTC data register
JR         NZ,$CRTC2

;
;Load ROM image (4P), initialize
_LDIMG     JR      $NOT4P      ;Changed to "JR 0" if 4P
SUB        A
OUT        (9CB),A           ;Make sure boot ROM is
switched out
INC        A
OUT        (84H),A           ;Write-enable 0000H-37FFH
LD         HL,0
IMBGN     EQU      $-2         ;Start address in buffer of
ROM image
CALL      LOAD                ;Load ROM image
LD         HL,4055H           ;Zero 4P boot ROM's data area
LD         DE,4056H
LD         BC,4070H-4055H-1
LD         (HL),B
LDIR
$NOT4P     LD         A,80H      ;Clear video page 1
OUT        (84H),A
LD         HL,3C00H
LD         DE,3C01H
LD         BC,4000H-3C00H-1
LD         (HL), ' '
LDIR
LD         A,C                ;Back to video page 0
OUT        (84H),A

;
;Copy Model III ROM data into RAM (normally part of ROM's
initialization)
LD         HL,36AAH           ;DE is 4000H from preceding LDIR
LD         C,36F6H-36AAH;Same addresses in all ROM and
image versions
LDIR
LD         L,0F9H
LD         DE,41E5H
LD         C,3739H-36F9H
LDIR

;
;Load 5.3 SYS0/SYS, alter it where necessary
LD         HL,0
SYSEGN     EQU      $-2         ;Start address in buffer of
5.3 SYS0/SYS
CALL      LOAD                ;Load SYS0
PUSH      HL                  ;Save SYS0 transfer address as
return address
LD         A,0C3H             ;Change NMI handler's "RST 0" to
"JP addr"
LD         (4049H),A          ;(normally done by ROM and boot
sector)
LD         HL,RDSEC02         ;Change sector reads to use our
routines
LD         (4E99H),HL
LD         HL,RDSEC00
LD         (4FF6H),HL
LD         HL,LDCFG           ;Our routine to load CONFIG/SYS

```

DoubleDUTY to use for program switching. (This currently is NOT possible to have DDUTY concurrent with some of the others but, remember this is a WISH LIST.)

Blocks \*8 -13: 192K RAMDISK image area - used for image of 40 track SS,DD disk. This can be a SCRATCH area for a DATA FILE, which when in operation can save wear and tear on a floppy disk, to say nothing about the tremendous amount of disk access time.

Blocks 14-36: 736K RAMDISK image area - to image 720K disk if needed.

Blocks 37-40: 4 blocks reserved for SPOOLER (block4).

That ought to about do it! 1.28meg, plus whatever might be needed if you are running LeScript (make that afford). If the WANT list is PARED back a bit, and the 720K disk image, 128K SPOOLER RAMDISK, DDUTY extra banks (let it run in bank 1 & 2 - since it currently isn't really compatible with all of the rest of the stuff), OVERDRIVE block, let it run in bank 1 or 2 (you probably don't have this program anyway - maybe Roy will let it out again!), you arrive at 256K! The real secret to utilizing huge RAMDISK areas, is to try not to spend too much time loading and unloading them. Obviously, they will have to be loaded and unloaded at least once. Maybe only unloaded only, if the application program builds them from an empty state, like the SPOOLER, and then unloads them while doing something else the use is transparent to the user. OVERDRIVE would be the same way, DDUTY and PROWAM also fall into that category! If the current offerings, SPOOLER, PROWAM, etc. would load themselves TOP-DOWN instead or BOTTOM-UP, an automatic loading sequence could be used which would load the system depending on the resources available. It is easy enough to generate a JCL file to perform the loading task for any given system, but that will only work for a similar system. There are programs like PARMDIR which use a temporary

disk area, and the RAMDISK is ideal.

It's funny that the extra storage space should work out to 256K + 64K (marked by '\*' above) of the expanded 128K model 4, since the XLR8er just happens to provide that storage, plus the added speed and instructions of the 64180 processor chip! The whole thing slips right inside too for \$200.00 (TMQ IV-I). I wonder if a 512K version will ever be available?

If you have been following the evolution of the bigger systems, it is interesting to note that there is a trend to writing smaller modules, and using overlay techniques to eliminate the time required to load a huge section of code to perform a small routine. Since the program always grows to exceed the available storage space, some program swapping will always occur. Why not design it in, rather than adding it on!

OOPS! A bit of a tangent there! Well, just a little over 1 meg seems to fill today's dream list! Comes in 1 meg chunks, so the 3 meg board for \$149.95 and a couple of 1 meg SIP's at \$125.00, lets you in for \$399.95 + taxes + shipping and handling. The extra 6 meg of SIP's at \$125.00 each would cost \$750.00. For that price, a nice 40meg HARDDISK is available from MISOSYS.

If you need that added memory, don't let this discourage you from getting all you can use. ANITEK is there when you are to fill your fondest dreams!



```

LD      (506DH),A      ;Always jump, even if <CLEAR>
pressed...
LD      (506EH),HL      ;      LDOS needs CONFIG/SYS to
access HD
LD      A,11H          ;"LD DE,nnnn"
LD      (4FF0H),A      ;Don't CALL DIRCYL
LD      (502FH),A      ;Don't CALL @DEBUG if <D> or down-
arrow key...
;      as that would suppress
CONFIG/SYS
;
;Move 6.x date and time into 5.3, and RET to 5.3 SYS0 entry
point
DT1 RET                ;Changed to NOP if 5.3 date/time
should be set
LD      HL,TIMEBUF      ;Move date
LD      A,(HL)          ;Year
SUB     50H
LD      (4413H),A
INC     HL
LD      A,(HL)          ;Day of month
LD      (4457H),A
INC     HL
LD      A,(HL)          ;Month
XOR     50H
LD      (442FH),A
INC     HL
LD      DE,4217H        ;Move time
LD      BC,3
LDIR
RET

;
;Resident code during SYS0 initialization and CONFIG/SYS
load
;
;Subroutine to load a file stored in memory
;Entry:      HL points to start of file
;Exit:      HL contains file's transfer address
;      AF, BC, DE altered
LOAD LD      A,(HL)      ;Block type
INC     HL
LD      B,0
LD      C,(HL)          ;Block length
INC     HL
DEC     A
JR      Z,$LOAD1        ;Jump if load block
DEC     A
JR      Z,$LOAD2        ;Jump if transfer address
ADD     HL,BC           ;else skip over block
JR      LOAD

;
$LOAD1 LD      E,(HL)      ;LSB of load address
INC     HL
DEC     C
LD      D,(HL)          ;MSB
INC     HL
DEC     C
JR      NZ,$LOAD1A
INC     B               ;0 means 100H bytes here
$LOAD1A LDIR
JR      LOAD

;
$LOAD2 LD      A,(HL)      ;LSB of transfer address

```

```

INC     HL
LD      H, (HL)      ;MSB
LD      L, A
RET

;
;Subroutine to load 5.3 CONFIG/SYS
;Entry:  Anything
;Exit:   HL has transfer address, Z set, A altered
LDCFGPUSH DE
        PUSH BC
        LD HL, 0
CFGGBGN EQU $-2      ;Start address in buffer of
5.3 CONFIG/SYS
        CALL LOAD
        POP BC
        POP DE
        XOR A
        RET

;
;Subroutines to replace @RDSEC and @RDSSC calls in SYS0
initialization
;Entry:  HL points to destination for sector data
;Exit:   Z set, A zeroed (same as 5.x @RDSEC)
RDSECO2 PUSH DE
        LD DE, SECTBUF ;Start of sector data in our
buffer
        JR $MVSECT

RDSECD0 PUSH DE
        LD DE, SECTBUF+100H

$MVSECT PUSH HL
        PUSH BC
        EX DE, HL
        LD BC, 0100H ;Sector size
        LDIR
        POP BC
        POP HL
        POP DE
        XOR A
        RET

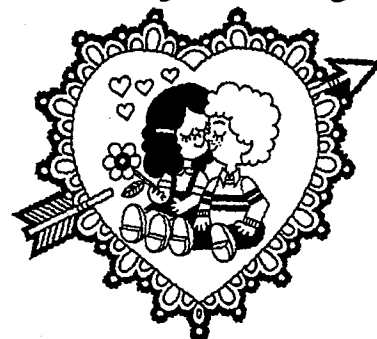
;
;Model III mode data storage
;DS statement is first, to make DB force a new load block
TIMEBUF DS 6 ;Date and time values to
transfer to 5.3
        DB ' ', 'CRTC' ;Eyecatcher in /CMD file
;The 16 bytes after 'CRTC' in /CMD file are CRTC init val-
ues, R0 to R15
;Untested guesses for CRTC init data, should be patched by
user
$CRTCIB DB 79, 64, 69, 10, 28, 3, 16, 18, 0, 8, 85, 8, 0, 0, 0, 0
        IFNE $, $CRTCIB+16 ;Check for right number of values
at assembly
        ERR 'CRTC init table is wrong length'
        ENDIF

;
        ORG .HIGH.$+1<8 ;Next page boundary
SECTBUF DS 200H ;Sector buffer (cyl 0 sect
2, dir cyl sect 0)
FILEBUF EQU $ ;Buffer for files, from here
to 6.x HIGH$

;
        END START

```

*Our capacity for love is infinite!  
What about your disk storage?*



*Improve your on-line storage with  
5.25" half-height 360K or 3.5" 720K  
floppy drives, or hard drive kits  
pre-assembled by MISOSYS*

#### Floppy Drives & Accessories

5.25" 360K 1/2 ht	\$75+\$5S&H
3.5" 720K in frame	\$85+\$4S&H
Our dual vertical external floppy disk drive enclosure will hold two 5.25" half- height disk drives.	
2SV5 drive case/PS	\$60+\$6S&H
8" 2-floppy extender	\$18
4 foot 1-drive floppy cable	\$12.50

#### Hard Drive Pre-assembled Kits

Our kits come assembled in a cabinet holding up to two half-height drives, 60 watt p/s, fan, SCSI controller, host adaptor, hardware clock, and cable. Software includes: driver, formatter, archive, restore, and sub-partitioning - all for LS-DOS 6.3 (LDOS software \$30 additional). Drive kits work with Model III or Model 4 hardware.

20 Megabyte kit	\$450+\$S&H
40 Megabyte kit	\$575+\$S&H
Joystick port option	\$20

S&H prices are UPS ground to continental U.S.

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



## 300 DOTS: AN UPDATE

Gary Shanafelt  
Dept. of History  
McMurry University  
Abilene, TX 79697

Some of you may remember an article which I wrote in the Fall 1990 issue of this periodical, "300 Dots on the TRS-80". If you anticipate a 300 dpi printer in your future, you may be interested in a number of developments which have occurred since then.

In the area of hardware, a major event is Hewlett-Packard's new Deskjet 500, which replaces the Deskjet Plus. The 500 accepts earlier model Deskjet fonts, but it also uses a new font format that allows kerning of letters. Kerning is when letters overlap each other, such as the right leg of a capital "A" extending under the left arm of a capital "V". This makes Italics and similar faces which use a heavy amount of kerning much more attractive than on the earlier Deskjets. Kerning is already standard on the Laserjet printers. HP has also introduced new ink cartridges for all the Deskjets, which produce printout that is much more water resistant than what the original ink allowed (or failed to allow). You no longer have to worry about a sweaty finger destroying a whole page of printing. The new 500 has a street price of under \$500. Of course, more and more Laserjet compatibles are reaching street prices of under \$1000, which means that you don't have to be a corporate executive to own a printer that makes even a 24-pin dot matrix machine look like something produced by Apple or Commodore ten years ago.

Prosoft, the producer of AllWrite, is now out of business. If you want this word processor, you may be able to find leftover copies with third party suppliers, but you can no longer get it from the source. Laser support is now

available through the Allwrite Laserjet and Deskjet Utility Packs compiled by Lee Rice and myself and distributed by *Computer News 80*. Each provides a printer driver supporting several public domain soft fonts included with the package and, in the case of the Deskjet version, support for the most common cartridge/internal fonts. More important, utilities are included which can read almost any HP soft font and generate a width table for it, as well as a driver compiler which allows you to create your own printer driver for any combination of fonts that you want (for a total of 64 fonts in any one driver). A downloader allows both individual and batch downloading of soft fonts into the printer. (The downloader and additional utilities to send specific escape codes to the printer will work with any program: you don't need Allwrite to use them). The cost of each utility pack is \$20, which includes several disks and a manual. The main limitation is that the utilities will only run on a Model 4, so if you have a Model 1 or 3 you can use the supplied driver but you can't create a new one yourself without buying a new computer.

The File Cabinet has begun a collection of both Laserjet and Deskjet public domain soft fonts for those who are interested.

Laser support has also materialized for users of Scripsit and Scripsit Pro, thanks to the efforts of David Goben. The tack taken here is a little different than what is available with Allwrite. Instead of a single big package, the user has the option of acquiring a number of small ones. There is first of all the Professional Print Driver which supports the Laserjet Plus (and compatible) internal fonts to print at 10, 12, and 16 pitch spacing. It also supports single and double underlining, boldface print, super and subscripting, and slashed zeroes. The driver works with all versions (Model 1, 3, and 4) of SuperScripsit; a separate package works with Model 4 Scripsit Pro. Each sells

for \$20, through *Commuter News 80*.

Goben has also produced a number of soft font drivers. Each is bundled with a particular public domain typeface. The 10-point Helvetica disk, for example, gives you several faces of 10-point Helvetica type (regular, Italic, boldface, etc.), a driver which supports the faces, and a downloader which downloads them into the printer, both singly or in batch mode. Goben is adding fonts at a regular pace and will customize a driver on request. Each font and driver disk costs \$10 through, again, *Commuter News 80*. Separate versions are available for Model 3 SuperScripsit, Model 4 SuperScripsit, and Model 4 Scripsit Pro, so you don't have to have a Model 4 to use them. But there are currently no Deskjet versions, so if you want to use a Deskjet you have to get Allwrite.

Of course, the current version of Anitek's LeScript word processor has Laserjet support, too, as detailed in my original article.

Goben has also written a very flexible utility for printing TRS-80 hi-res files on the Laserjet (and Deskjet) printers, HRLSRPRT/CMD. You can print in any of the four graphics resolutions supported by the printers (75, 100, 150, 300 dpi) and you don't need a hi-res board. The program is one of a number of hi-res utilities which he recently created. HPPRINT/BAS, mentioned in my original article, has also been upgraded so that you can print files from disk without a hi-res board. It now runs from BASIC as well as MicroLabs' GBASIC, though the features which do require the hi-res board also require GBASIC. HRLSRPRT/CMD and HPPRINT/BAS are both available from The File Cabinet; the latter is additionally included as a supplemental utility in the Allwrite Deskjet Utility Pack.

The bottom line in all this is that you don't have to buy a whole new computer system to use a laser printer. Unless you want to do fancy page layout work, you can go right on using your current word processor (at least if it is Allwrite, SuperScripsit, Scripsit Pro, or LeScript) with a minimal financial outlay beyond the cost of the new printer itself.



# LAIR OF THE DRAGON

## HINT SHEETS #1 through #4

by David Goben  
417 Prospect St., 1st Flr  
Willimantic, CT 06226

### MAPS OF ALL MAZES

Mazes within Lair of the Dragon can seem a great bother, but they are more than just a tradition in adventures. In most adventures, they are used to conceal important items, or make attaining a goal (and usually an important one) more difficult. These two basic concepts are used here.

The maze of twisting passages is found within the mines worked by the dwarves. This maze both conceals items [it will be left for you to figure out what they are), and make travel difficult toward attaining an important object that is one of the main keys in solving the final puzzle in the adventure. The dead ends are also worth exploring.

The map for this maze uses the location labels to the left of the map grid below. The direction of travel from any of these locations is situated at the top of the grid.

To move, you would align yourself with the current location description to the left of the grid. Select a direction from the list at the top of the grid, and make note of the location number that exists in the row and column common to both the current location and the desired direction. The targeted location number can be found to the left of the location descriptions.

The map also uses the following legend: DE = Dead End, and OUT = out of maze.

The Black forest is the most treacherous maze, as its labels are all alike; a maze feature common to many adventures. Unlike the maze in the dwarf mines, this maze conceals nothing (at first), but makes the trek to the dragon's mountain much more difficult. It is also the home of the Giant Black Widow Spiders. One nice feature is that all but a few exits logically place locations from each other. Thus if you go west, you can

usually return to the previous location by going east. But take note that some passages are one-way. This map will also use a grid. Arbitrary location numbers will be used to mark each section of the Black Forest on the left-hand side of the grid. Directions of travel are marked at the top. Again, DE represents a Dead End, and OUT represents a way out of the maze.

#### MAZE OF BLACK FOREST

	N	S	E	W	NW	SE	NE	SW	U	D
1	1	1	3	1	1	1	1	OUT	1	1
2	5	2	2	2	2	2	6	2	2	2
3	7	DE	3	1	3	3	3	3	3	3
4	12	4	DE	3	4	4	8	4	4	4
5	5	2	5	5	9	5	5	5	5	5
6	6	6	7	6	6	6	1	1	2	6
7	7	3	7	6	7	7	7	7	7	7
8	8	8	8	8	8	8	1	3	4	8
9	DE	9	10	9	5	9	9	9	9	9
10	10	11	10	10	10	12	10	9	10	10
11	11	11	11	11	11	11	11	6	11	11
12	10	12	12	12	12	12	12	4	12	12
13	DE	13	13	13	13	8	OUT	13	13	13

A final note about this maze: if per chance something happens that causes you to lose an object against your will, you will be able to find it here.

MAZE OF TWISTING PASSAGES	N	S	E	W	NW	SE	NE	SW	U	D
1. twisting maze of passages	1	2	1	1	OUT	1	1	1	1	1
2. maze of twisting passages	1	2	2	2	2	2	2	3	2	2
3. maze of twisty little passages	3	3	6	3	3	3	2	3	3	3
4. maze of little twisting passages	4	4	9	4	4	4	6	4	4	4
5. twisting maze of little passages	5	5	10	1	5	1	5	5	5	5
6. twisty maze of passages	DE	6	6	3	6	6	6	4	6	6
7. twisty maze of little passages	7	8	7	5	7	7	7	7	7	7
8. maze of twisting little passages	8	8	8	8	7	9	8	8	8	8
9. maze of passages	9	9	9	4	8	9	9	9	9	9
10. twisting maze of passages	10	11	10	5	10	10	10	10	10	10
11. maze of little twisty passages	10	11	11	12	9	11	11	OUT	11	11
12. edge of pit	12	12	11	12	12	12	12	DE	12	OUT

## "SOLVING PUZZLES IN THE LOWER REGION"

The best help you can get is to read the manual that accompanied the program disk, and to keep a **sharp** eye out for cleverly incorporated clues.

**BASEMENT OF THE INN:** Many people seem to be having trouble here. To see what is happening, you will need light. A person who has a feel for his environment will know where to look.

Getting out of it has presented problems for others. There are actually two different ways out. The first can be found if you have managed to explore another place nearby first. Of course you will still need light to see by. Another object, concealed elsewhere, but nearby, can be used to force your way out, once you have managed to produce some light.

**CROSSING WATER:** Crossing the bridge close to home is a cautious procedure, as it cannot hold much weight. There is a ferry boat nearby that will be of great help here, and is in fact the preferred method of crossing. Of course later on you may find that the rope that moors it to each shore might be useful for something, and so still another object is provided to accomplish this. Of course it is left for you to figure out how.

If you are determined to explore the area below the bridge, you might try getting into the boat upstream and cutting the rope to do so, if all other attempts fail.

**LUMBER SHACK AREA:** Here a lot of problems arise. Here you must also be curious and daring. Be sure to explore the fireplace, as a clue will be provided that will eventually lead to the means for getting into the dragon's lair. When you get to the top of the building (somehow), pay close attention to your surroundings, as that will lead to more points. Be sure to find the lumber pile and examine it care-

fully. Don't accidentally drop anything.

**FOREST:** The forest (not the **Black Forest** found elsewhere) is bigger than what you think. Be sure to read all descriptions **carefully**. Wording, such as *virtually*, or *about* do not indicate a limit, but only that you are **NEAR** that limit. Also, some forest paths, like dark tunnels, can get confusing, and exits are not always obvious. So don't be thrown off simply because an exit is not **explicitly** spelled out. Take note of how some descriptions are worded. Also, a careful adventurer will be sure to explore each location carefully, to insure that they have not missed any inobvious avenue that will lead to greater riches. A lot of adventurers have never made it into the dwarve's mines because they were thrown off by cleverly concealed clues that elude to such avenues.

**DEALING WITH THE THIEF:** If for some reason you meet up with this shady character, don't make any attempt to raise his ire. Though you can easily hurt him, he may prove your greatest ally when you need help the most (when and where — except that is elsewhere — are for you to discover). But, by killing him you can get into the dragon's lair from even here. A few adventurers have even managed to accomplish all tasks (except one) because of it. It might be an interesting diversion once you have successfully completed the adventure at least once.

**OTHER ADVERSARIES:** The other characters that will not seem so nice to you in this region can be successfully dealt with if you take notice of how they arm themselves, and you try to arm yourself likewise. Arming yourself to face your first such opponent is a difficult task, and requires a sharp eye, quick wit (and quick retreats) before you will find it.

## "SOLVING PUZZLES IN THE UPPER REGION"

The best help you can get is to read the manual that accompanied the program disk, and to keep a **sharp** eye out for cleverly incorporated clues.

The middle region includes two mazes, which can be easily navigated with "Maps of all Mazes". Once you are past the Black Forest, and before you get into the Dragon's Lair, is actually quite simple, and is in fact much easier to navigate than the lower region. This was done in part to offer you some relief from your trek through the mazes (providing that you did not use the hint sheet). But aside from fending off the foes that haunt you here, there are two major puzzles here. Only after accomplishing both tasks will you be able to enter into the mountain, which is the home of the dragon.

**OPENING THE OUTER DOOR OF THE LAIR:** One puzzle is to open the padlock. For some reason, this has frustrated a lot of players to a degree that I find alarming. It would seem that none of them had ever watched many spy movies, or something. If you cannot open it by means that would at first seem obvious, then try what any **good** thief would try. Of course what you need can only be found a **long** way away from here. This object, which I will not reveal, has frustrated several adventurers simply because they are actually **afraid** to obtain it, expecting disaster if they do. Be brave. It is harmless, and only has one "tooth". The world will not tumble down around you if you take it. Of course to take it, you will need something else with which to garner it. Another object, also near the other, can be found at the top of another object. Try one or the other.

**OPENING THE INNER DOOR OF THE LAIR:** This puzzle actually consists of three other puzzles. The key to doing it lies elsewhere. Once you have gained

entrance through the outer door, examine the surrounding beyond it. To some the clues are obvious, and to others they are invisible. What they indicate is another room elsewhere, which can be attained only by accomplishing two of the three puzzles.

The first puzzle is actually so easy that it is almost stupid. And that is, if you are too heavy to scale something; this indicates that you must drop a few things to make yourself lighter. This may seem obvious, but it's amazing how many people are getting stuck here. Don't worry about the thief, as his appearances, if at all, are rare (although sometimes he seems to appear at every turn). A good guideline is to take only what you feel is necessary to deal with the inhabitants, plus something in the light department.

The next puzzle is much more difficult. You will find yourself at a very tight passage, and cannot in fact pass it unless you go in virtually empty-handed. The problem is of course—keeping light and protection. There are three light sources in the adventure (aside from the box of matches). Choose the smallest.

The third puzzle will only work if you have first opened the outer door. If it is closed, attempting to do anything will come of naught. This third puzzle, simply because of this requirement, has been deemed much more difficult than what it indeed is. Deep within the mountain, beyond the tight opening, you will find a small chamber from which you can open the inner door to the lair by mechanical means. Proper manipulation of the device there will prove fruitful.

Once these tasks are completed, it's then time to sharpen your adventuring skills, as further exploration of the lair will not be quite so easy. Happy adventures.

## "SOLVING PUZZLES IN THE DRAGON'S LAIR"

This is the most difficult region within the adventure. Even those who have fully explored the lair can easily slip up if they are not careful. So **be cautious, be careful, and be observant.**

**RAT DEN:** The first crucial obstacle is found in the Rat Den, where other obstacles and puzzles, not immediately apparent, will be found. The rats are very treacherous. Try stalling them with something along the lines of what they are looking for. If that doesn't last long enough, look around and see if you can find something else. That may finish the job. Once done, you will be safe from them.

If at a time before you have tried sneaking anywhere, a clue as to what you need would have been provided. You will find it here. Its usefulness will become very apparent were you to explore very far **without** them in certain parts of the mountain.

Rendering proper respect to the dead will also grant you points. Do what you would do naturally if you were actually in that situation (outside of perversion).

**BABY DRAGON:** Once you are well away from the den by climbing the well-like passage above it, on your return you will meet up with a surprise: a baby dragon. You will need two things to stop it; 1) a way to protect yourself from its fire, and 2) something to quench its fire. These two items can be found somewhere within the mountain. There is also a use for the dragon egg found there. Be sure to examine it.

**SANDY CAVERN:** Properly armed with the right tool, you will find something that will buy you passage out of the mountain after you have vanquished the dragon.

**SWORD IN STONE:** For some reason, this seems to stump a lot of people. Examine everything **carefully.** Hopefully, you

have fully explored the dwarves' mines.

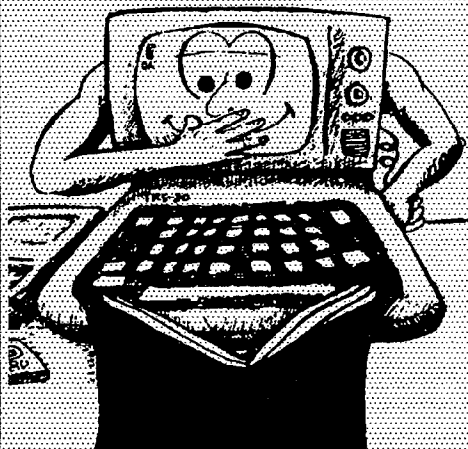
**THE WEST HALL:** There is a passage south of the west hall, which the dragon is blocking. Try giving the dragon something so that it will move. Appeal to its vanity. You might choose to wake it first. Either way, dragons are **very** light sleepers.

Getting down the ridge requires a rope tied to the rail. Once down, there are several puzzles to solve. Beyond the edge of the pool, there is something that will provide you with clues toward attaining other tasks. Pay close attention to the dead end and the small flat rock. The boulder at the dead end is the key to killing the dragon, but explore elsewhere to find the means to use it. Never expect clues to be as obvious as they may seem. You are entering the **master** portion of the adventure here, and you will have to keep your wits about you. Just remember this: make sure that the dragon cannot escape, if it attempts it.

Getting out of this area can be accomplished by going through the double doors, or by negotiating the green wall that will be blocking your way back to the west hall. Be daring, especially if you are attempting to escape the disaster that you had previously put into effect to bring on the dragon's demise. Once you have initiated the disaster, until you have escaped the dragon, valor and heroic deeds are expected of you at every turn. Throw caution to the wind (easy to do with game saves). At first take on the dragon as though there was no tomorrow. If all goes well (e.g., you being properly armed), the dragon will leave momentarily. Hopefully it will return, and try to make mincemeat out of you. Grab whatever you can and **race** out of the mountain through the tunnel you first entered the lair from. Put **anything** possible between you and the dragon. If you meet the dragon in the meantime, don't tarry for idle chit-chat.

If all went well, you should find yourself trapped in the long tunnel. Getting out of here will depend upon how careful an observer you are, and if you did **not** kill the thief. The clue toward solving this final puzzle can be found in the black book. Figuring that will be left entirely to you. After all, to gain a master rating, you have to earn it somehow.

# TRSTimes magazine



TRSTimes is the bi-monthly magazine devoted exclusively to the TRS-80 Models I, III & 4/4P/4D.

We are in our fourth year of publication and each issue typically features: 'Type-in' programs in Basic and Assembly language, Hands-on tutorials, Hints & Tips, Reviews, Questions & Answers, Letters, Nationwide ads, Humor and more.

1991 calendar year subscription rates (6 issues):

U.S. & Canada: \$18.00

Europe & South America: \$23.00 surface or \$29.00 air mail

Asia, Australia & New Zealand: \$25.00 surface or \$32.00 air mail  
(all payments in U.S. currency, please)

TRSTimes magazine

5721 Topanga Canyon Blvd. #4  
Woodland Hills, CA. 91367

## GIF4MOD4 AND HR2GIF

for the Model 4/4P/4D

They say a picture is worth 1,000 words. This picture was converted from GIF to TRS-80 format using GIF4MOD4. Until now, Model 4 users had no way to view GIF images or to send their own hi-res graphics creations to other types of computers.

GIF4MOD4 will decode any GIF image up to 640 x 480 x 256 (UGA)



and put it on your hi-res screen. If you have no hi-res board, GIF4MOD4 puts it in an /HR disk file so you can dump it to your dot-matrix printer. HR2GIF converts /HR, /CHR and /BLK files to GIF format.

ORDER FORM -- Mail to  
J.F.R. Slinkman, 1511 Old Compton Rd., Richmond, VA 23233  
Please send me GIF4MOD4 & HR2GIF for my TRS-80 Model 4.  
I am enclosing \$37.95 + \$2.00 S&H (add \$2.00 outside N.  
America. Virginia residents please add 4-1/2% [\$1.71]).

Name \_\_\_\_\_  
Address \_\_\_\_\_  
City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_



# PowerMail Plus

"This is one of the BEST mail-list programs we've seen for the TRS-80, and for its price, it's a down-right bargain."

INFOWORLD

Rated FOUR STARS in the May 1984 80-Micro!

## PowerMail Plus

★★★★

MISOSYS, Inc.  
PO Box 239  
Sterling, VA 22170  
Model 4  
(Model I/III and II/12/16  
versions also available)  
\$150

Easy to use? ★★★★★  
Good docs? ★★★★★  
Bug free? ★★★★★  
Does the job? ★★★★★

### ● INFOWORLD's Essential Guide To The TRS-80 said...

"If you need to maintain large mailing data-bases, you should definitely consider PowerMAIL Plus... It is exceptionally fast The program is intrinsically efficient... overall concept and design are excellent. PowerMail Plus offers the kind of features that businesses, in particular, need."

### ● 80-MICRO said in their May '84 issue,

"If you're looking for a mailing-list system, I highly recommend PowerSoft's PowerMail Plus. (The) manual is high in quality, content, and workmanship. The commands were easy to use and remember."

### ● OMNI's Complete Catalog of Computer Software wrote...

"This is a very potent mailing-list program with many versatile features. ...It lives up to (PowerSoft's) normal high-standards... it's an excellent mailing list program."

PowerMAIL Plus is a highly acclaimed, deluxe mailing/data storage system written entirely in machine language for maximum operating speed. Besides being a mailing system, it is also capable of generating custom personalized "form letters" to whatever groupings of labels you wish using the TEXT-MERGE module, available separately. Many of its features, such as extensive data "flags" enable you to use it as a limited data-base manager, too. PowerMAIL+ is a "DISK-BASED" system, rather than "RAM-BASED", therefore it can keep track of over 520,000 individual entries by combining up to eight different mailing lists (which, of course would require a hard drive). There are no "slow" periods when PowerMAIL+ is running. Features have been added to this program that others have always lacked. You now have the ability to keep track of mailings using 24 user defined "flags" that are incorporated into the PowerMAIL+ program. Separate any category in any manner you wish.

PowerMail+ can use multiple floppies by logging them in and out of the system. The program will run in as little as a 32K one drive environment, but dual drives or 80 track or double sided drives, or a hard drive are recommended for more serious use. The more available freespace, the more names you can store. A Model III or 4 data disk will hold about 1150 names. Double that for 80 track or 40 track double-sided drives (4D). Works great on hard drives as well as floppies! Handles drives 0-7. PowerSoft used it on their hard drive system for over four years with excellent results! Does not pre-allocate the entire drive, but allows you to define how many records you need at initialization time. If you need more room later, there is a way get your names onto a LARGER file easily. Also you may have a file on each disk or platter, if you have multiple drives, or a hard drive. This further increases overall storage, as PowerMAIL+ will treat the files individually, or as an overall total. TWELVE LEVEL SORT! The program will sort up to TWELVE levels that you wish in any order. Other features include the ability to separate your flags and put them onto another file, merge files together, and then separate whatever you want, improved "key" search, improved field lengths, improved disk I/O, and much improved print routines. The main thing here is that PowerMAIL+ was designed to be easy to use, easy to start up, and easy to train people to enter your names.

With PowerMAIL+, you have many print options based on Labels or Lists. If you choose labels, you have a choice of two modes and are asked how many labels across and the spacing required. The default settings are for the standard one up label commonly available (you may choose up to 4 across). If you choose listings, you have four choices to choose from there. If printing a LIST rather than LABELS, a user-definable "header" will be printed, along with the page number. This really makes an impressive printout, as well as adds to the usefulness.

PowerMAIL Plus' print system can be controlled from flag settings, allowing you to print only a part of the file. You may keep lists within lists. No need to keep separate files for different classifications. Use the flag or DATA fields for separating them for printout. There is a lot of flexibility here! In addition, you can tell it to SET FLAGS after printing to designate WHO got a label, etc.

This will contribute tremendously to the efficiency of your mass mailings, because for the first time you can keep track of who has been sent a particular mailing. Avoid duplicate mailings with ease! Let PowerMAIL Plus do it for you!

PowerMAIL Plus also incorporates a very fast search mode for locating a particular entry in the shortest possible time (supports wild cards too), as well as a sort-merge routine which allows you to sort files larger than can fit into memory. Best of all, PowerMAIL Plus includes routines which will convert files created by some of the other popular mailing list systems to its own internal format, so you don't waste any time retyping.

### Converts data files from these mail systems:

Radio Shack MailList Expanded  
Radio Shack MailList Compressed  
Special Delivery  
Extra Special Delivery

POSTMAN  
PowerMail 1.0  
Galactic Mailfile

## Text-Merge

TEXT-MERGE is a stand-alone form letter utility for PowerMAIL Plus. It will take a form letter prepared in ASCII format by any word processor or text editor (i.e. TED), and merge the contents of a PowerMAIL Plus ADDER file into the letter. The form letter may be as long as available memory (usually about 32K) or as short as a few lines. TEXT-MERGE is ideal for creating personalized mailings or specially-formatted mailing labels. You use PowerMAIL Plus' flag system to separate those records for which you want form letters printed into an ADDER file, then simply run TEXT-MERGE.

Prompts allow you to specify such parameters as page size, number of printed lines per page, line length, left margin, whether or not linefeeds follow carriage returns, and whether or not you want to pause after each page. During printing a count of letters printed is displayed on the screen.

Since TEXT-MERGE accepts all ASCII codes from 0 to 255 as valid data, you can create special print effects by embedding printer control codes in your form letter text. If you have a word processor which will permit this.

Fields from the PowerMAIL Plus record are inserted into the form letter at places marked by @n, where n is a digit from 0 to 9, which corresponds to the record fields. Field data may be inserted in any order, and each field may appear as many times as necessary.

Here is how you might start out a form letter;

Mr. @2 @1  
@3  
@4 @5  
@6, @7 @8

Dear Mr. @1 ,

Here is what the resulting printout would look like;

Mr. Robert Jones  
ABC Paper Co.  
1632 North St. Suite 101  
Ft. Worth, TX 76751

Dear Mr. Jones,

Model I/III/MAX-80 or Model 4/4P. Please Specify which computer!



# BACK/REST 1.3

with RESTORE/DCT now included!

A FAST Hard Drive "Image" Backup and Restore Utility.  
Backups or restores an entire logical drive, or file-by-file.  
*Files larger than a floppy are NO PROBLEM any longer!*

This is a **MUST-HAVE** program if you use a **hard drive** with LDOS or TRSDOS 6".

Awarded \*5 STARS\* in the October '85 Issue of 80-MICRO! To quote from the review...

*"BACK/REST is strongly recommended as insurance against possible digital disaster."*

★ ★ ★ ★ ★

80 Micro, October 1985

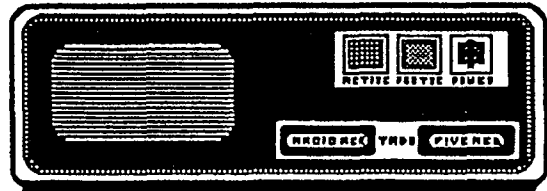
Backrest runs on the Models I, III (48K), and 4 (64K) and requires one disk drive and a hard drive. Powersoft Products, 17060 Dallas Parkway, Suite 114, Dallas, TX 75248, 214-733-4475. \$99.95.

Easy to use: ★ ★ ★ ★ ★  
Good docs: ★ ★ ★ ★ ★  
Bugs: ★ ★ ★ ★ ★  
Does the job: ★ ★ ★ ★ ★

- 1991 Price: \$34.95 + S&H
- For Model I/III or Mod 4 TRS-80 (or MAX-80).
- All versions included on same disk.
- Works under LDOS 5 or TRSDOS/LS-DOS 6.x.
- TRSDOS 6.2/LS-DOS 6.3 is fully supported.
- Program is not protected.
- BACK/REST will work with all TANDY hard drives and most others that use LDOS or TRSDOS 6.

How many times have you hard drive users put off making that all-important backup because of the length of time it takes? You know you should backup everyday, right? Let's face it, the normal file-by-file backup method employed by today's operating systems takes so long that it actually discourages people from doing backups. So you only do it every now and then... When you think of it... Maybe once a week. *Right?* You're ok until the day that disaster strikes.

Then there's ALWAYS the problem of files too large to fit on a floppy! Like PROFILE or other types of data files, BACK/REST solves all that! It segments those and can restore them back without you having to worry about it and figure it all out. BACK/REST is a FAST backup and restore utility for hard drives which can create a mirror-image of your hard drive partition on floppy disks in 20% of the time a file-by-file backup would normally take; a 5 megabyte drive should take about 10-15 minutes. There is NO EXCUSE for not backing up your hard drive now... do yourself a favor. BACK/REST is simple to use and will even tell you how



many floppies you will need to have formatted in advance for backup. We've tried to think of everything! It's a terrible feeling when you are doing a backup and are requested to insert another disk, yet you don't have any more formatted! Now BACK/REST will tell you how many disks you need to have on hand BEFORE you start. Furthermore, only allocated cylinders of the hard drive are copied to speed things up substantially.

Through the use of a special DRIVER program (RESTORE/DCT) supplied with BACK/REST 1.3, you can READ or COPY individual files from your BACK/REST backup floppies back to the hard drive. RESTORE/DCT is installed with the SYSTEM (DRIVE=x, DRIVER) command just like any other disk driver and will prompt you when to swap disks in order to extract the file you want. You now have the best of all worlds: a FAST, COMPLETE backup/restore utility for a hard drive that gives you the option of restoring either an entire hard drive partition OR a single file! When restoring to hard drive using the "image" mode, the floppies can be inserted and read in any order. When restoring "file-by-file", you are prompted for the proper disk #. You may restore a single file or any set of files that you want to. Never be at the mercy of a crashed hard drive again. BACK/REST is the answer to every hard drive user's prayers!

If you've invested good money into a hard drive system, it doesn't make any sense not to have a BACKUP routine. Especially if you're running your business on it! BACK/REST makes that job easier and saves much time. And **TIME IS MONEY**. *Don't delay! Save HOURS!* BACK/REST will work with all TANDY hard drives and most others that use LDOS or TRSDOS- 6, including the Tandy 15 and 35 Meg drives.

**MISOSYS, Inc.**

**PO Box 239**

**Sterling, VA 22170-0239**

**703-450-4181 (orders to 800-MISOSYS)**

# TRSCROSS™

(Pronounced TRISS-CROSS)

**TRSCROSS runs on your PC or compatible, yet reads your TRS-80 diskettes! Copy files in either direction!**

*The FASTEST and EASIEST file transfer and conversion program for moving files off the TRS-80™ and over to MS-DOS (or PC-DOS) or back*

**TRSCROSS™**  
Copyright 1986, 1987 by MISOSYS, Inc.  
All rights reserved

- 1 - Copy from TRS-80 diskette
- 2 - Copy to TRS-80 diskette
- 3 - Format TRS-80 diskette
- 4 - Purge TRS-80 diskette
- 5 - Display directory (PC or TRS-80)
- 6 - Exit

*Shown above is the Main Menu displayed when running TRSCROSS on your PC or compatible.*

*TRSCROSS is as easy to use as it looks to be!* The program is very straight forward, well thought out, and simple to operate. TRSCROSS has several "help" features built into the program to keep operation as easy as possible. Just pop your TRS-80 disk into your PC and copy the files right to your PC data disk or hard disk. *It couldn't be any faster or easier!* All steps are detailed in the instruction manual. Advanced features, for those that desire to use them include executing menu options right from DOS or from a batch file or macro. This can really speed up transfers when similar operations are performed frequently.

**TRSCROSS allows you to "TAG" all files to be moved in ONE pass!**

**TRSCROSS converts TRS-80 BASIC programs and SuperSCRIPSIT files in ONE PASS while COPYING to MS-DOS!**

No need to save your programs or files in ASCII or run a separate conversion program first before transferring. TRSCROSS reads your tokenized BASIC program or SuperSCRIPSIT files directly off your TRS-80 disk and performs the conversion all in ONE pass while being transferred directly to your PC or compatible computer. *Automatically* converts most BASIC syntax, and lines that need special attention can be listed to a printer. (Does not convert PEEKs, POKEs, graphics, machine language calls or sub-routines.)

**TRSCROSS will even FORMAT a TRS-80 disk right on your PC!** (Handy for those who use both machines!) Former TRS-80 users who no longer have their TRS-80, but still have diskettes with valuable data... this is exactly what you've been waiting for!

**TRSCROSS will READ FROM and COPY to the following**

**TRS-80 double-density formats:**

**TRSDOS 1.2/1.3, TRSDOS 6.2\*, LDOS 5.3\*,  
DOSPLUS, NEWDOS/80\*, & MultiDOS.**

DOS formats listed above flagged with \* signify that earlier versions of these DOS's are readable as well, but one or more sectors may be skipped due to a format problem in that version of the DOS. (Disks that were formatted with SUPER UTILITY™ or SUW4/4P™ do not have this problem.) TRSDOS 6.02.01, or higher should not have this problem. Disks formatted in any 5.25" 80 track format, or single density are not supported; 3.5" 720K disks are readable in a 720K 3.5" MSDOS disk drive.

TRSCROSS Requires: PC or compatible computer, 128K and a normal 360KB (40 track) PC or 1.2MB (80 track) AT drive. Double-sided operation is fully supported. If you have more than one disk drive, fixed drive, or RAM disk, operation will be much smoother. TANDY 1000 requires more than 128KB memory (DMA). TANDY 2000 is not supported at this time due to a difference in disk controller and floppy drives. "Special" data files (like PROFILE+) would need to be converted to ASCII on a TRS-80 first before they would be of use on a PC or compatible.

If you use both types of computers, or you plan to retire your TRS-80, this is for you. TRSCROSS will allow access to your TRS-80 diskettes for years to come. Copy your TRS-80 word processor data files as well as your Visicalc data files over to MS-DOS and continue using them with your new application.

**Only \$89.95**

Plus \$4 S&H (U.S.) or \$5 Canada or \$6 Foreign  
Virginia Residents must add appropriate sales tax.

**MISOSYS, Inc.**  
P.O. Box 239  
Sterling, VA 22170  
Phone: 703-450-4181 (Orders only: 800-MISOSYS)

# MISOSYS, Inc.

MISOSYS sponsors a forum on CompuServe: PCS49



**When you don't have to write in stone, don't let your editor weigh you down. You need SAID-86! Editing was never so easy!**

SAID-86 is a fast, flexible, full screen text editor for PC's. It is perfect for editing batch files, program listings, README files, CONFIG.SYS files, and anything you now do with EDLIN or the non-document mode of a word processor. Why struggle with huge editors; when all is said and done, SAID-86 will be your text editor of choice!

### Check out this list of features

- ✓ WordStar-like editing commands are easy to use
- ✓ Pull-down menu system for commanding SAID-86
- ✓ Supports nine editing buffers with automatic swap to disk
- ✓ Supports up to 30 user-defined macros; 255 characters each
- ✓ Undelete the last nine deleted lines can save your bacon
- ✓ MOUSE support with automatic recognition
- ✓ HELP facility; shell to invoke DOS commands from SAID-86
- ✓ SAID-86 can expand or contract TABs

**SAID-86 is reasonably priced at just \$29.95 + \$3S&H**

**MISOSYS, Inc.**  
P.O. Box 239  
Sterling, VA 22170-0239  
800-MISOSYS  
(US&Canada)  
or 703-450-4181  
M/C & VISA accepted.  
S&H are U.S. only.

### TRSCROSS

Now you can transfer TRS-80 Model III/4 files directly to your MS-DOS disks right on your PC. Convert BASIC programs; Convert SuperScript document files to DCA-RFT. Only \$89.95 + \$4S&H

**Why buy just a FAX board, when the ZOFAX 96/24 from MISOSYS includes a 2400 baud modem for a few bucks more? Turn your PC into a FAX machine!**

- ✓ Send and receive FAX from any CCITT Group III Fax Machine or PC Fax
  - ✓ Auto receive and print incoming Fax messages
  - ✓ Distribut Fax messages to multiple destinations
  - ✓ Time schedule transmission to take advantage of low nighttime rates
  - ✓ 2400 bps Fully Hayes Compatible Modem
  - ✓ Includes powerful but easy to use BITCOM and BITFAX software
- \$225 + \$6 S&H (\$200 when purchased with any other MSDOS product)**

### Speed up your program's computational execution with an IIT Math Coprocessor

IIT's CMOS coprocessors use less power, execute faster, support 4 x 4 matrix transformation, have thirty-two 80-bit numeric registers, come with a factory 5-year warranty, and cost less! If you use your machine for spreadsheets, desktop publishing, CAD, etc., consider an IIT coprocessor. For XT's or AT's with a math coprocessor socket. Installation instructions included. S&H is \$5.	IT-2C87-100	\$237.00
	IT-2C87-125	\$289.00
	IT-2C87-200	\$307.00
	IT-3C87-20	\$349.00
	IT-3C87-25	\$447.00
	IT-3C87-33	\$545.00
	IT-3C87SX-16	\$354.00
	IT-3C87SX-20	\$385.00

### EXPANZ!™ Disk Expander Card: On SALE!

With the new EXPANZ! data compression card, you can boost hard disk capacity up to three times. EXPANZ! plugs into any open slot and intercepts calls to and from the disk controllers. Compresses and decompresses in real time. Requires PC/XT/AT or compatible running DOS 3.x. Just \$125.00 + \$5S&H.

**Are you still fussing with floppies for BACKUP? CMS' DJ10 QIC40 tape drive from MISOSYS is your solution!**



The Colorado Memory Systems' JUMBO tape drive is one drive that fits all computers. It comes ready for internal use in AT's, XT's, and PC's, and connects to your floppy disk controller. The AB10 adaptor board can be used to connect JUMBO when all FDC connectors are in use. Kits are available which convert Jumbo to external use.

- ▶ In about 5.5 minutes, JUMBO backs up 10MB's file-by-file - the fastest in the industry! 40MB's gets backed up in about 18 to 20 minutes. Uses industry-standard DC2000 or DC2120 tape cartridges.
- ▶ JUMBO plugs into your floppy disk controller to save cost, power, and a slot. Needs 5-1/4" (or 3.5" with faceplate) mounting slot.
- ▶ Our tape adaptor board mounts in your host computer to provide an additional tape port — especially useful in 286 and 386 computers. When used with our external JUMBO, it lets you share your drive between computers. Note: external adaptor includes "Tape Adaptor"
- ▶ JUMBO has custom chips, high speed brushless motors, automatic circuits, no pots to vibrate out of calibration, and fewest parts to assure long life and technical leadership for years to come.

DJ10 Jumbo	\$275 (\$5S&H)
Tape Adaptor	\$75 (\$3S&H)
External Adaptor	\$110 (\$5S&H)
DC2000 tape (40M)	\$22.50
DC2120 tape (60M)	\$25.00

Note: DJ10 price includes one DC2000 tape.

**Now get the clearest view of your disk drive and memory contents with our DED86™**



### Powerful features in Version 2!

When you need to travel through your disk drive, why settle for a tool that isolates you from the road? DED86 gives you the direct controls you need to explore your disk. It's a full-screen sector-oriented disk/file editor and a page-oriented memory editor. When you want to "unerase" erased files, DED86's flexible KEEP facility does the job without you fussing over FATs.

### Check out this list of features

- ✓ Look by cylinder/head/sector, sector or cluster; even reserved/hidden sectors
- ✓ Scan free clusters to search for erased data
- ✓ Keep sectors & clusters for writing to a file; Great for partial file recovery
- ✓ Edit bytes in hexadecimal or ASCII, zap in Os; Undo last edit!
- ✓ Search your disk drive, a file, or memory for ASCII or hexadecimal strings
- ✓ Touch a directory file entry with your date and/or time
- ✓ Obtain complete disk statistics in one screen; includes usage data
- ✓ Alter file attributes: archive, system, hidden, read
- ✓ Save/Restore sectors to/from auxiliary buffers; for moving data around
- ✓ DOS subshell available while using DED86 lets you shell to other programs
- ✓ Handles 5.25" & 3.5" drives & RAM disks; even large partition drives

**DED86 is easy on your pocketbook; only \$29.95 + \$5S&H**

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

**800-MISOSYS**  
or 703-450-4181

# MISOSYS 20 or 40 MB Hard

1991 Prices currently in effect:

## Drives for your Model III or 4

### Complete Drive Kits:

20 Megabyte kit:	\$450
40 Megabyte kit:	\$575
Joystick option	\$20
LDOS software interface	\$30

### Piece Parts:

20 Meg drive (KL320)	\$200
40 Meg drive (ST-251-1)	\$320
Case & Power Supply	\$125
H/A with software	\$75
Xebec 1421 HDC	\$75
Adaptec 4010 HDC	\$75
Drive power Y cable	\$5
XT drive cable set	\$5

Note: freight charges are additional.

Prices subject to change without notice.

Our 15.5" x 7" x 5.25" (LWH) beige drive case has space for two half-height drives, 115V/230V 60 watt power supply and fan, hard disk controller (HDC), host adaptor, and a 50-pin SCSI female connector for the host interface. Our host adaptor sports a hardware real time clock. With its internal battery lifetime in excess of 10 years, never enter date and time again. A joystick port option adds a Kraft MAZEMASTER joystick with a port interface identical to the old Alpha Products joystick; thus, any software which operated from that joystick will operate from this one. Software provided with the host adaptor includes: a low level formatter; an installation utility and driver; a high level formatter used to add DOS directory information; a sub-disk partitioning utility; Utilities to archive/restore the hard disk files onto floppy diskettes; a utility to park the drive's read/write head; a utility to set or read the hardware clock; a keyboard filter which allows the optional joystick to generate five keycodes; and a utility to change the joystick filter's generated "keystroke" values after installation. Optional LDOS 5.3 software is available. 20MB drive packages are currently built with a Kalok 3.5" drive; 40MB packages use a Seagate ST251-1 28 millisecond drive. Drive packages are offered as 'pre-assembled kits', assembled to order and fully tested; all you will need to do is plug it in and install the software. 50-pin host interface cable included.

## PRO-WAM™ Version 2

### Window & Application Manager

Our applications turn your 128K Model 4 into a sophisticated business or personal machine because easily installed PRO-WAM comes with many useful and powerful menu-driven time savers and work organizers. PRO-WAM is accessed with a single keystroke; its export and import functions allow you to move data across windows between programs.

- Address CARDS, LABELS, and HEAD display & export
- BRINGUP tickler file; new PRINTING and sorting
- CALENDAR flags BRINGUP items visually on screen
- Ten 3 x 5 CARD files with FORMS and FIELDS
- Virtual PHRASE access for export
- New TODO list manager with "who does it"
- Plus many other vital applications!

PRO-WAM

M-51-025



## PRO-WAM

### Programmers' Toolkit

The Programmers' Toolkit provides all you need to know in order to write programs for the PRO-WAM environment. The package includes revised documentation which is in a convenient 5.5" by 8.5" format; the same as PRO-WAM release 2. The included WINLINK device driver supports programmed invocation of applications from any programming language environment which supports device I/O. Also included is a WINDOW/CCC function library for MC, and an assembler source code file for the PHRASE application to better illustrate how to write a PRO-WAM application.

Programmers' Toolkit

M-51-225

## PRO-WAM Application Pack

Mister ED is loaded with editor applications. All are full screen which make your editing jobs easy. Best of all, these are all PRO-WAM applications so they can pop up even when you are using other programs and applications

• DED edits disk sectors; • FED edits file records; and • MED edits memory pages (even alternate banks). All use a similar display screen and strikingly similar commands to enable you to edit anything. Get comfortable with one and you will know how to use all three of these editors.

• VED lets you edit the video screen with CARD-type editing. You get cut & paste; with this, you can easily use it as the clipboard facility found on more expensive systems.

• TED is just like the editor you get with LS-DOS 6.3; but ours works from PRO-WAM while you are using other programs! It's friendly, fast, and great for writing notes when you are right in the middle of a program you can't interrupt.

Mister ED

M-51-028

## Ribbon Cable Assemblies

We use a Cirris Systems tester for 100% test of shorts and opens on custom manufactured ribbon cables. Using switchable test assemblies, different kinds of cables can be assembled and tested for you. We can custom fabricate a cable according to your specifications or provide standard replacement cables; these all use unshielded ribbon cable. Cables can use: DB-9 and DB-25 M/F; 20, 34, 40, or 50-pin header; 34 or 50-pin edgecard M/F; 36-pin printer; 50-pin SCSI M/F.

We make replacement FDC-Floppy cables @ \$20 for all TRS-80 Model 4's (4, 4D, 4P); printer cables @ \$20 for Model III/4 or II/12/16/6000; RSHD primary-secondary interconnect cables @ \$20/set; RSHD host cable @ \$20. Need a replacement RS232 or printer cable? We also stock KelAM 34-pin M/F edgecard connectors (\$13/pair).

## Floppy Drives and Accessories

5.25" 360K 1/2-height	\$75+\$5S&H
3.5" 720K in 5.25" 1/2-height frame	\$85+\$4 S&H
2SV5 drive case & P/S	\$60+\$6 S&H
Single drive host cable *	\$12.50
Dual floppy extender cable **	\$18
Our Model 2SV5 dual vertical external floppy disk drive case will hold two 5.25" half-height disk drives or one standard height drive. * needed for one or two drives; ** needed for two drives. S&H prices are UPS ground to continental U.S.	

## MISOSYS, Inc.

P. O. Box 239

Sterling, VA 22170-0239

703-450-4181

orders to 800-MISOSYS (647-6797)

VISA and MasterCard accepted

Virginia residents please include 4.5% State sales tax



# Let our LB Data Manager solve your data storage problems

***LB Version 2: A Flat File Data Manager with more powerful and easy to use features in this latest enhancement of Little Brother!***

We've added many features asked for over the past few years by LB users; yet LB is still just about the easiest, most flexible data manager you can use for managing your data. Absolutely no programming is needed to create a database with numerous fields, construct input screens for adding and editing data, and create your own customized report. Quickly you define your data fields in response to LB's prompts, and then draw your data input screen using simple keystrokes. In no time at all, you're entering data. Customize your printed reports with user-definable print screen definitions. LB is just what you need in a data manager!



## **Data capacity per database:**

LB supports up to 65,534 records per data base; 1,024 characters (64 fields) per record; and up to 254 characters per field.

## **Field types supported:**

LB allows ten field types for flexibility: *alphabetic* {A-Z, a-z}, *calculated* {operations on "numeric" fields using +, -, \*, /}, *date last modified* {YYYY/MM/DD automatically maintained}, *dollar* {±ddddddd.dd}, *floating point* {±ddddddd.ddddddd}, *literal* {any ASCII character}, *numeric* {0-9, -, .}, *right-justified numeric* {flush right numeric}, *upper case alphabetic* {A-Z, automatic conversion of a-z}, and *upper case literal* {literal with automatic conversion of a-z}. All field types utilize input editing verification so invalid data cannot be added to a record. Field name strings can be up to 19 characters long.

## **Data entry and editing:**

LB allows you to design up to ten different input/update screens to provide extreme flexibility for selectively viewing your database fields. Using a database password provides the capability of selectively protecting fields from being displayed or printed without entry of the correct database password, or they can be protected from being altered. This is quite useful in a work-group environment. Fields may selectively be established to require a data entry before a record being added or edited is saved. You can enable a special index file to keep track of records being added. This can be subsequently used, for example, for a special mailing to newly added *customers*. Flexible editing includes global search and replace with wild-card character match and source string substitution. Search and replace can be performed on all records, or on records referenced in an unsorted or sorted index file.

## **Record selection and sorting:**

You can maintain up to ten different index files to keep your data organized per your multiple specifications; keep one alphabetic, another by address. Records may be selected for reference in an index file by search criteria using six different field comparisons: EQ, NE, GT, GE, LT, and LE. You can select on up to eight different fields with AND and OR connectives. Index files can be left unsorted, or you can sort in ascending or descending order. By attaching a sorted index file, any record may be found within seconds - even in a very large database.

## **Automatic operation:**

For automating your processing needs, LB can be run in an *automatic* mode, without operator intervention. Frequently used procedures can be saved by LB's built-in macro recorder for future use. Entire job streams may be produced, so that LB operations may be intermixed with literally any DOS function that can be *batch* processed.

## **Report generation:**

Report generation incorporates a great deal of flexibility. Your report presentation is totally customized through print definition formats which you define on the screen as easily as you define the input/update screens. You can truncate field data or strip trailing spaces. You control exactly where you want each field to appear. LB provides for a report header complete with database statistics: database name, date, time, and page numbers. A report footer provides subtotaling, totaling, and averaging for dollar, floating point, and calculated fields; print number of records printed per page and per report. Attach any of the ten index files and you control exactly what records get printed; even a subset of indexed records can be selected for printing to give you a means of recovering from that printer jam halfway through your 30-page printout. You can even force a new page when the key field of an index file changes value. Up to ten different printout definition formats can be maintained for each database. Reports may be sent easily to a printer, the console display screen, or to a disk file - useful for subsequent printing or downstream data export to other programs. Report formatting allows for multiple across mailing labels, multiple copies of the same record, or even printing one record per page for sales books. You can easily generate mail/merge files of address or other data for your word processor. Or you can use LB's built-in form letter capability.

## **Maintenance utilities:**

To make it easy for you to grow your database as your data needs grow, we provide two utility programs for managing your database. One allows you to construct a new database with an altered data structure and populate it with data from your existing database. Another allows you to duplicate your database structure, copy or move records from one to another, or automatically purge un-needed records.

## **Help is on the way:**

The main menu even provides a shell to DOS so you can temporarily exit LB to perform other DOS commands. LB provides extensive on-line help available from almost every sub-command. A 200-page User Manual documents every facet of LB's operation.

Specify MS-DOS or TRS-80 version. LB is priced at \$99 + \$5 S&H (US; \$6 Canada; \$7 Europe; \$9 Asia, Pacific Rim, and Australia). To upgrade from version 1.0, send Table of Contents page and \$40 +S&H. Remit to:

**MISOSYS, Inc.**

PO Box 239

Sterling, VA 22170

703-450-4181 or orders to 800-MISOSYS



# LS-DOS 6.3.1: The latest for your Model 4

- ☆ The DATE command, "Date?" prompt on boot, and the @DATE SVC now support a date range of 32 years; from **January 1, 1980 through December 31, 2011**.
- ☆ **Enable or disable the printer time-out** and error generation with SYSTEM (PRTIME=ON|OFF)
- ☆ Customize the display of the time field in the DIR command to display **12-hr or 24-hr clock time** with SYSTEM (AMPM=ON|OFF).
- ☆ Both ASCII and hexadecimal display output from the LIST command is **paged a screen at a time**. Or run it non-stop under your control.
- ☆ MEMORY displays (or prints) the status of switchable memory banks known to the DOS, as well as a **map of modules** resident in I/O driver system memory and high memory.
- ☆ Specify SYSTEM (DRIVE=d1,SWAP=d2) to **switch drive d1 for d2**. Either may be the system drive, and a Job Control Language file may be active on either of the swapped drives.
- ☆ The TED text editor now has commands to **print the entire text buffer**, or the contents of the first block encountered. Obtain directories from TED, too!
- ☆ Have extended memory **known to the DOS**? The SPOOL command now permits the BANK parameter entry to range from 0-30 instead of 0-7.
- ☆ **Alter the logical record length** of a file with "RESET filespec (LRL=n)"
- ☆ Specify "RESET filespec (DATE=OFF)" to restore a file's directory entry to the old-style dating of pre-6.3 release. Specify "RESET filespec (DATE=ON)" to establish a file's directory date as that of the **current system date and time**.
- ☆ Felt uncomfortable with the *alleged* protection scheme of 6.3? **LS-DOS 6.3.1 has no anti-piracy protection!** MISOSYS trusts its customers to honor our copyrights.
- ☆ Best of all, **a 6.3.1 diskette is available as a replacement for your 6.3.0 diskette for \$15** (plus \$2 S&H in US). There's no need to return your current master.
- ☆ The 6.3.1 diskette comes with a 30-day warranty; written customer support is available for 30 days from the purchase date. Versions for the Model 4 and Model II/12 are available. **If you do not already have an LS-DOS 6.3.0, order the 6.3.1 Upgrade Kit with 90 days of customer support for \$39.95 (+\$3 S&H).**

MISOSYS, Inc.  
P. O. Box 239  
Sterling, VA 22170-0239  
703-450-4181

[orders to 800-MISOSYS (647-6797) from U.S. or CANADA]

**LB Data Manager Version 2.1**  
**Upgrade now available**  
**See Details Inside!**



MISOSYS, Inc.  
PO Box 239  
Sterling, VA 22170-0239  
U.S.A.

Contents: Printed Matter

**BULK RATE**  
**U. S. POSTAGE**  
**PAID**  
**Sterling, VA**  
**PERMIT NO. 74**

Attention Postmaster: Address Correction Requested  
Forwarding and return postage guaranteed