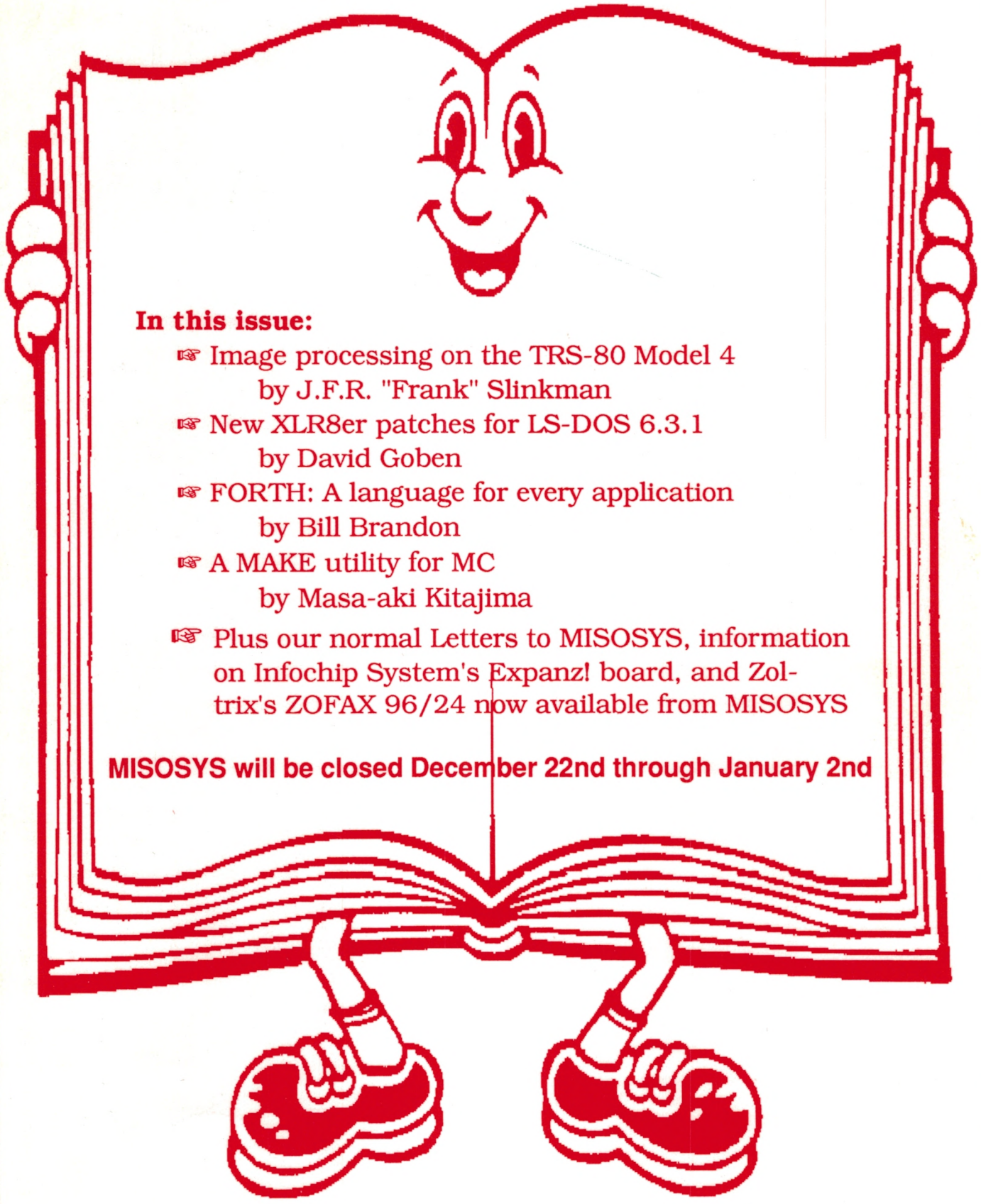
THE MISOSYS QUARTERLY



In this issue:

- ☞ Image processing on the TRS-80 Model 4
  by J.F.R. "Frank" Slinkman
- ☞ New XLR8er patches for LS-DOS 6.3.1
  by David Goben
- ☞ FORTH: A language for every application
  by Bill Brandon
- ☞ A MAKE utility for MC
  by Masa-aki Kitajima
- ☞ Plus our normal Letters to MISOSYS, information on Infochip System's Expanz! board, and Zoltrix's ZOFAX 96/24 now available from MISOSYS

MISOSYS will be closed December 22nd through January 2nd

Volume V.ii **$10** **Winter 1990/1991**

HAVE A DEVIL OF THE TIME WITH MODEL I/III(4) GAMES FROM MISOSYS

Savings up to 60% off until March 31st

It's BOUNCEOIDS...

Utilizing your newly issued class 4 annihilator, you must reduce the massive BOUNCEOID BOULDERS to cosmic dust! BEWARE! They bounce off of walls like giant pool balls, and the smallest touch means instant death! And as an added complication, terrible shaking bugs come slithering out of null space to crunch you into bug lunch! If you can prove yourself worthy, you will be challenged by the mysterious alien invaders in a nerve grinding test of incredible coordination and targeting skills. Will you survive? **Just $7.48 until March 31st**

Become a "Crazy Painter" and create a masterpiece ...if you can

A mischievous puppy, snakes and poisonous turpentine buckets force changes in your painting. In higher skill levels, "paint eaters" start chewing up your work. Suddenly, everyone's a critic. Can you overcome them? And on top of it all, in the Exclusive Challenge Mode, you have to catch an army of runaway puppies. Crazy Painter is joystick compatible. **Just $7.48 until March 31st**

The Official FROGGER ™ by SEGA

The popular arcade game for TRS-80 Models I & III.

Just $7.48 until March 31sth

SCARFMAN

Who are the brave, who dare to enter the maze where the monsters live. SCARFMAN features quality GRAPHICS, SOUND, REAL TIME MONSTERS, and hours of ENJOYMENT. Nine progressive skill levels make the game more difficult as you get better. Do you have what it takes to reach skill level 9? **Just $7.48 until March 31st**

SPACE CASTLE

Will you save the Andromeda Galaxy by destroying the Space Castle or will the evil warlord Yugdab continue to rule, uncontested? Your main defense is your ability to skillfully handle your ship and it's projectiles. If you are not careful, Yugdab's intelligent mines will hunt you down and blow you into space dust. Space Castle has fast action and exciting sound. **Just $7.48 until March 31st**

The Gobbling Box

This fast-paced action **arcade-type game** runs on the TRS-80 Model I, III, and 4/4P/4D. It generates a variety of special sound effects and music which complement the action on the screen. The arrow keys or Alpha Products joy-stick control the movements of the GOBBLER in this game. You want your GOBBLER to eat as many dots as possible, while trying to avoid the ZONKERS who won't stop chasing your GOBBLER until one of them eats it or until the GOBBLER eats all dots on the GameBox. The GOBBLER's reward is a new Box; there's 3 in all. The GOBBLER can tame the ZONKERS for a short while by eating one of the ENERGIZERS on the board. Then it's the GOBBLER's turn to chase, catch and eat the ZONKERS. The game has two skill levels; the pace is fast; the sound is great; the action is continuous. You can't beat this bargain of a game. **Just $9.98 until March 31st**

MISOSYS, Inc.
P.O. Box 239, Sterling, VA 22170-0239
703-450-4181 or 800-MISOSYS
U.S. & Canada S&H: $2 (1) $3 (2-4) $5 (5+); Others add $2

Leo Christopherson's Animated Game Disk with sound

This is one of the greatest values in games ever produced. Leo Christopherson wrote the very first animated game for the TRS-80 and the country went wild for it. Android Nim will make everyone laugh to watch these life-like creatures as they shake their heads up and down or side to side and blink at you stupidly as they wait for you to make a move. Then Leo invented how to make the TRS-80 produce sound and added it to NIM. You get ALL of Leo's programs on one disk for one great low price. If you haven't yet bought this disk, do it now! It is a classic! Each one of these games originally sold for $9.95 - $19.95 EACH. The disk includes the famous games: ANDROID NIM, BEEWARY, DUELING DROIDS, DANCING DEMONS, SNAKE EGGS, and ANIMATED LIFE. All games feature full sound effects and some of them are even in 3-part harmony! You and your family will just love this disk! Dancing Demon even features saving your song and dance routines to disk and four of them are included! The possibilities are endless and it is always entertaining. A great way to "show off" what your computer can do and always fun. **Just $9.71 until March 31st**

KIM WATT's Space Colony, Symon, Capture, Horse Race

Kim watt, author of Super Utility and other famous programs wrote some games that Adventure International published back in the early 70's. They are interesting in that most people have never seen these, as by the time SUPER UTILITY was established in the market, these games, as well as Adventure, were gone. Originally these were sold on three separate diskettes (or tapes), but we have combined them all on one disk for you collectors. **Just $7.16until March 31st**

LANCE MICKLUS' GREATEST GAMES -3 Disk Set!

This is a great collection as it features space games (Space Trek), adventure games (Dog Star Adventure), gambling games (The Mean Craps Machine, which also includes a Craps tutorial booklet on disk), board games (Mean Checkers Machine), as well as some darn useful programs that you might use for real purposes. Also has some educational games for the kids. You will enjoy owning this set of programs by one of the early pioneer programmers for the TRS-80. **Just $10.77 until March 31st**

Lair of the Dragon, by David Goben

Lair of the Dragon MegAdventure is unlike virtually any other interactive fiction adventure that you have ever played, for it will more than just paint its pictures upon the canvas of your imagination - it will slap the sweat right onto your forehead! If you truly believe that discovery is one of the finest points in life, if you would like to test your ability to think logically to the fullest extent of your ability, if you would like to take on the largest adventure ever written in the genre of interactive fiction, and if you have the guts to face that which would make any other mortal elf cringe in fear, then ***Lair of the Dragon*** is your cup of poison; for reward is a hard-earned commodity here, not given easily to the timid and the faint-hearted. If you are an old hand at adventuring, then be pre-pared for a worthy opponent. MegAdventure rips the door to adventure right off its hinges! **Just $11.98 until March 31st**

19-12-90 2-1-91 3-77

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 5 Megabytes of RAM, a Seagate ST4096 80-Megabyte hard drive, a Colorado Memory Systems 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 WORD Version 5.0; Submissions on paper and letters are scanned and converted to text using ReadRight optical character recognition software by OCR Systems. Final page composition is developed using PageMaker 3.0 by Aldus. Cover art and clip art comes from CLIPPER, a product of Dynamic Graphics.

Table of Contents

The Blurb 2
- Points to Ponder 2
- Tandy 1000 TL/2 and hard drives 2
- Infochip's Expanz! data compression board 4
- Earlier TMQ's available 5
- PD Software Librarian 6
- TMQ Advertising 6
- DISK NOTES 5.2 6
- Ribbon cable assemblies 6
- IIT Math Coprocessors 7
- Zoltrix ZOFAX 96/24 FAX/modem 7

Letters to the MISOSYS 8
- BOOT/SYS and DIR/SYS 9
- DOS Directory Structure 10
- EnhComp FIELD error-checking bug 13
- DO bugs? Workaround for extended command lines 14
- Bargain Floppy Drives? 15
- LDOS 5.3 BACKUP bug 16
- What's the best word processor? Still a question! 20
- Merging LB data bases 22
- PC Tape Backup drives 23
- Ignore some TRSCROSS errors 24

Inside TMQ 25
- Image processing on the TRS-80 Model 4, by J.F.R. Slinkman 25
- Nedw XLR8er patches for LS-DOS 6.3.1, by David Goben 33
- FORTH: A language for every application, by Bill Brandon 36
- A MAKE utility for MC, by Masa-aki Kitajima 40

List of Advertisors

Computer News 80	44
Javna Management Company	32
MISOSYS, Inc.	IFC,45-48,IRC,RC
Pacific Computer Exchange	39
TRSTimes magazine	43

List of Patches in this Issue

Fix to PRO-EnhComp to correct FIELD error checking	13
Fix to EnhComp to correct FIELD error checking	14

Points to Ponder

Metric update: Normally The Blurb is entirely my column - a soap box, so to speak; letters to the editor are usually relegated to our "*Letters*" section. However, Lawrence Rossiter, from Victoria, BC, Canada, sent me the following piece which I felt merited placement herein. Here's what Lawrence had to report concerning metrics.

Dear Roy; With reference to the item in your Blurb in the Fall issue of *TMQ* headed 'Metric update' and prediction that Americans will be buying gasoline measured in liters by the turn of the century, here is a quote from an editorial in Locomotive Engineering for April, 1896, concerning the same subject. "*For the last thirty years the advocates of the French system of weights and measures have been laboring unceasingly to make the metric system the only legal one in this country, and there appears reason to believe that they will eventually succeed in their self-imposed mission.*" A few sentences further on: "*Now comes a report from Washington that the House Committee on Coinage, Weights and Measures, have recommended that the metric system of weights and measures shall be adopted by the various departments of the Government by July 1, 1892, and by the nation at large January 1, 1901.*" The editor goes on for another three columns before ending with: "*The time may come when sentiment will rise so far above business considerations that a change will be made from the inch to the metric unit. But it is certain that the cost of that luxury will run into many millions of dollars.*" So here you are, almost 95 years later, not even as close to the joys of metric measurements as you were then!

Incidentally, each monthly issue of the Locomotive Engineering (published in New York) averaged over 90 pages and the annual subscription rate was $2.00, postage paid to any part of the world!

A tip of the hat to Lawrence for that input. As an aside, my prediction referenced the turn of the century: a point in time which some folks still are unclear of the specific date!

The Blurb by Roy Soltoff

Where's Tandy Department: It wasn't that many years ago when the industry press refused to acknowledge the presence of Tandy in the computer marketplace. Statistics on computers sold invariably did not list sales of Radio Shack computers. But that's no more. Tandy's efforts, at least in the PC-compatible world, have been recognized by the media.

A recent issue of *Computer Reseller News* presents the following tally of computer stores in the United States:

Tandy	7,034
Intertan	2,481
Intelligent Electronics	987
Computerland	902
Microage	732
Software Etc.	223
Egghead	190
Babbage's	168
Electronics Boutique	161
Inacomp	109
Businessland	96
Compuadd	89
Computer Factory	66
Software City	64
Waldensoftware	37
Computercraft	23
Soft Warehouse	10

Yes, I know, Intertan is the international spin-off from Tandy, so what's it doing in this list?

FAX Chips: Prices continue to tumble in the fax board market as large scale integrated electronics continues to decrease the real estate needed to put together a board. Note the low cost of the fax/modem board currently available for MISOSYS.

Now comes news of a 2-chip fax/data chip set from Exar Corp which implements V.29 facsimile and V.22 bis data with Microcom Network Protocol 5 (MNP-5) and the emerging V.42/V.42bis standards. According to Exar, all you need to complete a fax/modem board is a dedicated microcontroller such as an 8051, 32 kbytes of static RAM, and a 256k EPROM. And if you buy the chip set in 1000 piece quantities, the cost is down to $36.82 a set!

Winter vacation reminder

MISOSYS closes up between Christmas and New Year's. That means this year, we will be closed from Saturday December 22nd until Wednesday January 2nd, 1991.

TMQ Schedule

Why do I bother to state targets. If you have a 4-issue subscription, you'll get four issues. I try to **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** (it didn't get out until December).

Note that your mailing label usually has the expiration date of your subscription. For instance, **those with "91/02" 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. It is best to just extend your **91/02** subscription for **two additional issues** for one-half of the normal price: A=$12.50; B=$15.00; D=$17.50; E=$20. That will adjust your expiration date to 91/08, good through issue V.iv.

As I write this sentence, it's Friday the 30th of November. This issue will be off to the printers next week.

1000 TL/2 and hard drives

Due to some misinformation about the *alleged* inability of our TRSCROSS product to successfully operate on a Tandy 1000 TL/2 computer, MISOSYS recently

Table I

Pin	Name	Description
01	Reset	Reset signal from host
02	Ground	
03-18		Host data 0-15
19	Ground	
20	Key	Unused pin, clipped
21	reserved	
22	Ground	
23	/HIOW	Host I/O write strobe
24	Ground	
25	/HIOR	Host I/O read strobe
26	Ground	
27	Reserved	
28	Host ALE	Host address latch enable
29	Reserved	
30	Ground	
31	IRQ14	Interrupt request drive to host
32	/HOST I/O16	When active, it indicates to host that the 16-bit data register is addressed
33	ADDR1	Host I/O address line 1
34	/PDIAG	Passed diagnostics
35	ADDR0	Host I/O address line 0
36	ADDR2	Host I/O address line 2
37	/HOST CS0	Host I/O chip select
38	/HOST CS1	Host I/O chip select
39	/HOST SLV/ACT	Active when drive is executing a command
40	Ground	

Table II

Pin	Name	Description
01	Reset	Reset signal from host
02	Ground	
03	SO7	data 7
04	Ground	
05	SO6	data 6
06	Ground	
07	SO5	data 5
08	Ground	
09	SO4	data 4
10	Ground	
11	SO3	data 3
12	Ground	
13	SO2	data 2
14	Ground	
15	SO1	data 1
16	Ground	
17	SO0	data 0
18	Ground	
19	Ground	
20	Key	Unused pin, clipped
21	AEN	Host address latch enablereserved
22	Ground	
23	/HIOW	Host I/O write strobe
24	Ground	
25	/HIOR	Host I/O read strobe
26	Ground	
27	DACK3-	
28	NC	
29	DRQ3	
30	Ground	
31	HDIRQ	Interrupt request drive to host
32	NC	
33	SA1	Host I/O address line 1
34	NC	
35	SA0	Host I/O address line 0
36	NC	
37	HDCS-	Host I/O chip select
38	NC	
39	NC	
40	Ground	

acquired such a machine. For the home or small business user, this appears to be an excellent machine to get started on. The 1000 TL/2 modestly sports an 8 Mhz 80286 processor and a single 3.5" floppy. It has the provision for another 3.5" form factor device and a 5.25" 1/2 height device. After ensuring the successful operation of TRSCROSS to read/write TRS-80 Model III/4 media files, I proceeded to install a hard drive into the TL/2.

The TL/2 supports the IDE drive, which incidentally, has been known to be an acronym for at least two different statements: Integrated Drive Electronics, and Intelligent Drive Electronics. I popped a 40 megabyte IDE Seagate ST157A drive into the 3.5" slot; I could not get it to work. I called Tandy support in Fort Worth and they were unable to offer assistance because the ST157A wasn't their drive. I called Seagate customer support who had no experience with the Tandy 1000 TL/2. So I proceeded to investigate the situation on my own behalf. I ordered the Tech manual for the 1000 TL/2 from Tandy National Parts, and talked Seagate out of an *AT Interface Manual* which documents the IDE interface. While waiting for the manuals, I acquired an AT-compatible IDE host adaptor, a Seagate ST07A. I then replaced the hard drive and MFM controller in my 386 machine with the IDE host adaptor and ST157A hard drive. The drive formatted, partitioned, and worked correctly, so I knew nothing was wrong with the hardware. When the documentation on the IDE interface came in, here's what I found. The AT Interface Manual presented the IDE pinout listed in Table I.

The TL/2 Tech Manual doesn't provide an easy-to-read table showing its IDE bus, but rather provides a difficult to read pinout in the schematic. Nevertheless, what I was able to discern there is listed in Table II.

I thought it's IDE interface was an industry standard hard drive interface. Here's what the 1000 TL/2 manual has to say about the TL/2's hard drive interface: "Intelligent hard disk drives include logic circuitry compatible with the computer's built-in IDE; simply plug the hard drive's cable into the on-board IDE connector." Now I know an industry standard when I see one; and IDE was supposed to be a standard. No where in the manuals which come with the TL/2 does it mention that it's IDE interface is different from the industry standard. The Radio Shack 1990 catalog on the TL/2 does not provide any clue. But it is a fact that the 1000 TL/2 has an 8-bit IDE interface! So let that be a lesson to anyone with a 1000 TL/2; it doesn't have an industry standard IDE interface. At this point, I don't know of any other manufacturer of PCs which is using "IDE" as a nomenclature for anything but the standard AT-compatible interface. Of course, since Tandy doesn't manufacture their own drives, some company should have the Tandy drive available.

In any event, I wound up installing a Kalok KL320 20 megabyte 3.5" drive and fed it with a Western Digital WDXT-GEN controller. That worked, out fine giving the 1000TL/2 20 megs of storage. I passed it to Brenda for upstairs use. I then gave her the challenge of becoming expert with Deskmate!

Infochip's Expanz! data compression

Here's a rundown on the newly released Expanz! board for PC compatibles. I have had this board installed in my AST 386 machine for over a month now. Normally retailing for $199, I am selling this board for $179.10 plus shipping. Okay, so what does it do?

I first read about the Expanz! board earlier this year in a few industry magazines covering the electronics marketplace. 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 IC-105 uses a proprietary compression algorithm which is claimed to provide *lossless* compression of from 2:1 to 15:1 relative to the mix of data compressed. "Lossless" is a term which means that the decompressed data recovered is 100% identical to the original data. The algorithms typically employed by archive utilities such as ARC and PKARC provide lossless compression.

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. It accomplishes this by what appears to be some sleight of hand.

I recognized the value of a hardware compression/decompression coprocessor for use in many systems bumping up against the limit of existing disk capacity. Changing a 20 or 40 Megabyte drive to a larger one sometimes can be more of a headache as the possibility of also requiring a change in hard disk controller becomes evident. So an inexpensive hardware solution which is relatively transparent to the user can be quite attractive. I like to look at leading edge technology - but also inexpensive to implement. Since I want to be relatively first in this regard, I jumped in. But I did take advantage of my Colorado Memory Systems tape backup drive to make two backups of my 80 Megabyte drive currently in my AST 386 machine.

Adding an Expanz! card to your system is relatively painless. However, 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 have to 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 preprinted 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.

Using release 1.0 of the provided software, I eventually found out that I was unable to boot after compressing drive C: - my boot partition. This problem was quickly resolved by the release of version 1.1 of the software. The SAFETY disk and my tape backups provided for a faultless recovery.

The Expanz! card implementation is not without its demerits; its 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. After installing the board and its driver, I had two programs I could no longer run. ReadRight, from OCR systems needs about 570 kbytes of free memory. Taking the 44k away from what was left after loading MS-DOS 3.3, DOS buffers, mouse driver, and scanner driver, I didn't have sufficient memory left. Nor was I able to run my scanner program out of WINDOWS since there was insufficient buffer memory left.

I eventually 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. In fact, I'd probably start carrying the product at a reasonable discount if my customer base expressed an interest. QEMM386 also comes with Manifest, a utility which beautifully presents the memory utilization of your system.

Getting back to the Expanz! board, let me show you an illustration of what you can achieve in expanded disk capacity. My E: drive is the second 32 megabyte partition on my 80 megabyte drive. The before and after statistics are as follows:

	Before	**After**
Files	496	496
Bytes	24,195,072	10,807,808
Free	9,123,840	22,195,712
Total	33,435,648	76,072,960

The Expanz! board includes an IDISK utility which replaces CHKDSK. This tool provides similar recovery functions and also provides additional information on the compression in effect. For instance, after compressing my C: drive, IDISK provided this display:

```
Logical Disk Allocation
    100,311,040 Maximum vol capacity
     56,832,979 projected vol capacity
     34,144,256 used
Physical Disk Allocation
    19,746,304 bytes used
    13,275,648 bytes available
    1.72:1 compression ratio
```

Based on this data, one can intuit the sleight of hand used to accomplish random access of compressed files. When you ICONVERT a partition, you specify a compression factor; the default is three. It certainly appears to me that the Expanz! driver then reconstructs the drive

with a logical sector size one third the original sector size. If it can compress each original sector to one third, then it still retains a one to one relationship in file sector access. Those that it can't are probably linked to another logical sector known to the Expanz! software. The maximum volume capacity shown above is the original drive partition size multiplied by the chosen compression factor. The projected volume capacity is the expected capacity which would be realized if the additional files added to the drive were at the same mix of compressibility as the existing files; in this case, the mix is 1.72:1 on my C: drive. But this partition contains a lot of .EXE files which are poor candidates for compression. IDISK statistics for my other two E: and F: partitions are:

```
Logical Disk Allocation
   100,311,040 maximum vol capacity
    72,765,440 projected vol capacity
    27,754,496 bytes used
Physical Disk Allocation
    12,609,536 bytes used
    20,393,984 bytes available
      2.20:1 compression ratio

Logical Disk Allocation
    38,670,336 maximum vol capacity
    18,806,511 projected vol capacity
     8,763,392 bytes used
Physical Disk Allocation
     5,877,248 bytes used
     6,732,800 bytes available
      1.49:1 compression ratio
```

Among the three partitions of my 80 megabyte drive, the Expanz! board turned the drive into almost a 150 megabyte drive.

At close to presstime, Infochip Systems forwarded me their version 1.2 release which was supposed to fix a few things. One thing it was supposed to do was have the ability of being loaded into high memory by QEMM 386. That would be a plus for me; however, installing every other driver into high memory using QEMM 386 has given me no further memory conflicts.

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 - and for me, it's no longer a drawback. Although I haven't done timing tests before and after installation, I have not perceived any degradation in disk I/O. Note also, that you can also expand floppy disk drives and RAM disks. It's not a bad idea to compress floppies if that's your only backup medium. 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. They are in stock here at MISOSYS.

Here's a few questions and answers concerning the board which traversed our MISOSYS CompuServe forum recently.

Question: How does the EXPANZ! board work?

Answer: That's a good question. It appears to alter the sector size; thus, compressing a normal 512-byte record into a smaller size. That appears how it can handle random access. The board requires some space for a DOS region driver (within the 640K) and 16K of himem.

Question: What kind of slot does the Expanz! board take up?

Answer: The Expanz! board is a short card which plugs into an 8-bit bus slot on a PC, XT, AT, or 386 machine. It requires the use of MS-DOS 3.x.

Question: There's a message on IBMHW which indicates that the EXPANZ! board will not work with MSDOS 4.0. Also, disk utilities such as Norton and others will not work. The people who make EXPANZ! is working on their own disk utilities.

Answer: That's correct; the EXPANZ! board works only with DOS 3.x at this time - and not Compaq's nor Zenith's; it is expected that Compaq's DOS and MS-DOS 4.01 will be supported in the Spring of 1991. Also, disk utilities will not work. Also, you cannot use de-fraggers. There's going to be some trade-offs. Some of the disk utilities only work with 512-byte sectors. DED-86 is one of them. I was not terribly pleased that I cannot use DED-86 since I use it a lot for disk exploration - but that may be surmountable if the EXPANZ! board uses a fixed logical sector size - seems like it must.

Question: In light of findings that the sector sizes are altered, I wonder how this will screw up disk caches.

Answer: It works with WINDOWS 3.0's Smartdrive - which is a disk cache. As I see the implementation of the EXPANZ! board, the physical sector size remains unchanged - but it appears to employ some sort of "logical sector".

Earlier TMQ's available

For out of print issues (Volume I and Volume II), we are providing back issues of *THE MISOSYS QUARTERLY* via copier reprint at **$12.50** plus $2.75 S&H in the U.S. and CANADA. For foreign zone D, the S&H rate is $5.50; zone E is $6.50. The price for regular back issues still in print is $10 + S&H. **Special deal now in effect for either all four issues of Volume III, or all four issues of Volume IV: just $24 per set + S&H $5 (US), $6 (CAN), $14) ZoneD, $20 (ZoneE).** Here's a synopsis of past issues:

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

III.ii Getting into computer math Part2; 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; SuperScripsit document file format' FELSWOOP PRO-WAM export utility.

IV.iv Five Twelve K: A better way; Multi-Command; Touch/ccc; 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.

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 redistributing 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.

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:

Full page	**$125**
Half page	**$75**
Quarter page	**$50**
Ninth page	**$20**

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*, give me a call.

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.

DISK NOTES 5.2

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 5.2 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 is priced at $10 **Plus S&H**. The S&H charges are $2 for US, Canada, and Mexico, $3 elsewhere.

XLR8er boards

I am down to my last seventeen XLR8er boards as of this writing. I have no replacement shields which are needed to install a board in a 26-1069 (non-gate array) Model 4. However, David Dahlager has informed me he has a few spare standard shields which he may be able to modify; more on that later. Folks with a gate array model 4 (26-1069A or 26-1070 4D) can install an XLR8er board using the "graphics card" method which is behind the motherboard. Since we need to archive a few boards for warranty replacement purposes, if you want to purchase a board, now is the time to do it.

Ribbon Cable Assemblies

MISOSYS uses a Cirris Systems cable tester for 100% testing of custom manufactured ribbon cables. My assortment of test assemblies supports 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 edgecard M/F, 36-pin printer, 50-pin edgecard F, 50-pin SCSI M/F, 34-pin Header M/F, as well as DB9 Male. This includes such things as replacement FDC-Floppy cables @ $15 for all TRS-80 Model 4's (4, 4D, 4P); printer cables @ $15 for Model III/4 or II/12/16/6000; RSHD primary-secondary interconnect cables @ $15/set; RSHD host cable @ $15.

Kel-AM 34-pin male edgecard @ $8
Kel-Am 34-pin female edgecard @ $5

PC-Compatible Product News

IIT Math Coprocessors

Integrated Information Technologies (IIT) makes CMOS coprocessors compatible with Intel's 80x87 instruction set; these processors use less power - great for laptop application, 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! My 386-20 AST now sports an IT-3C87-20. If you use your MS-DOS machine for numeric work (spreadsheets, desktop publishing, CAD, etc.) and don't have a coprocessor installed, consider an IIT coprocessor from MISOSYS. Here's the skinny on pricing:

IT-2C87-100	**10 Mhz AT**	**$237**
IT-2C87-125	**12.5 Mhz AT**	**$289**
IT-2C87-200	**20 Mhz AT**	**$307**
IT-3C87-20	**20 Mhz 386**	**$349**
IT-3C87-25	**25 Mhz 386**	**$447**
IT-3C87-33	**33 Mhz 386**	**$545**
IT-3C87SX-16	**16 Mhz SX**	**$354**
IT-3C87SX-20	**20 Mhz SX**	**$385**

2C87 parts are for AT-class machines using an 80286 processor; 3C87 parts are for 386 machines using an 80386 processor; 3C87SX parts are for machines using an 80386SX processor. S&H is $5 in U.S. If you want genuine Intel coprocessors, I can obtain them for about 30% more. But why pay more for less?

ZOLTRIX FAX/Modem

Those of you who have been following my product line may be aware that I have been looking for a reasonably priced PC FAX board. Recognizing that Group III fax is a standard, and noting that the entire protocol has been fabricated into a single-chip, I was waiting for advanced technology to bring the price down from the $700-$1000 price fax boards sold but a year ago.

My requirements for a fax board were group III compatibility at 9600 baud, send and receive capability, and a 2400 baud modem included. Why include a modem as well? Because without a combined modem, you would either have to switch your phone line between modem and fax board, add a second telephone line jack for your modem and ensure that some technique exists to preclude both units answering a call, or purchase an electronic call director.

One of the lowest priced fax boards is Frecom's with a suggested retail price (SRP) of $199 (street priced at $175). But that does not include a modem. Intel's new SatisFAXtion board is elegant and feature laden, but at a SRP of $499, I don't consider that low priced.

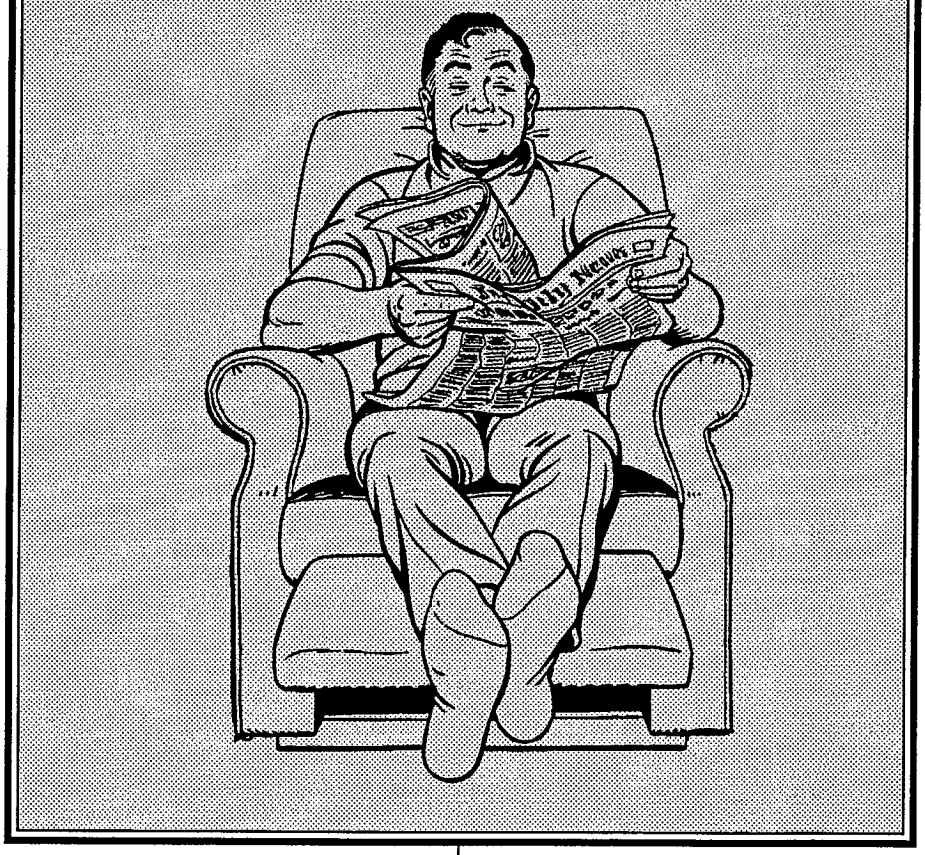
I investigated the Zoltrix ZOFAX 96/24 board which satisfies my technical requirements. This board is a Group III 9600 baud send/receive fax with 2400 baud modem. It comes with BITCOM and BITFAX software. The BITCOM modem software supports eight file transfer protocols: MODEM CheckSum, XMODEM CRC, Relaxed XMODEM, YMODEM, YMODEM batch, Ymodem-G, Compuserve B+, and Kermit. The software appears flawless. I have been using BITCOM for over a month in accessing Compuserve.

The BITFAX software is powerful, yet easy to use. Contrary to what has appeared in print, BITFAX does indeed allow you to receive a fax in the background while you are running another program; RECV_FAX is a removable TSR. You may also send a fax either in foreground from the command line, in a batch file, or as a hot-key pop-up, or as a scheduled task in background. The send side allows you to preview your fax before it is sent; allows you to optionally automatically include a cover page, allows you to merge graphics files (TIF, etc) with text, or combine multiple graphics or text files together on one page, or even merge the contents of your database with a text file for mail merge automated sending to multiple destinations. BITFAX supports printing a received fax directly to a 9-pin or 24-pin dot matrix printer or an HP LaserJet laser printer at either 150 DPI or 300 DPI. I print my faxes to my NEC Postscript printer using its HP LaserJet emulation at 150 DPI.

The advantages of a fax board over a stand-alone fax machine are many. First, you have no paper waste; you can print the fax only when you want to. For another, the quality of a fax transmitted from a board - when the original was a text file - is far superior to that of a scanned and transmitted image. For another, the computing power of the host computer can provide the auxiliary fax utilities costing hundreds of dollars - if not thousands - more for equivalent power from the stand-alone unit. The one drawback is that the fax board is not necessarily targeted for the user receiving a great deal of unscheduled faxes. And if you need to transmit non-original documents, unless you have a scanner attached to your computer, you won't be able to send them with a fax board. The bottom line is to evaluate your requirements for faxing. And if you don't already have a 2400 baud modem, now is the time to plan for this implementation.

If you are looking for a very low-priced fax/modem board for your PC, I think you should consider the Zoltrix ZOFAX 96/24 fax/modem board. All you need is a free 8-bit slot. And I now have a quantity of these boards in stock just waiting for your order. My regular price is $250 + $6 S&H (U.S.), but I have them on sale until January 31st at $225 + S&H. Frecom advertises its $199-priced 9600 baud S/R fax as "the lowest"; that's easily matched by an additional $51 for this board which incorporates a complete 2400 baud modem. And to think I paid $400 or more for my Smartmodem 1200 some years ago!

Letters to MISOSYS



Duplication bug in some 6.3.1H

Fm Walter Klasson: Dear Roy, Since I spoke to you earlier this afternoon, I have ascertained that I have real problems with my copy of 6.3.1.

I have now tried both a verified duplicate copy as well as the original disk in two different Mod 4's and gotten the same results. Basically something is screwy with error reporting and recovery and it also affects direct operations of at least the COPY command. Before each test I have booted a non-configured 6.3.1 disk (no "**SYSGEN**") into a standard 2 floppy drive configuration. I enter the date and push <ENTER> at the time prompt. The easiest test is to do a DIR 7. The result I keep getting is "Error 22H" which is both the wrong code and wrong type of message. I enclose a copy of printout of some operations directly following a boot as per above. The "Error 22H" in the DEVICE command is not constant, which is worrying.

I also get very weird results if I have an erroneous drive or parameter in FORMAT, usually starting with an "Error 22H" and then a sequence of extended error messages followed by a system hang. With a legal command such as COPY Filename/Ext /Bak (which works with a 6.3.0 floppy) I also get a variety of different results and the system usually hangs. I haven't tried other commands, but it seems clear enough that I have at least 1 clobbered system file somewhere or something else is very wrong.

Accordingly, I return my master disk. I'd appreciate a refreshed disk which doesn't do these things. Please let me know if you need any further information.

Thank you for your continued support.

Fm MISOSYS, Inc: Walter, Attached is a copy of a letter I sent on September 20, 1990, along with a replacement 6.3.1 diskette to LS-DOS 6.3.1 Level 1H recipients.

We have discovered that a small corruption of two system files occurred in the creation of the master disk used to generate LS-DOS 6.3.1 Level 1H. The level string appears only when you start up your system. The two corrupted files were SYS6/SYS and SYS4/SYS. A corrupt 6.3.1-1H would have a file date of August 30th, 1990 for the SYS6/SYS file.

The problem has been corrected by generating a new master disk. It is believed that you may have been the recipient of the corrupted version as your order was shipped subsequent to August 30th and prior to the detection of the problem.

The disk enclosed with this letter is a correct version of 6.3.1-1H. Please transfer your serial number from the corrupt version to this disk and destroy the corrupt version.

All copies of the master which you have made need to be updated. You may do this by simply copying the two corrected files from this new diskette to any of your copies. Do this with the two commands,

COPY SYS4/SYS.SYSTEM6:S :D
COPY SYS6/SYS.SYSTEM6:S :D

The strings ":S" and ":D" refer to your source and destination drive numbers.

Note: I'm human. In the letter which was sent, even the patch line had a mistype; both lines referenced SYS6! Incidentally, I believe that I sent a replacement disk to all affected parties. If you still have this kind of problem, get in touch with me.

BASIC's CMD Fluke

Fm Ted Williams: This message goes out to all who may have ideas, having run into the same problem. In trying to write a program that will work across three different Operating systems, I have run into a stone wall. LDOS 5x, and Newdos v2, all have a neat command CMD"doscmd", that will let you perform certain DOS functions from within Basic. However, the closest TRSDOS 1.3 comes is CMD"T","doscmd".

So what's the problem you say? Well, as the manual says, and I quote, "Using this command may overwrite parts of Basic". The kicker is that in every instance I have tried, it ALWAYS overwrites Basic and returns you to TRSDOS ready. Has anyone every had that command work? All I am trying to do is read a text file on the disk instead of having it buried in my Basic program.

Is that TOO much to ask of TRSDOS 1.3? My typical statement under LDOS would be CMD"LIST ABC/TXT" and once done continue Basic computing. Any info would be greatly appreciated. BTW, if you have TRSDOS 1.3, do get Henry Herrdegen's upgrades from Computer News80. Great Stuff!

Fm Gary Phillips: Ted, I'm afraid you are asking too much of TRSDOS 1.3. That operating system was almost (if not completely) brain dead at birth. The best solution is just to use LDOS if you need a model 3 operating system. The LDOS version of Basic is superior anyway. I don't believe CMD"T"never returns you to Basic, because Basic is **always** overlaid by running a DOS command. You might be able to use CMD"T" to execute a program that in turn calls Basic back into memory when it finishes, or use it to execute a batch command file via the DO command that calls Basic as its last command. The awkwardness of this situation is enough that you should seriously consider using a different DOS and Basic combination.

Hard Disk device drivers

Fm Chuck Todd: Roy, do you offer device drivers or source for device drivers, i can build a drive interface, (i am a hardware Tech), But I need info on how to create the drivers.

Also I am interested in your Newest Version 5.3? of the LDOS system. Also can you put me on you mailing list and send me a current catalog of you offerings?

Fm Roy Soltoff: Chuck, We offer hard disk drivers for some commercial hard drives previously available to the TRS-80, as well as software for our own host adaptor. We neither make custom drivers for others nor release source code for our own. I suggest you consider my SCSI host adaptor. I will soon have software available for it to work with embedded controller SCSI drives. Currently, I support Adaptec 4000/4010 and Xebec 1421 with software. I also have drivers for the Radio Shack drive, Percom drive, VR Data drive, and possibly the BiTech drive.

BOOT and DIR

Fm Mark Giannini: Roy, I used the program MAPPER/CMD from MISOSYS's GO:MTC package and came up with a couple **Error** messages on some of the Cylinders. What does this mean? One time after removing a file before that space the **Error** changed to **Empty** then it was back again a few commands later. Also, some of the files are not displayed when mapper is run. Is mapper just a program that displays where the files are saved on disk?Concerning the problem with DIRCHECK :d (H) (also from GO:MTC) I read about the problem in TMQ IV.ii, p.55 and had the same thing happen to me. I was only having the problem with drive partitions above 153 Cyl so I reformatted using 153 Cyl and that "problem"? was solved. Except for 8 Granules of Cyl 77 (old DIR used this space), allocated but not used. Which I let DIRCHECK fix (and it did). I guess when using 154 Cyl the space per Cyl was not enough for the 32k DIR/SYS file. Are there any programs that do maintenance on hard drives? And what will a disk optimizer do for my hard drive? I have a file called DISKOP/CMD.On another subject, my BOOT/SYS & DIR/SYS files were created by MSCSIF6 (the high-level formatter for MISOSYS's hard drive). Can patches to BOOT/SYS be applied to this version of BOOT/SYS? Note: this version has PW of LSIDOS but the DIR/SYS is SYSTEM6. I ask because I use M.A.D. Software's HD auto boot track and would like to install Houde's patches to BOOT/SYS. M.A.D. has XLR8SET/CMD which installs "their" patches to boot and sys0 once they are in memory. BUT, that wont work if I want to create a ERAMDISK to be used as a system disk, will it?Would like to create an ERAMDISK as a system disk. How do I go about this to make a JCL run upon boot and can boot and sys0 be put in ERAMDISK and then loaded to memory (with Houde's patches?) or will I have to patch them in memory with XLR8SET OR can my hard drives boot be patched <sigh>. Hope someone can help.

P.S. I sure wish all M4's and users used the same hardware/software. <sigh> I guess that is where the word "incompatible" comes in, eh?

Fm Roy Soltoff: Mark, The fix to MAPPER (GMAP01/FIX) was in TMQ IV.iv. MAPPER is a tool which displays in what file, by name, is allocated to each granule of a disk. When it is working correctly, the string "**ERROR**" displayed in a granule field indicates that the Granule Allocation Table (GAT) shows that granule as allocated but no directory entry is found with a file using that granule. Another error condition shows up as an arrow; that's where more than one file is allocated to the same granule. The bug in MAPPER was that it incorrectly calculated the length of the directory when the DIR/SYS file was 34 records long; MAPPER didn't examine the last two sectors. Therefore, any file with an entry in those two sectors would not show up in

the map causing the granules allocated to those files to be reflected as an **ERROR**.

You mention that you had experienced the "problem" in DIRCHECK noted in TMQ IV.ii; however, you claim that you solved the problem by reformatting using 153 cylinder partitions - except that eight granules of cylinder 77 showed up as an error. That was exactly the problem! Without the corrective patch applied, DIRCHECK would show an error for the entire cylinder of the hard drive directory. For you to let DIRCHECK fix it was the worst thing you could do. That probably de-allocated the directory cylinder in the granule allocation table (GAT). At some future point in time, some other file will get assigned to that cylinder and overwrite the directory. A word to the wise for using any repair utility - know what it is doing or why it is reflecting an error before attempting automated corrective action. I suggest that you check the actual GAT of that directory to make sure the directory cylinder is allocated - and not to any file other than DIR/SYS. You should be able to use MAPPER, DIRCHECK, and FREE to check up on that. Incidentally, I do not know what DISKOP/CMD is; I am not aware of any disk optimizer (de-fragger) available for the TRS-80. But then, that doesn't mean such a tool does not exist.

When you format a disk, it is the formatter which sets up the information in the directory for the BOOT/SYS and DIR/SYS files. When files are backed up to that disk - other than a mirror image backup - the password entries for those files remain unchanged. But they are easily changed by use of the ATTRIB command. Certainly patches can be applied to BOOT/SYS formatted by MSCSIF6; but a program which uses "LSIDOS" as a password needs to be corrected or you can change the password of BOOT/SYS back to "LSIDOS" (see ATTRIB). You can also patch MSCSIF6 using the fixes printed in TMQ V.i to change MSCSIF6's setting of the password to "SYSTEM6" from "LSIDOS" such that the next time you use it to format a partition, the password will be the "SYSTEM6" password used in LSDOS 6.3.1.

My understanding of XLR8SET is that it replaces the BOOT/SYS and DIR/SYS patches of the Houdé software interface. The use of XLR8SET should not inhibit the use of ERAMDISK. If you want to create a ram drive as the system drive, that can be done from a JCL file. To make a JCL run upon BOOT, use an AUTO command of the form,

AUTO DO STARTUP/JCL

Then the JCL file contains whatever commands you want to have invoked in the job stream. If you are using M.A.D.'s XLR8SET command, you can pass that "auto" JCL in the same command line - or invoke XLR8SET as the first command of the job stream. See XLR8SET's documentation.Finally, to expect all users to use the same hardware and software would be extremely unrealistic - and extremely dull. Variety is the spice of life. Besides, it keeps all of us on our toes.

DOS Directory structure

Fm Carl Berger: Dear Roy, I have two more challenges for you or your readers to solve for me.

Number one pertains to using a JCL to backup from the hard drive to a floppy drive (alphabetically). When a floppy diskette gets full the JCL aborts with a "Can't switch disks while DO in effect" message. The JCL allows a SYSTEM(SYSTEM=DRIVE) command, so why not allow a floppy swap during a disk backup? Could you give me a patch to allow me to "switch disks while DO in effect"?

Number two is a question about how LSDOS formats the directory cylinder. I had a problem with files (text, etc.) getting written to Cylinder 48 (the directory cylinder on my hard drive) starting at Sector 64 through Sector 191.

When Verify was on (I used to always run with Verify on) I got "Error Code 6 (Attempted to read system data record) when trying to save, copy or backup files to drive 0.

Sometimes when looking at the attributes of DIR/SYS I would notice that the file size had changed from 48.00K to 15.00K which could be correct if you multiply the 34 records of DIR/SYS by the LRL of 256, (8K granules), but is wrong since the whole 48k cylinder was supposed to be allocated to the directory.

Some program or programs that I use must look for free disk space in a way not originally intended by the writers of LSDOS or maybe 48K 6 head cylinders were not accounted for in the scheme of things.

The directories on my 96 Cylinder logical drives are on Cylinder 48 and of course the whole 48.00K cylinder shows allocated to the directory on a free space map.

Doing a "DIR DIR(S)" reveals to me a discrepancy in multiplication. 34 records times the LRL of 256 gives an actual DIR/SYS file size of 8.704k and I'm wondering if some programs calculate the free disk space this way and ignore the fact that the whole cylinder is supposed to be reserved for the directory.

The program must be seeing the 34 records of the directory and using the unused space just past it. This "helpful" program must also try to "correct" the "discrepancy" by changing byte X'30' of DIR/SYS Record X'00' from X'FF' to X'C3' and byte X'17' of DIR/SYS Record 3 from X'05' to X'01'.

With PMOD6/CMD I changed the C3 back to FF and the 01 back to 05 and I changed byte X'14' in DIR/SYS record 3 from X'22' to X'BB' and that causes the #Recs Attribute of the DIR/SYS to show 187 instead of 34. I did this a couple of months ago and haven't had any problems with files showing up in the Directory cylinder since.

Should FORMAT/CMD automatically make the #Recs equal the File Size divided by the LRL the of DIR/SYS? Is what I did to correct my problem ok? By the way my drive is 40 meg and config-

ured to have seven 96 cyl 6 head logical drives in the Model 4 mode with formatted space of 4608K each drive. The hard drive drivers are the PowerSoft WD series.

Fm MISOSYS, Inc: Carl, the restriction on switching diskettes during a BACKUP operation while a batch file (Job Control Language) is in effect is based on the fundamental philosophy of batch operation; it is considered to be a *hands off* operation. Thus, it would not be expected to have a program reach a point of requiring operator intervention; that must be considered some unexpected error. That's how BACKUP is programmed. FORMAT and BACKUP are also programmed to abort under other *unexpected* conditions, such as password prompts, where JCL is active. This method of operation is not the same concept as that utilized for being able to switch the logical position of two drives - which assumes that both drives already contain the proper disks. What should abort, and in fact does abort, is a SYSTEM (SYSTEM=d) command invoked from an active JCL when the "d" drive is found to not contain a SYSTEM disk. That type of request would be equally unexpected from a batch job.

Your assumptions about the directory file's DIR/SYS size and ending record number (ERN) are wrong. In fact, those assumptions are where a lot of programmers go wrong and wind up corrupting entries in the directory.

Let's go back to a little history in the development of increasing disk storage associated with an LDOS or LS-DOS type environment. This discussion extends to virtually all other DOS products which are in the TRSDOS family tree. I'll also mix tense because what was in effect then is stil in effect now. In the beginning, there was one size of disk - a 35 track single density 5.25" floppy. That was the size of the Model I diskette and drive. The structure (i.e. the format) that was created for this diskette was ten sectors of 256 bytes each per track. Sectors were numbered 0-9; tracks were numbered 0-34. Each sector on the diskette is identified by a header record which contains information about the sector; this information is used by the floppy disk controller. The file system of the DOS utilized a directory of files; the directory was itself a file. The actual directory file was written in about the middle track of the disk and occupied the entire track. The scheme utilized for the directory allowed it to contain data structures for up to 256 files. Each structure required a minimum of 32 bytes. Two sectors in the directory file were needed for other than file data structures. Thus, a maximum directory could occupy no more than 34 sectors. But the DOS also used no more than a single track to contain the directory; thus, that first disk had a capacity to store no more than 64 files (eight sectors each containing eight file structures and two sectors containing other data).

As track sizes increased, so did the size of the directory. For instance, going to double density disk drives provided 18 sectors per track. That increased the size of the DIR/SYS file to 16 sectors containing file structures (128 file capacity) and two more for other data. As two sided disk drives were developed, the DOS employed the concept of a cylinder, coupling like-numbered tracks on each head to a cylinder of the same number.

Another thing of interest in the DOS stems from the ability of the floppy disk controller (FDC) to encode a particular *address mark* in the header information of each sector on the diskette. The floppy disk drive was first designed by IBM. Their design introduced the concept of two types of sectors: a standard data sector and a deleted data sector. These *flags*, if you will, were encoded in the header record of each sector. Intuitively, the deleted data sector flag indicated that that particular sector was not in use. When Western Digital developed their 1771 FDC, they allowed for four distinct data address marks. Our TRS-80 DOS made use of two of these marks: one was used to indicate a data sector; the other was used to indicate a directory sector. When a read sector command is completed by the FDC, it provides the state of the address mark in the completion status. The DOS makes use of this information, such that if it thinks it is reading the directory but gets the wrong address mark, it will assume that it wasn't reading the right location. Conversely, if the DOS reads what it expects to be a standard data sector but gets the address mark indicating a directory sector, it is also indicative of an error: you get the "Attempt to read system data record" error diagnostic.

Enter two-sided double density disks with 36 sectors per cylinder (two tracks of 18 sectors). The DIR/SYS file is at most 34 sectors, so there would be two unused sectors on the *directory cylinder*. This brings us to the point of being able to discuss what the ending record number (ERN) should be for the DIR/SYS file. There are two ways to access the directory from a common program. One is to use the DOS service calls which perform directory read/write operations. But those are not necessarily standard across all DOS branches on the tree. Another method is to open the DIR/SYS file and read the records. All DOS branches except Model III TRSDOS use the same directory format. NEWDOS 80 was slightly different in the length of its directory allowing for a maximum-sized directory regardless of "cylinder" size. So in reading the directory as a file, you ignore the first two sectors and then read each sector until you get an *end of file encountered* error. This, of course, assumes that the DOS provides READ ACCESS privileges to the DIR/SYS file. Most do; Model I TRSDOS didn't which required programmers to employ contortions to read TRSDOS 2.3's directory. Now if a program reads the directory using standard read service calls, it has to rely on the system to provide the end of file error at the proper point. You certainly would expect BASIC, for example, to properly deal with an ON ERROR GOTO when reading DIR/SYS.

The plot starts to thicken when hard drives came onto the scene. The first hard disk controller to my knowledge which was used in a production environment with a TRS-80 was Lobo Drive's Universal Controller (UVC). The UVC was designed to support two data address marks. Unfortunately, as controller technology progressed, highly integrated chip sets were developed employing command sets which did not support a data address mark convention. In order to use these controllers with a TRSDOS-type OS, it required the

directory data address mark to be generated in software. The method employed in LDOS and subsequently in the Model 4 DOS was to assume that a system data sector was being read if the cylinder being read was the directory cylinder as contained in the drive control table data structure. Fetch the table value and compare it to the requested cylinder; if they are the same and no other error was encountered, then report an error 6 because the read directory access service calls want to see that return code.

Because of the derived address mark, it required that the entire cylinder on a hard drive where the directory was contained had to be reserved. The hard disk drive has no way of knowing the exact file being read; it's job is only to translate drive/cylinder/sector information to what is necessary to communicate with the controller. If a normal file was stored in the cylinder space unused by the directory, any attempt to read one of its sectors would return an error 6. So it is necessary to allocate the entire cylinder of a hard drive where the directory is stored. A hard drive could have up to 256 sectors per cylinder; so it could conceivably *waste* 222 sectors of space.

Now we get to the point where problems in ERN get serious. Early hard drive formatters overlooked entering the correct ERN for the DIR/SYS file in the ERN field when the drive was formatted. This caused no problems when the drive partition being formatted contained only 32 sectors; it was correct. If it contained 64 sectors, programs which read DIR/SYS until EOF read past the last directory record. That's wrong. If a drive contained 256 sectors per cylinder, the ERN was erroneously entered as zero! That's because the calculation for cylinder size only used an 8-bit register which overflows to zero for an exact value of 256. An attempt to read that DIR/SYS file gets one an EOF for the first sector. These problems were soon fixed, but programs still exist which incorrectly calculate the length of the directory.

Now then, let's look at file size. First a question. What do you really want to find out when asking the size of a file? Are you interested in how many pages it will take to print it out? Or are you interested in knowing how much disk space does it occupy? The answer could require a totally different calculation. Most folks want to know the latter. Since disk space is allocated in chunks called *granules* (NEWDOS afficionados refer to these as *lumps*) a one byte file takes up as much disk space as a 4096 byte file if the granule size is 16 sectors. If you request a directory, you are looking at file space calculated as if the file occupied the entire last granule. In fact, this space is allocated to the file, so it is entirely correct to figure it that way. Copy a six-sector file from a double density disk to a single density disk then request a directory on each. Surprisingly, the file has magically become larger on the single density disk! No it hasn't, but because the SDEN disk uses 5-sector granules whereas the DDEN disk uses 6-sector granules, it takes two granules to store that file on the SDEN disk but only one to store it on the DDEN disk. Confusing, isn't it?

Now the *size* of the DIR/SYS file on a drive partition which can contain the maximum sized directory could be different based on the granule size of the partition. Let's assume a 64-sector cylinder. At 32 sectors per granule (SPG), that DIR/SYS would take up 16K since it would need the two granules. At 16 SPG, it would need three granules and take up only 12K; the additional granule of 4K would still be unusable. At 8 SPG, the DIR/SYS file would take up five granules or 10K. In no case should it ever take up more than 16K because that would have to imply an ERN larger than 34; the size of a granule cannot be greater than 32). So your particular case claiming a DIR/SYS size of 48.00K had to be caused by some error.

You now should see where were wrong to change the ERN to X'BB'. You were correct to change the value in the Granule Allocation Table from X'C3' to X'FF'. When you changed the DIR/SYS record value in X'17' from X'05' to X'01', you were telling the system that your DIR/SYS file used two granules. Since your drive assigns six heads per logical drive, that should typically be allocated in granules of 32 sectors; so you would be correct in showing the directory using two granules.

Now how did it get wrong? I can't tell you that. But a formatter should set the number of records in the DIR/SYS entry equal to the actual number of records in the file.

Anitek's speedup

Fm Alan Varga: Hi, Roy. I recently purchased a 6.7 mHz turbo speed-up kit from Anitek. They recommended that I upgrade my memory from 150 ns chips to 120's, which I've done. I had bought their 5.1 mHz speed-up kit a couple of months ago, replacing the Z80, PAL (U3) and 74xx245 (U71) chips.

I've now replaced the Z80 and 74xx245 (U71) chips again with the ones in the Turbo kit, and everything works fine. However, when I replace the PAL chip (U3) I can't get past the 6.3.x logo without locking up. I called Anitek and they had two suggestions.

First, there is a one line program which can be run in Model III BASIC after booting with <SHIFT><BREAK>. It is:

```
10 POKE 16912,40: PRINT
PEEK(16912);: POKE 16912,104:
PRINT PEEK(16912);: GOTO 10
```

This program runs fine at 6.7 mHz, so they are sure that all of my chips are OK.

Their other suggestion was that my DOS has a timing problem during the booting process. Anitek is running DOSPLUS 3.5 on a hard disk, but I want to run 6.3.1 from floppy. They think that LS-DOS isn't waiting long enough for a response back from the drive controller before trying to read whatever comes after the bootstrap routines. Is there a patch I can make to verify this?

Fm Roy Soltoff: Alan, What makes you

think that I am going to work up a patch? In this day and age when a vendor of hardware add-ons to a Model 4 doesn't even support the standard DOS for the machine, I don't think that is very wise. For Anitek to claim they don't know of the problem because they use DOSPLUS on the Model 4 is ridiculous. I am not going to spend my precious time running down a problem with someone's speedup kit. Let them work it out if they want to sell their hardware.

Fm Alan Varga: Roy, I don't expect you to work on anybody else's problems. I just thought that since you were so familiar with LS-DOS that there might have been a single value in the boot-up routine that I could have twiddled on my own if you could only tell me where. I'll check back with Anitek and see if they're willing to work up something. If not, I'll have to bend up their chip and do a little soldering.

Fm MISOSYS, Inc: Alan, I pointed you into the right direction. We have sold *THE SOURCE* for many years - and copies of the three volume set are still available at $40/set. There is no reason on earth why someone who produces an add-on hardware product which needs software support cannot simply obtain a copy of *THE SOURCE* and do their stuff rather than leaving it to the brunt of their customers - who invariably wind up calling on my assistance. *THE SOURCE* makes it easy to find where changes need to occur.

Fm Alan Varga: Roy, I didn't stop to consider that they might have the resources (*THE SOURCE*) to create a patch to 6.3.1 readily available!! I'll give them a call and let you know what they have to say, if anything. Thanks for the suggestion.

PRO-WAM's TODO/DAT and PSORT

Fm Ralf Folkerts: Roy, I have a little 'problem' with ProWAM's TODO/DAT! I can't pack it! I just looked at a few back issues of TMQ but haven't found a patch for this. When I 'delete' a record by 'Done' it will not be deleted during the Pack process of PSORT. I looked at the file and found out that only bit 4 of byte 0 will be reset when the record has been deleted; all other fields remain untouched.

Because the Byte 0 is not part of the sort key (even if it were - since the bit is reset the deleted records would remain at the start of the file) the deleted records remain in the middle of the file during pack - and will not be 'wiped out' of the file!

Do you have a patch for this? Or have I missed something in TMQ/ProWAM manual? Please let me know !

Fm MISOSYS, Inc: No one has ever raised that issue; however, the answer is straightforward - but probably not what you are looking for.

The "packing" facility in PRO-WAM's PSORT utility relies on the appearance of an X'FF' value in the first byte position of a record to recognize it as "deleted". You have correctly ascertained the manner in which TODO designates an active/inactive record, which does not rely on this *character format scheme.*

I would be unable to come up with a patch to alter the behavior of TODO/APP; on the other hand, it would not take too complicated a program to create a new file using the active records in the current file. All one would need to do is to open the TODO/DAT file with a record size of 32; read and copy the first 16 records (which are the header pieces; copy the remaining active records; then update the number of records field in the first record. This could be done even in BASIC; I leave it to the reader as an exercise.

EnhComp FIELD bug

Fm Ralf Folkerts: Roy, I have a problem compiling a program with EnhComp. When I invoke BC either direct or via the S 'RUN' command it bombs the out the system. It either starts to beep and hangs or switches screen-format to 16*64 or something like that. Below is the program I'm trying to compile (or part of it; I just want to see if it works up to this point). It should (when ready) convert a Data-Base from Super to Profile]I[plus. Any tips?

Thank you for your help. By the way, is there a patch for CED to 'force' BLH and ELS when starting ?

```
REM ** CONVERT **
REM CONVERT DATA-BASE FROM
SUPER TO PROFILE ]I[+ FORMAT
REM 10/04/90 BY RALF FOLKERTS
ALLOCATE 2
OPEN "r",1,"weistran/dat",84
FIELD 1,2 AS ZZ$,2 AS DATM$,1
AS Z1$,2 AS DATT$,1 AS Z2$,2 AS
DATJ$,15 AS KUNDE$,6 AS
LIEF$,11 AS EKP$,11 AS VKP$,3
AS HCODE$,3 AS GERAET$,1 AS
FKZ$,12 AS ITYPE$,12 AS HIN-
WEIS$
OPEN "r",2,"weiss000/key",128
FIELD 2,8 AS ADATUM$,15 AS
ANACHNAME$,10 AS AVORNAME$,6 AS
ALIEF$,8 AS AEKP$,8 AS AVKP$,3
AS AHCODE,3 AS AGERAET,12 AS
ATYPE,1 AS AFKZ,54 AS AZZ$
CLOSE
END
```

Fm MISOSYS, Inc: Ralf, You are entirely correct about the PRO-EnhComp problem using FIELD which locked up your system. The problem is in the second FIELD statement; three of the field variables were not string variables. EnhComp was trying to report an error but there was a register pair on the stack when a RET was performed by the error handler. The following patch should correctly handle the TYPE MISMATCH error generation:

```
PATCH BC (D25,9C=C4:F25,9C=C2)
```

This is for the Model 4 version. That changes the JP NZ,... to CALL NZ, which picks up the correct code sequence to exit from FIELD compiling. An equivalent patch, designated BC56/FIX, for the Model I/III version of BC/CMD would be the

following:

```
PATCH BC (D25,89=C4:F25,89=C2)
```

I have no patches to CED to force BLH and ELS.

DO bugs?

Fm Adam Rubin: I'm back! Anyway, I think I've found two previously undiscovered bugs in 6.3's DO command. (At least the second also seems to be in 5.x.) On the second (or subsequent) lines of parameters, if the parameters typed take up the entire length of the input line (or even one character less), the system crashes. The init code at X'2400' intentionally gets reused as a line buffer, but a long parameter line also overwrites a few bytes of code following this.

According to the manual, logical operators are evaluated strictly left-to-right. So, if T is a logical true and F is a logical false, then F&T+T would be evaluated as (F&T)+T, and (F&T) is false so we get F+T, and the entire expression would be considered true. Also, T+F&F would be evaluated as (T+F)&F = (T)&F = false. In that case, why does:

```
. TEST/JCL
//set T
//reset F
//if F&T+T
. F&T+T is TRUE
//else
. F&T+T is FALSE
//end
//if T+F&F
. T+F&F is TRUE
//else
. T+F&F is FALSE
//end
```

get compiled to:

```
. TEST/JCL
. F&T+T is FALSE
. T+F&F is TRUE
```

It looks like the code (a) considers a line to be false as soon as it encounters "false and...", and (b) effectively puts parentheses around all values ANDed together.

Am I misinterpreting the manual somewhere? Or are these indeed bugs? Thanks for any comments or suggestions!

Fm MISOSYS, Inc: Let me respond to your second query first, since it is easier to explain. Correctness is a term which when applied to logic, requires some fundamental premises. It is true that both the Job Control Language logical evaluation in both LDOS and LS-DOS proceeds strictly on a left to right basis with no hierarchy of operators. But it is also true that in its evaluation, *symbol* ***AND*** *string* will be treated as **logically false** if *symbol* is FALSE. If *symbol* is **TRUE**, then the expression analyzer continues to examine the value of the next symbol in *string*. This is how the compiler has been implemented since day one; thus, without any knowledge to the contrary, it is necessary to only improve the documentation. In the evaluation of any compound expression, the expression will be **FALSE** if the left hand symbol of an AND is false.

On the other hand, the compiler evaluates T&F+T as TRUE but F&T+T as FALSE. I don't necessarily agree that is how one would assume the logic to evaluate; both of these expressions should be logically equivalent. But don't forget that the evaluation of an arithmetic expression such as 3+2*4 would result in either 20 or 11 depending on the assumption of operator precedence.

Now for the first question. You stated LS-DOS 6.3, so I'll assume that you don't mean 6.3.1. It is true that the DO compiler library module re-uses the first 80 bytes starting at address 2400H as an internal line buffer. But I don't believe the problem you had with long subsequent command line inputs was caused by overflowing this buffer causing an overwrite of the subsequent code. I would surmise that very few folks make use of the line continuation capability of the JCL command line - that of terminating the line with a semicolon following a closing parenthesis. That could be why no one previously reported the bug.

I'm not sure when the library module was restructured, but if one examines page 176 of *THE SOURCE* Volume 2 to TRSDOS 6.2, one will detect that a pointer to the start of the command line text following the "DO" is saved. Subsequently on page 177 at the label INBUF, that pointer is picked up and decremented by two to seemingly adjust the pointer so that it points to the beginning of the 80-character input buffer. But unfortunately, such is not the case. At the entry to a program, register HL points to the first non-blank character which follows the command name. In this case, the name is "DO". Whoever coded this believed that the decrement was backing up to the "D"; however, it really decrements the pointer to the "O", at most. In fact, entering the DO command with two or more spaces after the DO further compounds the problem. The best solution would be to alter the module so that it saves the contents of register BC - the pointer to the start of the input buffer - and forget about the incorrect decrement which would then be unnecessary. A LD (nn),BC takes four bytes; thus, an in-line patch could not be done. On the other hand, getting rid of the two decrements saves two bytes.

One could correct the problem by a monstrous patch repositioning 108 bytes of code. A re-assembly would be easier for the next release. Alternatively, one could document the bug. In the best case, one could ensure that subsequent command lines are only 78 characters in length with one more for the terminating <ENTER>. That should be bug free if only one space followed the "DO". To force a limit of 78 characters instead of 79 characters, one could change the @@KEYIN limit on page 177. What do my readers want? A monstrous patch? Or just knowledge of the bug azlong with a workaround?

By the way, the crash would occur with a continued command line of 79 characters because the input buffer immediately precedes the drive control table (DCT) in memory. If 80 characters are inserted into memory starting at the second byte of the buffer, the 80th character would load into the first byte of the DCT. That happens to be the jump vector for the system drive. Guess what happens when you next need to access drive :0?

Bargain Floppy Drives?

Fm Henry A. Blumenthal: Dear Mr, Soltoff: A few minutes after I talked to you on the telephone today, I leafed through a flyer that came in the mail from Radio Shack. On page 21, on special price this month, are external 3-1/2 and 5-1/4 drives, each for $99 and in cases! I suppose the cable is extra.

Well, the drives certainly carry a lower price tag than MISOSYS or any of your competitors serving the Model 4 market, I called one of the Computer Centers in this sprawling city of ours to ask if their advertised drives would work with the Model 4. I was told no, because of differences in MS-DOS and LS-DOS interrupts. Yet one of my computer acquaintances has told me that the drives are generic and that it's necessary only to adjust "staking pins" in a drive, whatever they are, and have the right cable configuration (yours?).

Wow ... if the Radio Shack PC external drives won't work with the Model 4, let's get back to what we discussed on the phone. Will the mounting of both a 3-1/2 and a 5-1/4 drive in one of your cases look cosmetically OK?

But if my acquaintance is correct, then it would seem that your dedication to the TRS-80 world should be rewarded by giving you as much financial support as we can. And I want to do that. But my wallet tells me that I have to go with the best value. What new light can you shed on my outboard quandary?

Since the README/TXT that came with my LDOS 5.3 disk (TC0A0628) shows the most recent correction having been made on 02/23/87, I would like to know what bugs have been found and corrected since then, so that I can send you the original diskette for an upgrade. I have been a subscriber to TMQ only since issue III.i, so may have missed some patches.

If corrections have been made since then, let me know your fee for upgrading, so that I can send it with the diskette.

I have three questions about minor annoyances with LDOS:

[1] Why is drive 0 very noisy when files on the distribution disk are accessed, including booting? Are pertinent files too scattered?

[2] Why must I run DATECONV on a freshly formatted LDOS diskette or SubDisk before I can create files or copy files to it? Why wouldn't the FORMAT and SDFORM5 commands automatically apply the current date- and time-stamping structure? Does the upgrade, if there is one, address this problem?

[3] Why must I have ffhard off if I am using Model III Scripsit (patched with your LSI Fix Disk II utility). If the forms filter with ffhard is in place, then Scripsit keeps forwarding the printer long beyond the perforation. Also, even if I set the system to SLOW, by the way, cursor movement in Scripsit is too fast. Did the Fix Disk not go far enough in patching Scripsit to run in an LDOS environment?

Fm MISOSYS, Inc: Dear Henry, The Radio Shack external floppy drives you are referring to are not directly usable with a TRS-80. To begin with, the norm for external PC drives is that they are powered from the host computer. Thus, the drive case contains no power supply; generates much less heat because of that absence; and can then be much smaller. The cable is usually supplied. It generally plugs into a 37-pin D-sub miniature connector. For you to use that drive, you would have to get a power supply, or attempt to manufacture a cabling arrangement to power it from the TRS-80 while probably having to add another power supply to your desktop TRS-80. Incidentally, you couldn't use my cable with Tandy's drive because of the different connectors involved and the power problem as noted.

Getting back to my drives. Mounting a 3.5" and 5.25" - both half-height - in one case will certainly be cosmetically neutral, other than the fact that I usually cannot get 5.25" black faceplates supplied on the 3.5" drive; the faceplates are gray. The faceplate on the 5.25" drive is black. That's the difference. Cosmetic problem for you?

To shed new light, I'm not sure we discussed adding one floppy - say a 3.5" - to your hard drive case. A floppy cable could be provided to connect to it in addition to the hard drive cable. The floppy could be powered from the HD power supply.

Turning to your next letter, I'm including copies of patches applied subsequent to 2/23/87 to LDOS 5.3. You can get your disk refreshed for $10 + $2S&H.

Drive 0 is very noisy because you haven't changed the step rate from the default of 30ms to 6ms. Simply issue SYSTEM (STEP=0,SYSGEN) to change the step rate. Don't forget that LDOS is for Model IIIs and some old machines can't handle a faster step rate on the floppy drives. When a fast drive is stepped slowly, it makes an irritating "grinding" noise.

Both FORMAT from LDOS 5.3 and SDFORM5 should construct a proper directory for extended dating; what's the problem? Give me an example.

Cursor movement in SCRIPSIT is controlled by the program; not the DOS. Changing blinking at the DOS level does nothing to SCRIPSIT; you'll have to either live with that or find a patch to SCRIPSIT. SCRIPSIT also does its own paging control and expects no external paging filter. This has nothing to do with the LDOS environment or the patch applied to SCRIPSIT.

Fm Henry A. Blumenthal: Dear Mr. Soltoff, No, you had not mentioned that your hard drive case could handle a floppy drive — although its face is the right size, come to think of it! This assumes that your installation instructions would guide unsure fingers into mounting a 3-1/2, 720K drive inside the case with little likelihood of my damaging the hard drive, and that the computer would thereupon recognize the 3-1/2 drive as drive 6 (once floppy/dct was applied). I further assume that the same AC switch behind the case would turn on both drives. And you're sure the power supply can handle both! I also

assume that the price would be your advertised $88 plus a one drive host cable. Please advise where my assumptions may have gone astray. I'll weigh your answers against the installation of one or two floppy drives in a separate case.

Thank you for your LDOS patch list; it was kind of you to send it. I was able to update all of the files you had checked off except SYS7/SYS, because it already was updated, thanks to a set of patches applied earlier from some source; I'LL have to look up my patches and see where I got them.

As for formatting under LDOS, it's just like I said: I get the old date structure on any files subsequently added (i.e., 1982 for 1990) unless I do a dateconv immediately after formatting. This is true with format/cmd and sdform5/cmd. Format/cmd is dated 1/5/87; sdform5/cmd is dated 9/14/89. See enclosed.

Fm MISOSYS, Inc: Henry, I don't normally provide instructions for installing a floppy drive into the hard drive cabinet; however, you can examine the instructions for opening up the drive case already provided in the MHDT34 user manual supplied with your drive. Adding another drive - be it a hard drive or a floppy - would incorporate the same method as discussed under *Adding a second hard drive*. Of course, you would not need to consider the cabling method discussed because a floppy drive would not connect to the hard disk controller.

I believe I may have found the answer to the problem you were having with LDOS 5.3 not recognizing a subdisk as if it were an "extended dating" disk. See my response to the next query.

LDOS 5.3 BACKUP bug

Fm Lloyd Evans: Roy, I may have found a bug in LDOS 5.3. I was attempting to put some TED files onto TRSDOS 1.3. The files were stored on a 6.2 data disk with a bit map of 11000010 (double density - single sided) in X'CD' of the directory. I formatted a single sided, double density, 35 track disk with 6.2. Resulting bit map of X'CD' was 10000001.

I them switched to LDOS 5.3 and used:

```
BACKUP /TXT:s :d
```

It did not appear to honor bit 3 which says these disks are to use the old date stamp. The source disk had X'9642' in both password fields just as it should. The 35 track disk received X'0002' for the second password. CONVERT from TRSDOS 1.3 would not work until I zapped these bytes.

When I used 6.3.1 and

```
BACKUP /TXT:s :d
```

I did not have the problem. This is not a big deal to me but I thought that you would like to know.

Fm MISOSYS, Inc: Lloyd, After digging around in LDOS 5.3's BACKUP utility, I came to the conclusion that there is a design flaw. As implemented, the utility does not directly examine the "extended year" bit in the directory (GAT+X'CD', bit 3) but rather uses the appropriate bit in the YFLAG$. This flag value is only updated by the @CKDRV service function which BACKUP doesn't use; thus, BACKUP will use the disk type detected from the last disk logged on that drive. Given time, I may look at a solution; but a patch may be unworkable. After almost three years, I don't think the problem is too severe.

There are a few work-arounds. One temporary solution is to either log a disk or target it with a DIR command: either way will log the disk with @CKDRV. This only needs to be done when you are switching between old disks and new disks (old meaning prior to release x.3).

If one is always going to be using new-style disks, as was relating to Henry's problem, then it may be useful to change the default for the YFLAG$ bits when LDOS 5.3 boots up. It currently defaults to assuming the old style per disk drive except for those logged as new style prior to a SYSTEM (SYSGEN). Henry could install a subdisk, log it, remove it, then SYSGEN to update the default YFLAG$. After that is done, any new re-boot will have the associated drive's flag value default to new so that a subsequent new-style subdisk attachment will be treated as the new style without the need for a DIR or LOG.

Finally, it's good to catch this prior to bringing out 5.3.1 were it is most practical to correct the problem in BACKUP.

Program exit conditions

Fm Hans de Wolf: Dear Roy, I want to report a problem which may be related to a bug in LS-DOS release 6.3.1. I encountered the problem with Mr. Slinkman's GIFMOD4 program, which displays GIF-formatted graphics files on the TRS-80 hires board. This program creates in some cases a temporary file, by means of the DOS command 'CREATE INTRLACE/TMP (lrl=..,...)'. This command is given from within the GIFMOD4/CMD program via the CMNDR SVC (13 hex). This worked fine under LS-DOS 6.3.0, but under LS-DOS 6.3.1 the program aborts with a 'No error' message.

I have been able to fix the problem in the following way: I extracted the 'create' module from SYS7/SYS of LS-DOS 6.3.0 with PROCESS/CMD, and called this MK/CMD. I changed the 'create' text in GIFMOD4/CMD to 'run MK'. This solved the problem. I think that this indicates that the problem is related to the exit condition of the new 'create' module in LS-DOS 6.3.1. The strange behavior would

be explained f the new 'create' not set the A register to 00h on exit, indicating 'No error', but does not adjust the Z-flag. I have reported this problem, and the work around solution to the author of GIFMOD4.

Fm MISOSYS, Inc: Dear Hans, Whether or not GIFMOD4 works with 6.3.0 and doesn't with 6.3.1 is not the point. If GIFMOD4 does indeed treat the accumulator and associated Z-flag value as a basis for a return code from CREATE, it is not following the documentation. @CMNDR is documented in both Tandy's *Technical Reference Manual* and *The Programmer's Guide to LDOS/TRSDOS Version 6* as having the return code in register HL. That is the only place to look for a return code from a program invoked from @CMNDR. The fact that GIFMOD4 worked under a previous version of DOS was just coincidence. There has never been any effort to ensure that the accumulator was zero and the Z-flag set upon exit from a library module.

I have placed the following short message to Frank on my CompuServe forum to that effect. By the way, I commend you on your resourcefulness in working around the problem.

Hans de Wolf informed me that GIFMOD4 aborts after it invokes CREATE under 6.3.1. The problem is that library modules exit with the return code in register HL; the A reg and flag register are meaningless as a return code. Only HL need contain either a zero or an error code. The state of the accumulator in CREATE of 6.3.1 is different from 6.3.0 only due to circumstance; but HL exits CREATE with a zero value. That's the only documented exit code for using @CMNDR.

Incidentally, Frank informs me that GIFMOD4 has been corrected.

A book of questions

Fm Harry Woodgear, 2044 Lake Rd.S.E., Lancaster, Ohio 43130:

1. I formatted my hard drive with 5.3 Ldos. I also have 6.3 Ls-Dos on it. I have been having trouble with my 6.3 Superscripsit. After I revise a file for a while, I get an error message "record length too long". What is the problem?

2. I used 5.1.4 on my hard disk for a long time until I formatted my hard disk with 5.3. Most of my files are under this version of Dos. I used "CONV" to convert my Superscripsit from Trsdos to 5.1.4 Ldos.

My dictionary never did work right under 5.1.4 nor 5.3. I can not add words to my dictionary because if I let it reach the end of the file, the computer will freeze up. I place the word "eenndd" at the end of the document and when I reach it, I press break. All the words I corrected remain corrected but I can not add any words to my dictionary. Do you have a 5.3 version of Super Scripsit (and dictionary) or patches to make it work under 5.3?

3. With my old model 2 I could get in over a hundred pages in Scripsit. When I get over 30 pages in a document with my models 3 or 4, I start having errors. Is this normal or can something be done to create large files in Super Scripsit?

4. Your Ldos quarterly is a magazine which comes out 4 times a year for the cost of $25, Right?

5. What are 'TMQ's? Are these magazines at the cost of $12 each?

6. (Z80 DSMBLR) Will this disassemble 5.1.4, 5.3 Ldos and TrsDos 1.3?

7. I have a lot of programs from TrsDos that I tried to convert to Ldos. I have had little success. I don't know if they have any different machine language codes or anything about the memory. Is there anything published on the differences between TrsDos and Ldos?

8. Is there any program made to convert other dos machine programs to Ldos?

9. Is there any communication program for 5.3 Ldos which can send and receive either Xmodem or ASCII files?

10. As far as I know basic programs are processed and converted into bits which the computer understands. It is slower because of the conversation time. It would seem that after a basic program was written and debugged that it could be processed and instead of sending the binary bits into the program, these bit patterns could be saved to a disk file and then the program was ready to use (instead of basic) pure 0100100110 machine language would be used with no conversion necessary. Is there a program that does this?

11. In school we have been working with sound generators. Is there any program or any method of sending a voice through a mike to the registers, altering the pitch of that voice and sending it back out the cassette port? Is there any such program written? 5.3 LDOS purpose: to produce story tapes and change the voice to match the character.

12. Can voice be recorded on disk? (5.3)

13. Will your 720K drives work with my 102?

14. Will your 720K drives work on my portable 600?

15. Can your 720K drives be made the primary drive on Models 3 / 4 ?

16. Can two 720k drives be attached to the disk expansion port?

17. LDOS tool box is the same as Super Utility plus and will work with 5.3 LDOS. Correct?

18. Does Power Mail Plus have a label printing utility?

19. When I type TED in on my 5.3, I get an error message "attempt to load read only memory." This program works fine on my 6.3. What is the problem?

20. My friends also bought some of the

last Model 4s. One of them is looking for a disk that has blackjack on it. Doss any of your game series disk have this?

21. Power Draw/Power Dot. Will both of these programs work with 5.3 Ldos?

22. (I have a DMP2100) I have 5.3 and 6.3, which of these drawing programs will produce the best graphics on paper?

23. I have one problem with all of my ASCII text editors scripsit, ted, teleterm, deskmate... The problem is that I can only get files of about 10 pages in them. Is there any ASCII text editor which can contain over 100 pages?

24. Is "Mod 4 features for Mod III" a piece of hardware that needs to be soldered or does it plug in. I have never had a computer apart. How hard is it to install?

25. As far as I know my model 4s can only have 128k in them but your literature seems to indicate a larger ram can be installed. Do you or any other vendor offer memory larger than 128k for the models 3 or 4?

26. Do you have any fonts for your graphic software (Power dot/draw)?

Fm MISOSYS, Inc: Dear Harry, This is in response to your undated letter (book?). I'm providing your address in TMQ in case others may be a source to your needs. Here's some answers to your questions.

1. I don't use Superscripsit; your question is best directed to Tandy. But it should have nothing to do with the DOS. I do know that SS has a finite limit on it's document files which relates to how much editing has been done. Your statement, "after I revise a file for a while" may be the clue. Read your SS manual; it may shed some light on that. According to some technical information which was published in TMQ IV.iii, a Superscripsit document may not contain more than 174 blocks.

2. I have no patches to SS or Dictionary; they both should perform just as they did under LDOS 5.1.4.

3. See answer to 1.

4. & 5. We don't have an "LDOS Quarterly"; we publish *The MISOSYS Quarterly* (TMQ) which, as its name implies, comes out four times a year. A subscription is $25 per year. Single issues still in print are $10 each plus shipping. The *LDOS Quarterly* was a publication of Logical Systems, who subsequently published the *LSI Journal.*

6. DSMBLR will disassemble any CMD file as well as memory regions. As such, it can disassemble LDOS and LS-DOS files as well as TRSDOS 1.3 files you have access to.

7. There is nothing which publishes the differences between LDOS and TRSDOS 1.3. The LDOS manual publishes the public entry points into LDOS. The standard entry points for DOS services is identical between LDOS and TRSDOS 1.3. What programs are you talking about? Some talk directly to non-standard addresses.

8. "Dos machine" programs? You don't provide enough in your question for me to know what you are referring to.

9. To begin with, LCOMM is supplied with LDOS; it can send and receive ASCII files. I used to sell XFTS which supported XMODEM, but discontinued it some time ago. There was an XMODEM written in C by David B. Lamkins which I published in TMQ III.i (Spring 1988). You could get a copy of it on DISK NOTES 3.1.

10. Our ENHCOMP BASIC compiler compiles a BASIC program of its dialect to a "pseudo-code" which can speed up runtime depending on the mix of BASIC statements used. You may also find a used copy of BASCOM or RS-BASIC, both of which were compilers sold by Radio Shack.

11. There were a few speech boards sold for the TRS-80. Best you check around on bulletin boards to see if any used ones are available. Dig around in old issues of 80 Micro for some references.

12. To record voice on disk, you need to convert its analog signal to digital. If you have the equipment, it is possible. I am not aware of any such peripheral plugging into the TRS-80. Speech is an analog waveform. Perhaps one of the boards which Alpha Products makes deals with this process.

To store its information in a computer, you need to convert it to a digital format. This is done by an analog to digital converter. There is a theorem in information theory which states that it is only necessary to sample a waveform at twice its highest frequency to accurately record its information. This theorem is applied to virtually all digital communication transmissions today. Somewhere the analog signal of speech gets converted to digital form. If you want to limit the speech frequency to that of telephone use - 300Hz to 3000KHz, I believe - you would need to sample the signal at a 6000KHz rate. You would then need to record its amplitude as a binary value. With an 8-bit value, you can have 256 different steps of amplitude. This conversion process is best done by integrated circuits designed for the job. There are some ICs which perform 12-bit A-D conversion at 25MHz. The playback of a digitally recorded signal is achieved with a chip of the opposite characteristics - one that converts digital to analog.

13. & 14. Our 720K drives are 3.5"; yet, I do not know what the interface is on the 102 or the 600; thus, I can't provide an affirmative.

15. & 16. The 720K 3.5" drives could be the primary drives if you so desire. Just replace the ones internal if you want to BOOT off of them. Or, use the SYSTEM command of LDOS to switch the external drive with the internal drive number. Two drives can be connected external to a Model 4, 4D, or 3; a 4P needs hardware modifications to add an external floppy port.

17. LDOS TOOL BOX is not the same as Super Utility Plus; it is a collection of programs that have features similar to some of the features in SU+. The TOOLBOX will work on hard drives; SU+ will not.

18. PowerMAIL+ has a label printing command - single or multi across.

19. If you got "attempt to load read only

memory" when trying to invoke a program under LDOS 5.3, most likely you were trying to invoke a Model 4 program! It is not wise to have a hard drive interfaced to two DOS versions with both versions having access to alternate DOS executable programs. Isolate the executables to system partitions known only to its respective owner.

20. I believe that MicroLabs sells a blackjack program.

21. & 22. Both PowerDraw and PowerDot are Model III version programs which work under LDOS. PowerDRAW is used to create screen images which can be merged with your BASIC programs and your assembly language programs; it is not functional for printing. PowerDOT creates virtual printer images which are targeted for printing on a dot matrix printer. There are different versions of PowerDOT for Epson, Tandy DMP Series, C.Itoh Prowriter, and Okidata 92/93 printers. Since you have a DMP2100, you would be interested in the PowerDOT Tandy DMP version.

23. Most text editors work with a RAM buffer; i.e. all of the text must fit into RAM. Only LeScript provides direct access to extended memory known to it to allow a RAM buffer larger than what remains in 64K after the program is loaded. Superscripsit operates from disk buffers, but as previously mentioned, it allows only 174 blocks of text (a block is something less than 1024 characters). There was at one time a public domain text editor called VED (Virtual memory editor) which allowed for text to flow to/from disk; you may find a copy if you look hard. Allwrite provides only for RAM buffers of text, but it does allow easy automatic chaining of files to enable the combining of correlated files - say chapters of a book. It would be best to be operating off of a fast hard disk. If you really do a lot of editing where you need huge documents, you better consider another machine. For instance, the text handling capability of Microsoft WORD is virtually limitless, since it uses the available disk space as a virtual buffer. But that's on an MS-DOS machine.

24. The Hardware Interface Kit is just software. It contains a keyboard driver to use all of the Model 4 keys, and a Memdisk to use the extra 64K as a RAMDISK.

25. Our XLR8er board has 256K of additional RAM; that's eight more banks of 32K. Anitek also sells add-on RAM boards.

26. Although PowerSOFT did have an index file folder containing some additional fonts for PowerDOT, it does not appear that they were ever released as a product. From reading the letters in the file, they were supplied by a PowerDOT user. So without any other knowledge, I cannot say that MISOSYS has other fonts for sale.

Whew!

Inquiry on 20 Meg Hard Disk; Tandy 2000

Fm L. Dean Dorsey: Dear Mr. Soltoff, I am interested in your 09/30/90 special offer subject to the answers to my questions:

1. Since your add lists a second software interface, can your equipment be configured to run under both LS-DOS 6.3.1 and LDOS 5.3 as initiated by the appropriate booting DOS diskette - a) by separate hard disk partitions of any size? OR b) by both DOS's accessing the entire hard disk?

2. Will the POWERSOFT Model 4 Toolbelt and POWERSOFT Backrest/Restore programs work with your hard disk, controller and driver combination? Or would I need to replace them with your GO:MTC?

I am interested in replacing my old RS 5 meg drive that was acquired as a demo. It has started to heat up and RS repair was totally incompetent in fixing the problem.

Last January I purchased the POWERSOFT PowerDot II program from MISOSYS. The program is still shrink wrapped since I was winding up my work in preparation for retirement at the time. I have only recently noticed that it was marked for EPSON & compatible printers. Is this the correct version to work with my TANDY 2120 printer? It is a 24 pin printer having IBM 2 character set mode, TANDY character set mode and graphics mode; from old advertisements it appears to be derived from the FUJITSU DL/DLC-2400 printer. I would appreciate your comment.

Congratulations on your MS-DOS utilities advertisement in the September, 1990 issue of PCM magazine! I hope it is a sign of longevity for MISOSYS. Let me know if your DED86 program would be compatible with my MS-DOS computer. It is a TANDY 2000 that is not 100 IBM PC compatible. The principal difference appears to be video addressing and processing. The CPU chip is 80186. Directly addressable memory is 896K via a custom modification of IO.SYS. It has TANDY's machine specific high resolution circuit board with machine specific color chips. TANDY's CM-1 RGB color monitor, in graphics mode, has a high resolution of 640 horizontal pixels by 400 vertical pixels and a low resolution of 320 horizontal pixels by 200 vertical pixels. Total colors are 16 including black. In high resolution mode, the border is black, one color is background and six colors are available for foreground; in low resolution mode, the border is one color, background in one color and three colors are available for foreground. This monitor operates in a non-interlaced mode.

The TANDY 2000 operates on a machine specific MS-DOS 2.11.03 which is the latest available for this computer. The mass storage devices are two internal 5 1/4" 720K floppy diskette drives, an internal 10 MEG hard drive and an external 10 MEG hard drive. The floppy drives are capable of reading, formatting and writing to 360K diskettes. All of the mass storage devices are TANDY supplied but may not be TANDY manufacture.

Fm MISOSYS, Inc: Dear Dean, The Job Control Language (JCL) file we provide to install the drive divides a drive into

four partitions; one for DOS. There is an option, called DUAL, which reserves space for a fifth partition. This additional partition is obtained by reducing the size of the SYSTEM partition; it is used to provide a system partition for the "alternate" DOS. The JCL included with the MSCSI5 software (LDOS) does exactly the same kind of reservation; both JCL files synchronize the actual utilization of the drive to be non-conflicting. Three partitions are accessed in common; however, you have complete manual control over what portion of the drive is seen by which DOS by eliminating the JCL and installing the drive manually.

The Powersoft Toolbelt and Backrest programs will work with the drive; however, Toolbelt has not been updated to understand the date-extended directory structure of DOS 5.3/6.3, so directory reconstruction using Toolbelt may not be 100% effective.

On the PowerDot II program you acquired, the Epson compatible version will probably be the best for you. Most dot matrix printers today support the "IBM" character set to work with MS-DOS. That is also Epson compatible since the Epson printer was the first used by IBM. Setting it to the IBM mode should be sufficient.

Finally, DED86 comes with two versions: one is universal for PC compatibles, of which the T2000 is not; the other version is specifically for the Tandy 2000. Both versions are supplied on the same disk. The reason why we do this is that DED86 was first developed on a Tandy 2000 by a Tandy 2000 user.

Incidentally, I don't know if they are still around, but there was a newsletter geared specifically to the Tandy 2000. It is (was) the Tandy 2000 Orphans, 387 Main St., Westport CT 06880. They were still around in the fall of 1989.

6.3.0 Disk refresh; 4P Floppy Drives

Fm Vincent L. Perinn: Dear Sirs, I own a TRS-80, Model 4, name plate model #26-1069A. The operating system is LS-DOS 6.3J, date of creation: 10 Feb 87 by Logical Systems. I have two requests.

Kindly update the enclosed diskette to the most recent version. I have enclosed a check for $10.00 which I understand, via telephone conversation with your offices, is the fee for this service.

A magazine article published several years ago described how to convert a model 4P to two-sided drives. (I do have a copy of this magazine.) I will now refer to *the MISOSYS Quarterly*, Vol, IV.i, "The Blurb", or page 5 of your catalog which I received last fall, Listed on the reverse of this page, under the category "Floppy Drives and Accessories", is an intriguing item: 5.25" 360K 1/2-height. My first question is: would this be a compatible drive to the one mentioned in the magazine article?

More precisely: Do you provide all the items necessary to convert my model 4 to a dual sided, (a la 4D), machine, and if so, the cost?

In the event you cannot, I require two new disk drives for my machine as the original Texas Specialty drives are slipping and sliding their way into oblivion. What is the availability and cost for replacement drives?

My first inclination is to convert to two-sided, double density drives. I have a RS hard drive utilizing your RSHARD, which works quite well, thank you.

Fm MISOSYS, Inc: Dear Mr. Perinn, To begin with, instead of refreshing your LS-DOS 6.3.0 disk to Level L+, the last 6.3.0 release, it would be prudent to consider purchasing the LS-DOS 6.3.1 replacement disk. I am including information on that release. At only $5 more than the "refresh" fee, I believe it would be more practical for you. Let me know of your decision as I am holding up the processing of your 6.3.0 disk until your reply.

As far as the 1/2 height 5.25" floppy drives I sell, they would fit into a 4P; however, they are not a "drop-in" replacement. You would need to drill new mounting holes because the Fujitsu drive has a thicker front bezel which would cause the drive to be inserted farther into the drive mounting (as the bezel presses against the cabinet front). All mounting holes on drives are in a standard location. But that location is measured relative to the drive casing. Given a thicker bezel, and given that the drive bezel itself dictates its mounting location, drives with differing thickness of bezels would need slightly different mounting holes. You also need to modify the cable connecting the floppy disk controller (FDC) to the drives. Stock Tandy cables have the side select pins pulled. If you choose to replace the cable, I can supply that as well for $15 + S&H.

You may also want to consider the 4P modification which adds a port for two external floppy drives. This was republished in TMQ IV.iii; the article originally was published in *80 Microcomputing*.

To change out desktop Model 4 drives with our Fujitsu 1/2 height drives, you would be left with a gap in the front equivalent to the remaining half of the faceplate. Some folks put two 1/2 height drives into the area reserved for one full size drive; then they can use the remaining drive as an "external" drive by connecting cabling from it to the external drive connector.

What's the best word processor?

Fm Robert Winn: Dear Roy, I am writing this letter to ask if you could tell me in your opinion what is the best and most powerful word processor for the Model 4. I currently have 2 Model 4's and one

Tandy 2000 and I always try to buy the best software for them.

The word processor I currently have for my Model 4 I bought at Radio Shack some time ago. It is SCRIPSIT PRO and I am happy with it however if there is a better word processor around I would like to know. I know you would know about most of the software for the Model 4 and since I am using my Model 4 more and more now if I can improve on its output I want to do so.

Fm MISOSYS, Inc: Dear Robert, I cannot recommend one word processor over another. My personal experience with word processors on the TRS-80 has been limited to Electric Pencil, Scripsit, and Allwrite. But the criteria which makes one WP program more beneficial to a user is highly personal. One user demands speed over wysiwyg. Another wants pure text files. A third wants flexible printer support. A fourth wants lean and mean. No one program will ever satisfy everybody. That's why in the MS-DOS world, there are over a dozen well known and widely sold word processing programs.

At this stage of the game, probably the three premier wp programs for the TRS-80 are Superscripsit (and -PRO), Allwrite, and LeScript. There are differences among them. But the term *"better"* is irrelevant. Is vanilla ice cream better than chocolate? Neither is good if you prefer mocha almond fudge. See what I mean? Also read the article on laser printing on the TRS-80 which appeared in TMQ issue V.i.

XLR8er & M.A.D

Fm Mike Sargent to Fred Oberding: I read your excellent article on M.A.D.'s software and agree with. It is an excellent product. I have installed the HBUILD6 software according to the instructions. I've also installed the xlr8set also, however, I cannot get the ramdisk to engage. I get a "Record number out of range" message. I do not understand its meaning or how to correct the problem. I have a Model 3/4 with MSCSI6 harddisk, Xlr8er, 128k memory, two disk drives. I've changed the cfgrd/jcl to match my current drive system 0,1,2,3 harddisk,4,5 for my disk drives. If I understand the instructions from Misosys the jcl configure would be (drive=7???). When it come to fighting fires and saving lives I am one of the best, but when it comes to computers I "who have a hard time getting out of bed in the morning, without explicit instructions". Please help, Thank you.

Fm Fred Oberding: Mike it isn't you, its RAMDISK/DCT. For some reason I haven't figured out, it will not work with LSDOS 6.3.1 and gives the error msg, "record number out of range." It will work OK with LSDOS 6.3.0 though. I haven't used RAMDISK/DCT since M. Houdé released ERAMDISK/CMD with his other patches (see TMQ III.iii). Unless you got your XLR8er before these Houdé patches were released you should have them on your XLR8er Installation Utilities diskette. If you don't have them send your XLR8er Utility disk back to MISOSYS and Roy will add them for you and I believe its at no charge, but better call Roy just in case.

The new XLR8er Utilities will also include a README/TXT file which will explain the use of ERAMDISK/CMD. You will have to re-do the CFGRD/JCL or scrap it all together as ERAMDISK/CMD is much easier to use. If you still have a problem, feel free to give me a shout again.

Fm Roy Soltoff: Mike, You are probably using LS-DOS 6.3.1 release, but an old version of the RAMDISK software. This was covered on the forum some time ago as well as discussed in The MISOSYS Quarterly. The SYSTEM command of 6.3.1 uses the entire DOS library overlay region (2600H-2FFFH) as is available to the DOS. But some /DCT modules were written to violate the DOS region and load into that region. Your problem is that the loading of the RAMDISK/DCT program is overlaying the SYSTEM command's code and corrupting it; thus, the "strange" error. Get an up-to-date version - or use ERAMDISK.

Another happy M.A.D. customer

Fm the A. N. McAninch's: Dear Roy, Thank you for sending the hard disk driver, TRSHD2/DCT & TRSFORM2/CMD, for my Model 16 so promptly. The MISOSYS QUARTERLY, Volume V.i, arrived today as well.

Please send me a copy of Disk Notes 5.1 right away! Thanks for such a data-packed issue! I particularly appreciated Fred Oberding's review of M.A.D.'s XROM, since I did such a review for Computer News 80. He is right on target. The XROM/XDROM is the most valuable modification one can add to his Model 4/4p/4d, and for only $15 to $35, it is the most cost effective. In addition to excellent documentation and support, it is one of the easiest installations I have made. I installed an XROM on an old, pre-revision "A" non-gate array, and an XDROM on a gate array machine. Both have SmartWatches and high resolution boards installed. I have experienced absolutely no problems with either machine.

Thanks, also, for Frank Durda's article on hard disk drives and Models 16/6000. Lamar Owen's letter together with your response concerning Model II/12 LS-DOS 6.3.1 were most helpful to me. I am eagerly awaiting your patches to update II/12 LS-DOS 6.3.1 to level 1G.

Help with Monty Plays Monopoly

Fm Irwin B. Burton, 4156 S Pagosa Ct, Aurora, CO 80013: Dear Roy, I wanted to include just a quick note with my order for GO:SYS.

Through the years I have become a proud owner of several MISOSYS products. I have also had the opportunity of writing to you with questions and you have always been most helpful. I want to thank you for your support of what I consider to be a fine machine.

I want to congratulate you on another year of The MISOSYS Quarterly. I devour every issue of this publication from cover to cover, then go back and re-read all the old issues while I wait for the [new] one. Keep up the good work!

I also want to congratulate you on your acquisitions of the LS-DOS, DoubleDuty, and other 8-bit products. I have wondered for some time if I should just follow the crowd and jump on the MS-DOS caravan. But my 4P, with 512K of AT memory and now 2 TEAC double sided 40 track drives, does about all I need to do. And with the continued support of people like you, I think I'LL hang on to my 4-P for a little longer. So from a person still into LS-DOS computing. THANKS!

I do have a question that maybe, if you do not have the answer, could be shared with other readers of TMQ. Has anyone been successful in converting the Monty Plays Monopoly program to the LDOS environment? It comes on TRSDOS 1.3, which was OK(?) for a Model III with 48K of Ram. But I nearly drool when I think of the possibility of it running under LDOS from a RamDisk in my 512K of AT memory - all that power being wasted!

Thanks again for your support over the years.

Merging LB data bases

Fm William J. Neuman, 2012 Sandalwood drive, Sanford, NC 27330-7312: Dear Mr. Soltoff; I had called your office several weeks ago to help with the task of merging two files together using "LITTLE BROTHER". You had very graciously pointed out the auto-file program in BASIC in the little booklet you had supplied with LB. I wrote my own little version of it as shown below and last night I "merged" the two programs beautifully. One file had 525 records and the other had 327 and both were formatted the same with an LRL of 88. So it was real easy. I have been using LB for some time now for records for my church and keeping a membership list of about 120 for my Lions club. I also use it for other projects such as Property Owners data, word-processing commands, etc. It always has behaved beautifully and would recommend it to any one. It is way above PROFILE 4 PLUS which I abandoned as soon as I started using LB. The BASIC program below is just adding some of my own ideas to memory jog myself when I went to establish the AUTO/JOB.

I know you still may be busy with other things but I have the beta 1.2.2 version of LB and was wondering have you done any more debugging of that version? It has some bells & whistles that would really make LB a real extra fine winner (more so than it is now!)

Also, what does the CHR$(27) and CHR$(26) do to the AUTO/JOB key strokes? Do they take the place of F3 to save? Looking up the character codes, CHR$(27) gives a SHIFT up Arrow or CTRL, and CHR$(26) gives a SHIFT down Arrow or CTRL Z (which CTRL Z is the way to end a /JOB file).

Fm MISOSYS, Inc: Glad to see that you were easily successful with the task of combining the two data bases. Your program is another illustration for which other LB users may find some hints.

As far as the special character values, all you need to do is just think about what keystrokes you manually strike to cause

```
10 'This program loads a "Little
Brother" data file into an AUTO
JOB
20 'to make two data bases of
the same format come together.
30 'See booklet with LB "Ex-
cerpts from THE MISOSYS QUAR-
TERLY"
40 CLS:PRINT CHR$(16) " MAKE
SURE WHEN RUNNING 'LITTLE
BROTHER' THAT ADD INDEX IS OFF
";CHR$(17)
42 PRINT CHR$(16) " ALSO MAKE
SURE THAT SCREEN 1 HAS BEEN
SELECTED !!!! ";CHR$(17)
43 PRINT
50 PRINT CHR$(16)" Has the
original data base file been
EXPANDED with LB Menu ";CHR$(17)
51 PRINT CHR$(16)" selection 7
to hold new information ?
";CHR$(17);" ";:LINE INPUT YN$
60 IF YN$="Y" OR YN$="y" THEN 65
ELSE 290
65 LINE INPUT "Enter name of
data file to be ENTERED (LRL
must match): ";AS
70 OPEN "R",1,AS,88:'LRL must
match existing file
80 FIELD 1,8 AS NO$,25 AS NA$,@0
AS NES,25 AS OT$,2 AS NSS,I AS
RS$,2 AS KA$,5 AS PR$,1 AS RI$,1
AS YG$
90 INPUT "How many records do
you want to add ", LE
100 OPEN "O",2,"AUTO/JOB:1":
'OPEN sequential LB auto file
110 PRINT #2,"2";CHR$(13);"A";:'
This starts LB menu selection 2
120 FOR L=1 TO LE: 'Do all
records in OLD file
130 GET 1,L
140 PRINT
#2,NO$;CHR$(13);:'Length=6
150 PRINT #2,NA$;CHR$(13);:' "
25
180 PRINT #2,NE$;CHRS(13);:' "
20
170 PRINT #2,OT$;CHR$(13);:' "
25
180 PRINT #2,NS$;CHRS(13);:' " 2
190 PRINT #2,RS$;CHRS(13);:' " 1
200 PRINT #2,KA$;CHR$(13);:' " 2
210 PRINT #2,PR$;CHR$(13);:' " 5
220 PRINT #2,RI$;CHR$(13);:' " 1
230 PRINT #2,YG$;CHR$(13);:' " 1
240 ' length of LRL total=88
250 PRINT #2,CHR$(27);
260 NEXT L
270 PRINT #2,CHR$(26);
280 CLS:PRINT "JOB DONE"
290 CLOSE
63999 END
64000 SAVE "LBAUTO/BAS:1'
```

an ADD of a record. From the MAIN MENU, you enter a "2" for the ADD command. That's where your first "2" comes from. You correctly noted that your LB data base must have a default screen assigned; if not, then the next query from LB would be for you to enter that screen number. By having it as a default (established via menu command 11), you don't need it in the AUTO file.

After "2" invokes the ADD command, the next input is a selection from the "Add mode scan menu"; Your "A" designates the "Add option". From that point on, each field is entered for the record to be added. But you need to terminate each field with <ENTER>; that's the "CHR$(13)" character. When all fields for a record are entered, you press <F3> to save the record. LB internally uses a decimal value of 27 for that function key value across both the TRS-80 and MS-DOS implementations; you can't just look at what it normally represents as an ASCII value. Finally, one way you can terminate an AUTO file is by entering a <CTRL-Z>; that is the decimal 26 value used as your last entry written to the AUTO command file.

PC Tape Backup drives

Fm Shane Dawalt: I'm looking for a Tape backup. I was thinking of buying a fast drive (Read: high priced drive) when I realized that Colorado Memory Systems and Mountain Computer both provide software which will provide unattended backups. If I'm not in attendance during the backup then I guess I really don't care how long it takes. On the other hand, restoring a 100+ Meg drive using a slow tape drive could be about as bad as using floppies (ignoring the fact that floppies require changing periodically).

I'm looking for a drive which has a 120MB capacity (at least). It seems the CMS DJ10+ provides this with the appropriate tape. This drive, however, transfers data at the same rate as a floppy drive — the tape drive is built to interface to a floppy controller. Mountain Computer's FS8000+, on the other hand, will use a dedicated controller and can attain a 6MB/minute transfer rate (using Mountain's MACH2 controller card). Obviously, this is much above the rate of the CMS drive. On the other hand, the FS8000+ is twice the price (depending upon the dealer you choose). Some people have told me that CompuAdd wants $630 or so. I need an external unit so the price goes up about $100.

Fm Roy Soltoff: Shane, If you want unattended backup operation, what difference does it make for the disk to tape transfer? That's kind of a non sequitur, if I'm using the term correctly. I sell the DJ10 drive. To get to the 120Meg point, you have to use the special 60Meg tape versions (DC2120 or equivalent). I have both the DC2000 (40 Meg) and DC2120 tapes now in stock. You then need to use the compression setting in the software.

Why do you need an external unit; no drive slot internal? Or no floppy connector? The CMS drive can use an internal adaptor board if both floppy ports are in use - or your motherboard doesn't support a third device. Call for more information.

Fm Shane Dawalt: Yes, after I left the message I thought about what I had said. This is something I never really considered until I rambled about it in the message to Joe. If the drive will be used in an unattended manner, the transfer rate is not very important (except when I must restore).

Ah HA! Now I see the special circumstances that allow 120MB support. CMS seems to have a "DJ10" and "DJ10+". Might the '+' version provide 120MB support? I don't have explicit information from CMS on any of their products which bothers me.

Yes, CMS has a floppy interface which is useful for persons installing their drive internally. I have no open drive bays. (One is reserved for another HD.)

People in IBMHW have noted that CMS (and Mountain) drives can be daisy chained on existing floppy cables without requiring an internal adaptor board. This is, of course, only usable if the drive is to be mounted internally. I definitely need the 120MB version as I plan to "carry" this drive from my present 286 to my next x86 system which will definitely have more than 80MB HD space.

Questions for CMS drives: Does their software allow multiple incremental backups on the same tape? Does their software provide unattended operation? What, exactly, is the DJ10's transfer rate (just for kicks)? What is the warranty on CMS drives just in the off chance something should happen? Since I do want an external drive, does CMS provide a package which includes the drive, a power supply, the software and an interface card or does it need to be bought piecemeal? I assume CMS also provides the required SHIELDED cable, yes?

Fm Roy Soltoff: Shane, The adaptor card essentially works like an electronic switch. It sits between the FDC and floppy drives connecting the tape drive. The DJ10 is a very low power device. The beauty of the DJ10 is that one size fits all. Optional accessories provide the external box. The KE10 goes for $110. That includes adaptor board, case (with a miniature fan), connecting cable, and new 3.5" faceplate. The drive is powered from the host machine. The kit also includes a power Y-cable and other interconnecting ribbon cables. You just insert the DJ10 into the external case and connect everything together. For an external installation, you need the DJ10 and external kit - along with tapes. CMS warrants the drives for one year.

The transfer rate is 500 KHz - same as for a hi-capacity floppy. Using compression, the effective transfer rate varies with the speed of the machine. Incremental backups are easily done, but you can't just write over an old copy of a file; you can erase the tape - or quick erase it which just nulls the directory. The TAPE program allows you to selectively backup to tape as well as selectively restore from tape.

Everything is menu-driven. Background archiving is done with a scheduler TSR. As I said, if you are going to archive on a delayed schedule, does it matter how fast it does it? If you really need speed, get a DAT drive for about 10 times the cost.

Fm Shane Dawalt: Wait a minute, let me get this straight. The adaptor card plugs into the FDC board? I thought this "adaptor card" was a self-contained card, i.e., no connections to any other internal device except the backplane.

"Incremental backups are easily done, but you can't just write over an old copy of a file". I think you missed my question. I want to store a full backup on one tape and use a second tape for a month's worth of incremental backups.

Can the DJ10 be written to like a floppy drive (within reason)? Say, did you get the DJ10 driver working for the Model 4?

Fm Roy Soltoff: No Shane, the tape adaptor card plugs into an 8-bit slot. The FDC-to-floppy connector on the FDC then is cabled to the tape adaptor. The T/A has a connector to receive the cable unplugged from the FDC which goes to the floppies. A third connector cables to the tape drive or is internally connected to an external connector on the tape adaptor bracket. The T/A also comes with a power Y cable to connect power to the bracket connector for use in powering the external tape drive. I recently moved my tape drive external to free up a slot for the 40Meg drive I added. 80 Megs - even with the EXPANZ! card was not going to be sufficient. I expect to need lots of room for FAX reception.

Let me provide you some more detail on the DJ10 per your questions. I'll assume that when you refer to "incremental backups", you are referring to backing up any file modified since the last backup. With that assumption as a basis, you can easily cause that to happen as the TAPE program provides that as a menu option. You can either perform a full backup or a "targeted" backup. The targeted backup allows you to backup either modified files, or user selection of either entire subdirectories or files specifically tagged. The backup of modified files can easily be performed at the command-line level. When you schedule a backup for a future time, what you are doing is scheduling a .BAT file; the batch file will contain whatever TAPE commands you want to perform. For instance, when I was backing up my entire 80 and 40 meg drives recently, since the TAPE program works on only one partition at a time, I had to backup drives C:, D:, E:, and F: - I have nothing on drive G: currently. I just created a 4-line batch file to perform the tape operation.

Each partition put to tape is considered a "volume"; a tape can have many volumes written to it. If you want to re-use a tape, you can "quick-erase" or "security-erase" it. The former just nulls the directory while the latter writes over those sectors which are in use. Although the tape drive is accessed from the floppy controller, you can't just "write to it like a floppy" as the software interface protocol is quite complex. True, a driver sits between the tape drive and the FDC board, and you read and write sectors of data. The tape can easily be accessed on a random sector basis but seeking throughout the tape would be an exceedingly slow process. It is normally accessed sequentially in a serpentine fashion.

I have not gotten far into my Model 4 tape program yet.

Ignore some TRSCROSS errors

Fm MISOSYS, Inc: *William M Dworzan contacted us to discuss a problem he was having in reading TRS-80 disks on his PC using TRSCROSS. Since I was unable to discern the cause, I asked him to send me a disk he was unable to read. The following letter relates my findings.*

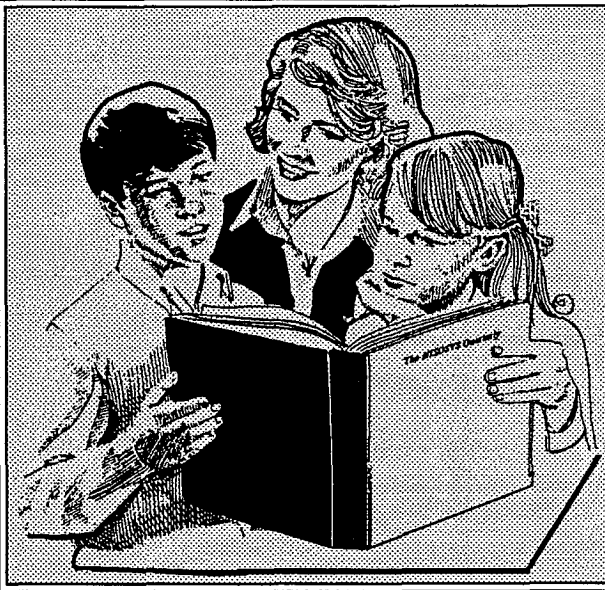
Dear William, This is in response to your problem of converting TRSDOS files to MS-DOS using our TRSCROSS program. I had virtually no problems with your disk. I did get the "*trscross: can't read diskette (abort or ignore?)* error message, but tried IGNORE to see what happens. TRSCROSS then displayed the directory of the TRS-80 diskette. I converted the three BASIC programs, then did a quick scan of them to judge the success.

Apparently, the TRSCROSS process worked perfectly. So if you are having difficulty, sometimes it may pay to continue on and ignore the first error. It may be successful on the second try. Your MS-DOS disk now contains the three files.

1000 TL/2 Booting trouble

Fm Sherry Gates: I'm having trouble booting with my Tandy TL/2 1000. For some reason it now needs the DOS 3.3 disk in A: drive to boot. Was doing fine earlier today. I have tried running setup program again with no luck; any suggestions will be appreciated.

Fm Roy Soltoff: Sherry, Boot from the disk, then go into SETUPTL2. Make sure that it is set to "boot from ROM". Also make sure that the two entries near the bottom - read config.sys and read autoexec.bat - are noted as NO. They will try to read from A: if set to yes (assuming no hard drive is present). The easiest way to accomplish this and normalize everything is to set the SETUP to default. That will restore everything to factory defaults: boot from ROM into DESKMATE.



Inside TMQ

IMAGE PROCESSING ON THE TRS-80 MODEL 4

by J.F.R. "Frank" Slinkman

The accompanying listing, MAC2HR/ASM, is a demonstration of some of the techniques used in graphics image processing. The main techniques demonstrated are scaling using pixel averaging, and error dispersion dithering.

The program reads MacPaint files, which are 576 pixels wide, 720 pixels high, and contain two greyscales. This is expressed as 576 x 720 x 2. It then"scales" them to a size fully displayable on the Model 4's 640 x 240 x 2 screen. In order to do this, the image must be reduced to approximately 1/9th its original size, and 2/9ths of its original number of pixels.

The 720 vertical dimension must be scaled to 240. However, since TRS-80 pixels have an aspect ratio of 1:2 (width:height), and MacIntosh pixels have an aspect ratio of 1:2, the 576 horizontal dimension is multiplied by 2/3rds, rather than 1/3rd, to preserve the correct image aspect ratio (4:5).

Thus the final size of the TRS-80 image will be 384 x 240 x 2. This means each TRS-80 pixel must contain the information from 4-1/2 MacPaint pixels. Since we are combining 4.5 Mac pixels into one TRS-80 pixel, we no longer have a simple monochrome image. MAC2HR converts the values to ten greyscales ranging from 0, which represents black, to 9, which represents white.

Since the Model 4 can display only two greyscales, a technique called "dithering" is used to display the luminence information another way (specifically, by substituting spacial resolution for luminence).

MAC2HR uses the latest and best method of dithering, called "error disbursion dithering." The "errors" to be disbursed are the difference between what is supposed to be displayed and what is actually displayed.

For example, say we need to display a pixel which has a luminence of 100/255ths of maximum (white). Yet the TRS-80 can only display 0/255ths and 255/255ths. Since 100 is closer to 0 than it is to 255, we will display 0/255ths (black), and the error is 100/255ths. Another pixel has a luminence of 240/255ths; so we display 255/255ths (white), and the error is -15/255ths.

This error is disbursed to neighboring, currently undisplayed, pixels in a set manner, determined by a "filter." MAC2HR uses the Slinkman Filter, which was designed for the TRS-80's 1:2 aspect ratio pixels. Filters are based on the fact that light intensity diminishes in proportion to the square of the distance from the source. Most filters modify the values (called "weights) thus determined to make them easy to calculate. The Slinkman Filter is:

```
      X 10 6

   2 3 6 3 2      32nds.
```

Dithering usually proceeds left-to-right, top-to-bottom. The "X" represents the pixel which is the source of the error. The "10" means that 10/32nds of this error is to be disbursed to the pixel immediately to the right of the source-pixel. The "6's"

means 6/32nds of the error goes to each of the pixels second to the right and immediately below. Three 32nds goes to each pixel diagonally down from the source pixel, etc.

If the dithering is to be done left-to-right, then a reflection of the filter would have to be used so that errors would only be disbursed to pixels which have not yet been displayed.

If an error added to a pixel brings it to a luminence greater than 255, it is "clipped" to 255, and the excess luminence discarded. Likewise, no pixel is permitted to have a luminence of less than zero.

The purpose and end result of dithering is best described by example. Suppose we have an area of an image that is supposed to be at 100/255ths of white. Since no single pixel can display the value of 100/255ths, we represent the luminence of that area by making 100 of each 255 pixels white, and 155 of them black. Thus the *area* will have an luminence of 100/255ths. Good filters arrange the white pixels in a seemingly random fashion so that no obvious patterns are created to detract from the image.

Other popular dithering filters are the Burkes Filter:

```
   X 4 2
1 2 4 2 1      16ths,
```

which is a truncation of the Stuki Filter:

```
   X 8 4
2 4 8 4 2
1 2 4 2 1      42nds;
```

and the (true) Floyd-Steinburg Filter:

```
  X 7
3 5 1   16ths
```

Also popular is the "false Floyd-Steinburg Filter," which neither Mssrs. Floyd nor Steinburg had anything to do with. It's easy to program, but produces decidedly inferior results, which why it is not diagrammed here. (Hopefully, if we ignore it long enough, it will go away.)

```
;MAC2HR/ASM      08/12/90
;(0x1B + 0x14 puts DMP printer in cond. mode)
;by J.F.R. Slinkman, 1511 Old Compton Road, Richmond, Va.  23233
;
;Routine to read and expand MACPAINT files, and reduce from original
;576 x 720 format to 384 x 240 format displayable on TRS-80
;Model III/4 high resolution graphics boards.
;
;The Slinkman Dithering Filter:          X 10  6
;                                  2  3  6  3  2          32nds
;
;which is designed specifically for pixels with a 1:2 aspect ratio,
;is in the public domain.  All code in this program, EXCEPT the
;DITHER section, is placed in the public domain.  The DITHER module
;is copyrighted, (c) 1989, J.F.R. Slinkman.
;
*GET    EQUATES                 ;SVC and HD64180 macros
LIN_LEN EQU     576
;
        ORG     3000H
;
MACBUF  DS      100H
MACFCB  DS      20H
OFFSET  EQU     MACFCB+5
;
CODEBUF DS      3*72
LINE1   DS      LIN_LEN
LINE2   DS      LIN_LEN
LINE3   DS      LIN_LEN
        DS      2
CUMERR1 DS      384+2
CUMERR2 DS      384+2
BYTECTR DB      0
ROWCTR  DB      0
;
BRITNES DB      0,28,57,85,113,142,170,198,227,255
;
SYNTAX  DM      'Correct syntax is:  MAC2HR filename[/ext]',LF,LF
        DM      TAB+5,'Where',TAB+10,'filename is the name of the
MACPAINT '
        DM      'file.',LF,TAB+20,'"/MAC" is the default
extension.',LF,CR
MACEXT  DM      'MAC'
MENU    DM      1CH,1FH
        DC      10,LF
        DM      TAB+27,16,'[V]',17,'iew - toggles image on and
off',LF,LF
        DM      TAB+27,16,'[Q]',17,'uit - exits to Dos Ready',CR
;
SYNERR  LD      HL,SYNTAX
        SVC     @DSPLY
        SVC     @ABORT
;
ERROR   OR      40H
        LD      C,A
        LD      A,0FEH
        OUT     (CONTROL),A
        SVC     @ERROR
;
BEGIN   LD      A,(HL)          ;p/u command line char
        CP      CR              ;was a filespec entered?
        JR      Z,SYNERR        ;syntax error if not
        LD      DE,MACFCB       ;   else copy filespec to FCB
        SVC     @FSPEC
        JR      NZ,ERROR
        LD      HL,MACEXT
        SVC     @FEXT           ;add /MAC file ext
        LD      HL,MACBUF
        LD      B,0
        SVC     @OPEN           ;open the /MAC file
        JR      NZ,ERROR
;
```

```
; /MAC file exists, and is open.  Now OK to clear gfx screen
;
CLRGFX  LD      A,74H    ;code for inc Y on write, text on, gfx off
        OUT     (CONTROL),A       ;write to port 0x83
        LD      DE,4FF0H ;D=column # (79), E=# of rows (240)
CGFX010 LD      A,D               ;p/u column #
        OUT     (XREG),A ;   and write to port 0x80
        XOR     A                 ;init board row # to zero
        OUT     (YREG),A ;   by writing 0 to port 0x81
        LD      B,E               ;count 240 rows
CGFX020 OUT     (GFXDAT),A        ;turn 8 pixels off by writing 0 to
port 0x82
        DJNZ    CGFX020           ;do next row
        DEC     D                 ;next lower column #
        JP      P,CGFX010         ;repeat while column # >= 0
        LD      A,0BBH            ;code for inc x on write, text
off, gfx on
        OUT     (CONTROL),A       ;write to port 0x83
        LD      DE,MACFCB
        LD      IX,BYTECTR
        LD      BC,2
        SVC     @POSN             ;position MAC file to 513th byte
        SVC     @GET              ;   and read it
        JR      NZ,ERROR
        OR      A
        JR      Z,BGN010 ;go if 513th byte is a nul
        SVC     @POSN    ;  else point FCB to 513th byte again
        JR      DECODE
;
BGN010  LD      A,80H
        LD      (OFFSET),A        ;position MAC file to 641st byte
;
DECODE  LD      B,240             ;count lines on TRS-80 display
DCOD010 PUSH    BC                ;   and save counter
        LD      DE,MACFCB
        LD      HL,CODEBUF
DCOD020 SVC     @GET              ;get data type byte
        JR      NZ,ERROR
        OR      A       ;is type byte positive or negative?
        JP      M,DCOD040         ;go if negative
        INC     A                 ;here if positive
        LD      B,A               ;count # of unique bytes to read
        ADD     A,(IX)
        LD      (IX),A            ;and accumulate # of bytes in
BYTECTR
DCOD030 SVC     @GET              ;get a data byte
        JP      NZ,ERROR
        CPL
        LD      (HL),A            ;write byte to work space
        INC     HL
        DJNZ    DCOD030
        JR      DCOD060
;
DCOD040 NEG                       ;here if negative
        INC     A
        LD      B,A      ;count # of times next byte is repeated
        ADD     A,(IX)
        LD      (IX),A            ;and accumulate in BYTECTR
        SVC     @GET              ;get data byte to be repeated
        JP      NZ,ERROR
        CPL
DCOD050 LD      (HL),A            ;write it to work space B times
        INC     HL
        DJNZ    DCOD050
;
DCOD060 LD      A,(IX)            ;p/u BYTECTR
        CP      3*72              ;have we read 3 full lines yet?
        JR      NZ,DCOD020        ;go if not
        LD      (IX),0            ;reset BYTECTR if so
        LD      HL,CODEBUF ;3 lines read - convert bits to bytes
        LD      DE,LINE1
        LD      B,3*72            ;count 3 lines worth of bytes
```

To see how scaling, pixel averaging and dithering are accomplished, let's look at the source code, starting at the label BEGIN. Up to the label CLRGFX, this code merely gets the name of the MacPaint file from the command line and opens the file.

You may find the GLRGFX routine useful, as it is much tighter and faster than the routine supplied with the Micro-Labs board documentation. Those using Radio Shack boards may wish to change bit 1 of the values sent to port 83H in this routine, to turn off the video waits prior to the clear to radically increase the speed, and turn them back on afterwards to eliminating hashing of the graphics image on the screen.

After clearing the graphics screen, the program reads the 513th byte of the MacPaint file. I do not know the purpose of this byte. Over 90% of the time, if it is a non-zero value, it is the first byte of the image data, and if it is zero, the image data starts at byte 641. However, I do have one MacPaint file which has 02H in the 513th byte which is not image data (which starts at byte 641). Perhaps a reader can supply 100% reliable information on how to determine the starting point of image data in a MacPaint file.

At any rate, the FCB is *supposed* to be pointed to the first byte of image data (and will be for more than 90% of MacPaint files) when the program reaches the label DECODE.

At DECODE, a counter is set up to count the 240 lines on the TRS-80 display, which also counts the 240 sets of three MacPaint lines which must be processed to create one TRS-80 line. This counter is saved on the stack, and image data is read in.

MacPaint image data consists of sets of from two to 129 bytes in size. If the first byte in a set is positive, it is a zero-based counter of the number (up to 128) of unique data bytes to follow. If it is negative, it is the negative of a zero-based counter of the number (also up to 128) of times the next byte is to be repeated.

In practice, however, the first byte never counts more than 72 bytes, since the MacPaint format is line-oriented, and it

only takes 72 bytes to hold the data for a line. Thus the largest positive number you will see is 47H, and the lowest negative will be 0B9H.

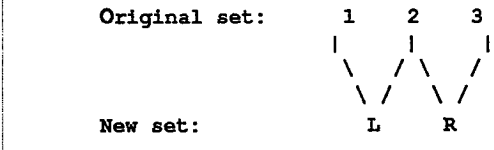
MacPaint image data, like TRS-80 image data, is packed 8 horizontal pixels per byte, with the left-most pixel in bit 7 and the right-most in bit 0. The only difference is that 1=black and 0=white, which indicates to me that the data is intended for output to a printer rather than to a monitor screen.

The code from DCOD010 to DCOD060 expands the compressed data into a 216 (3 * 72) byte field labeled CODEBUF. The three instructions starting at DCOD060 manage a counter named BYTECTR, which is pointed to by IX, which tells the program when 216 bytes of data have been read.

The remaining code in this section unpacks each pixel luminence into one byte per pixel, which data is stored in the three 576-byte fields starting at label LINE1. Think of this data as being arranged in three rows and 576 columns.

The code at VRTSCAL does the vertical scaling by adding the three values in each column, and writing the result to the top byte in that column. Now the data from each column of 3 pixels has been combined into a single byte, with a luminence in the range 0=black to 3=white.

The code at HRZSCAL does the horizontal scaling. Each set of three pixels must be converted to two. This is done as follows:



where the value in original #2 is divided equally between L and R.

HRZSCAL does this by first taking the value of original #1, doubling it, adding the value of original #2, and writing the result to the left pixel of the created pair. Then it takes twice the value of original #3, adds the value of original #2, and

```
DCOD070 PUSH    BC              ;   and save counter
        LD      C,(HL)          ;p/u code byte
        LD      B,8             ;count 8 bits/byte
DCOD080 XOR     A               ;clear for next code
        RL      C               ;shift bit 7 to CY
        RL      A               ;shift CY to bit 0 of A
        LD      (DE),A          ;write byte to LINEs area
        INC     DE
        DJNZ    DCOD080         ;do next bit
        INC     HL              ;-> next code byte
        POP     BC              ;pop code byte counter
        DJNZ    DCOD070         ;do next code byte
;
; At this point, 3 lines of the MACPAINT image are in LINE1...LINE3,
; expanded so that there is one byte per pixel, where 1 = white and
; 0 = black.  These 3 lines must be compressed into one to get the
; image to fit 720-line MACPAINT on the TRS-80's 240-line display.
;
VRTSCAL LD      HL,LINE1 ;-> 1st pixel on LINE1
        LD      BC,LIN_LEN      ;count # of pixels/line
        LD      DE,LIN_LEN ;this is the distance between lines
VSCL010 PUSH    HL
        LD      A,(HL)          ;p/u value from pixel on LINE1
        ADD     HL,DE
        ADD     A,(HL)          ;add value from pixel on LINE2
        ADD     HL,DE
        ADD     A,(HL)          ;add value from pixel on LINE3
        POP     HL              ;-> original LINE1 pixel
        LD      (HL),A          ;store total on LINE1
        INC     HL              ;-> next LINE1 pixel
        DEC     BC
        LD      A,B
        OR      C
        JR      NZ,VSCL010
;
; LINE1 now contains the combined data for all three lines, each
;byte having a value in the range 0 to 3, where 0=black and 3=white.
;  Because TRS-80 pixel have an aspect ratio of 1:2, the horizontal
; scaling must be 2/3 instead of the 1/3 used for vertical scaling.
;
HRZSCAL LD      IY,LINE1
        LD      HL,LINE1
        LD      DE,3
        LD      BC,384          ;576 * 2/3 = 384
HSCL010 LD      A,(IY)    ;p/u value of 1st pixel in set of 3
        ADD     A,A             ;double it
        ADD     A,(IY+1) ;add value of 2nd pixel in set.  2/3rds of
                ; the value in A is from pixel #1; 1/3 from #2
        LD      (HL),A          ;store result
        INC     HL
        LD      A,(IY+2) ;p/u value of 3rd pixel in set of 3
        ADD     A,A             ;double it
        ADD     A,(IY+1) ;add value of 2nd pixel in set.  2/3rds of
                ;  the value in A is from pixel #3; 1/3 from #2
        LD      (HL),A          ;store result
        INC     HL
        ADD     IY,DE           ;IY -> 1st pixel in next set
        DEC     BC
        LD      A,B
        OR      C
        JR      NZ,HSCL010
;
; First 384 bytes of LINE1 now hold all data from the three lines of
; the MACPAINT image, each byte having a value in the range 0 to 9,
; representing ten greyscales, where 0 is the darkest (black) and 9
; is the lightest (white). For dithering purposes, these brighness
; levels must be changed from 9ths to 255ths.  This is done via the
; pre-calculated values in the BRITNES table.
;
CONVERT LD      HL,LINE1
        LD      IY,BRITNES
        LD      BC,384          ;count 384 bytes
```

```
CVRT010 LD      A,(HL)          ;p/u brightness level in 9ths
        LD      (CVRT020),A     ;load into index byte of next
instruction
        LD      A,(IY+00H)      ;p/u brightness level in 255ths
CVRT020 EQU     $-1
        LD      (HL),A          ;replace 9ths with 255ths
        INC     HL              ;-> next 9ths value
        DEC     BC
        LD      A,B
        OR      C
        JR      NZ,CVRT010      ;and convert next value
;
; Now do Slinkman left-to-right dither
;
DITHER  LD      HL,LINE1 ;-> line to be dithered
        LD      BC,384   ;count # of pixel luminences to dither
        LD      IX,CUMERR1      ;-> start of same-line error
buffer
        LD      IY,CUMERR2      ;-> start of next-line error
buffer
;
; The following code is (c)1989, J.F.R. Slinkman.  It may not be
included in
; any program intended for sale or for distribution for profit
without express
; written permission of the copyright holder.
;
DTH010  LD      A,(IX)          ;p/u accumulated errors for this
pixel position
        LD      D,A             ;store error value
        ADD     A,(HL)   ;add current pixel value to error value
        BIT     7,D             ;p/u sign of error (does not
affect C flag)
        JR      C,DTH020 ;go if add caused overflow
        JR      Z,DTH030 ;if no overflow and error positive, value
OK
        XOR     A               ;if no overflow and error nega-
tive, clip to 0
        JR      DTH030
DTH020  JR      NZ,DTH030 ;if overflow and error negative, value OK
        LD      A,-1     ;if overflow and error positive, clip to
255
DTH030  LD      (HL),A   ;write clipped value to current pixel
        OR      A
        JP      P,DTH040 ;if positive, the value is the error
        INC     A        ;if negative, error = value - 255
DTH040  JR      Z,DTH050 ;if result == 0, there is no error to
disburse
        LD      DE,WEIGHTS      ;-> base of 10/32nds table
        LD      E,A             ;-> weight for this error
;
        LD      A,(DE)   ;p/u weight (10/32nds in this case)
        ADD     A,(IX+1) ;add weight to cum errors for pixel to
right
        LD      (IX+1),A ;   and accumulate in buffer
        INC     D               ;-> 6/32nds table
        LD      A,(DE)          ;p/u 6/32nds weight for this error
        ADD     A,(IY)          ;do pixel below
        LD      (IY),A
        LD      A,(DE)          ;p/u 6/32nds weight again
        ADD     A,(IX+2) ;do pixel 2nd to right
        LD      (IX+2),A
        INC     D               ;-> 3/32nds table
        LD      A,(DE)          ;p/u 3/32nds weight for this error
        ADD     A,(IY-1) ;do pixel one down, one left
        LD      (IY-1),A
        LD      A,(DE)          ;p/u 3/32nds again
        ADD     A,(IY+1) ;do pixel one down, one right
        LD      (IY+1),A
        INC     D               ;-> 2/32nds table
        LD      A,(DE)          ;p/u 2/32nds weight for this error
        ADD     A,(IY-2) ;do pixel one down, 2nd to left
```

writes the result to the right pixel of the created pair.

The data from the three pixel set has now been converted to two pixels, which each contain a luminence value in the range 0 (black) to 9 (white).

The processes used in VRTSCAL and HRZSCAL are known as "pixel averaging." None of the original data has been lost, it is just represented in a different form.

The code at CONVERT converts the values of our 384 bytes of 10-greyscale pixels so their luminences are expressed in 255ths, instead of 9ths, of maximum luminence (white). This is done to make dithering easier. For speed, the calculations are made via a precalculated lookup table, labeled BRITNES.

Now we get to do the dithering. The DITHER section manages three fields, namely the line to be dithered (LINE1), a buffer to hold errors dispursed to the current line (CUMERR1), and a buffer to hold errors disbursed to the next line (CUMERR2). These error buffers are used instead of dispursing the errors directly to the image data to minimize the amount of data lost due to clipping, and to minimize the creation of what are called "dithering artifacts" — lines and shapes in the image rendition which were not in the original, which are caused by the sudden "dumping" of accumulated errors.

If all error dispursion (up to seven errors being added to each pixel) is made to the actual data, instead of the accumulated errors being added to the pixel all at once, strange things can happen.

For example, suppose you have a large area which is nearly, but not quite white. Say all its pixels have a luminence of 253/255ths. If you divide up and apply the -2/255ths error to the neighboring pixels, you pull their luminences down very slowly. All pixels will display white until you reach the point where a whole line has virtually all its pixels just below 128/255ths, the threshold for displaying white. When this happens, you will get a line across the screen which will have roughly

every other pixel black. This will appear as a dotted line — a line which was not in the original image — a dithering artifact.

The use of the error accumulation buffers causes the accumulated errors to be displayed more quickly. In the case of a large area with 253/255ths luminence, the first line would have every 506th or 507th pixel black, and each subsequent line would have every 126th or 127th pixel black. This faster use of the error information almost completely eliminates dumping and artifacts.

For speed, all calculation of error weights is done via the precalculated lookup tables at the end of the program, starting at the label WEIGHTS.

First, HL, IX and IY are set up as pointers to LINE1, CUMERR1 and CUMERR2, respectively. BC is used to count the 384 bytes to dither.

At DTH010, the previously accumulated errors to be disbursed to this pixel are loaded into the A register, and copied to the D register. Then they are added to the luminence value for this pixel. The following code to DTH040 tests for overflows and does any necessary clipping. The program logic is tricky, and it may take a little study to understand how and why it works.

(Hint: build a "truth table" which charts the state of the Z and CY flags vs. combinations of positive and negative errors and luminences. Remember, the CY flag is affected by the addition of luminence + error and the state of the Z flag depends on the sign of the error which has been added. What makes this tricky is that errors are SIGNED integers while luminences are UNsigned.)

At DTH040, the state of the Z flag indicates whether there is any error to disburse. If the value to display is either 0 or 255, in which cases there is no display error, the Z flag will be set. Otherwise, there is an error, and it must be disbursed to the seven neighboring pixels.

The first step in error dispursion is to point DE to the first (10/32nds) weight table. When the A register is copied to the

```
        LD      (IY-2),A
        LD      A,(DE)              ;p/u 2/32nds again
        ADD     A,(IY+2) ;do pixel one down, 2nd to right
        LD      (IY+2),A
DTH050  INC     HL                  ;-> next luminence value
        INC     IX      ;-> corresponding byte in same-line error buf
        INC     IY      ;-> corresponding byte in next-line error buf
        DEC     BC
        LD      A,B
        OR      C
        JR      NZ,DTH010           ;do next luminence
;
        LD      (CUMERR2-2),BC      ;null out middle dithering spill-
over space
        LD      HL,CUMERR2-2        ;copy middle spillover space to
left spillover
        LD      DE,CUMERR1-2        ;   space and next-line error
buffer to this-
        LD      BC,386              ;   line error buffer for next
iteration
        LDIR
        LD      HL,CUMERR2-1        ;clear next-line error buffer and
right spill-
        LD      DE,CUMERR2          ;   over space for next iteration
        LD      BC,386
        LDIR
;
; This ends this portion of copyrighted code. At this point,
; the data in the first 384 bytes of LINE1 have been dithered.
; Now they must be packed 8 pixels per byte, and then written
; to the hi-res board.  The display value is in bit 7 of each byte.
;
COMPRES LD      HL,LINE1 ;-> 384 bytes of luminence values
        LD      DE,LINE2 ;-> 48 byte display data storage area
        LD      B,384/8             ;count # of bytes after packing
CPRS010 LD      C,8                 ;count # of pixels/byte
CPRS020 RL      (HL)                ;shift bit 7 of luminence to CY
        RLA                         ;shift CY to bit 0 of A
        INC     HL                  ;-> next data byte
        DEC     C
        JR      NZ,CPRS020          ;do next bit
        LD      (DE),A              ;store packed byte
        INC     DE
        DJNZ    CPRS010             ;pack next 8 pixels
;
DISPLAY LD      HL,ROWCTR           ;-> gfx board row counter
        LD      A,(HL)    ;p/u # of hi-res board row to write to
        INC     (HL)                ;bump counter for next iteration
        OUT     (YREG),A ;write row # to port 0x81
        LD      A,16     ;left margin byte of image on hi-res board
        OUT     (XREG),A ;write column # to port 0x80
        LD      HL,LINE2            ;-> display data
        LD      BC,3082H            ;B counts 48 bytes to send,
                                    ;  C = graphic data  port number.
        OTIR                        ;write line to graphics board
;
        POP     BC                  ;restore MACPAINT line-set counter
        DEC     B
        JP      NZ,DCOD010          ;do next set of lines
;
; Image now complete.  Give disposition menu to user.
;
FINISH  LD      HL,MENU             ;write menu to video RAM
        SVC     @DSPLY
FNSH010 LD      A,0FEH    ;code for no clocks, text on, gfx off
        OUT     (CONTROL),A         ;turn graphics off, text on
FNSH020 SVC     @KEY                ;wait for keystroke
        AND     5FH                 ;convert to U/C
        CP      'Q'                 ;'Q'uit?
        JR      Z,EXIT              ;exit if so
        CP      'V'                 ;'V'iew image?
        JR      NZ,FNSH020          ;invalid input if not
```

```
        LD      A,0FFH   ;code for no clocks, text off, gfx on
        OUT     (CONTROL),A      ;turn text off, graphics on
FNSH030 SVC     @KEY             ;wait for keystroke
        CP      80H
        JR      Z,FNSH010     ;in case user panics and hits break
        AND     5FH
        CP      'Q'
        JR      Z,EXIT
        CP      'V'
        JR      Z,FNSH010
        JR      FNSH030          ;go if invalid input
;
EXIT    LD      A,0FEH
        OUT     (CONTROL),A
        LD      HL,0
        SVC     @EXIT
;
WEIGHTS EQU     $+255.AND.0FF00H
        ORG     WEIGHTS
;
; weight tables for Slinkman Error Dispersion Dither Filter
; The following code is (c)1989, J.F.R. Slinkman.  It may not be
;included in any program intended for sale or for distribution for
; profit without express written permission of the copyright holder.
;
        DB      0,0,0,1,2,3,2,3,2,3,2,3,4,5,4,5          ;10/32nds
        DB      4,5,6,5,6,7,8,7,8,7,8,7,8,9,8,9
        DB      10,11,12,11,12,13,12,13,12,13,12,13,14,15,14,15
        DB      14,15,16,15,16,17,18,17,18,17,18,17,18,19,18,19
        DB      20,21,22,21,22,23,22,23,22,23,22,23,24,25,24,25
        DB      24,25,26,25,26,27,28,27,28,27,28,27,28,29,28,29
        DB      30,31,32,31,32,33,32,33,32,33,32,33,34,35,34,35
        DB      34,35,36,35,36,37,38,37,38,37,38,37,38,39,38,39
        DB      -40,-39,-38,-39,-38,-37,-38,-37,-38,-37,-38,-37,-
36,-35,-36,-35
        DB      -34,-35,-34,-35,-34,-33,-32,-33,-32,-33,-32,-33,-
32,-31,-32,-31
        DB      -30,-29,-28,-29,-28,-27,-28,-27,-28,-27,-28,-27,-
26,-25,-26,-25
        DB      -24,-25,-24,-25,-24,-23,-22,-23,-22,-23,-22,-23,-
22,-21,-22,-21
        DB      -20,-19,-18,-19,-18,-17,-18,-17,-18,-17,-18,-17,-
16,-15,-16,-15
        DB      -14,-15,-14,-15,-14,-13,-12,-13,-12,-13,-12,-13,-
12,-11,-12,-11
        DB      -10,-9,-8,-9,-8,-7,-8,-7,-8,-7,-8,-7,-6,-5,-6,-5
        DB      -4,-5,-4,-5,-4,-3,-2,-3,-2,-3,-2,-3,-2,-1,-2,-1
;
        DB      0,0,1,1,1,1,1,1,1,1,2,2,2,2,3,3          ;6/32nds
        DB      3,3,3,4,4,4,4,4,5,4,5,5,5,5,5,6,6,6
        DB      6,6,6,7,7,7,7,7,7,7,8,8,8,8,9,9
        DB      9,9,9,10,10,10,10,11,10,11,11,11,11,11,12,12
        DB      12,12,12,13,13,13,13,13,13,13,14,14,14,14,15,15
        DB      15,15,15,16,16,16,16,17,16,17,17,17,17,17,18,18
        DB      18,18,18,19,19,19,19,19,19,19,20,20,20,20,21,21
        DB      21,21,21,22,22,22,22,23,22,23,23,23,23,23,24,24
        DB      -24,-24,-24,-23,-23,-23,-23,-23,-22,-23,-22,-22,-
22,-22,-21,-21
        DB      -21,-21,-21,-20,-20,-20,-20,-19,-19,-19,-19,-19,-
19,-19,-18,-18
        DB      -18,-18,-18,-17,-17,-17,-17,-17,-16,-17,-16,-16,-
16,-16,-15,-15
        DB      -15,-15,-15,-14,-14,-14,-14,-13,-13,-13,-13,-13,-
13,-13,-12,-12
        DB      -12,-12,-12,-11,-11,-11,-11,-11,-10,-11,-10,-10,-
10,-10,-9,-9
        DB      -9,-9,-9,-8,-8,-8,-8,-7,-7,-7,-7,-7,-7,-7,-6,-6
        DB      -6,-6,-6,-5,-5,-5,-5,-5,-4,-5,-4,-4,-4,-4,-3,-3
        DB      -3,-3,-3,-2,-2,-2,-2,-1,-1,-1,-1,-1,-1,-1,0,0
;
        DB      0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1          ;3/32nds
        DB      2,2,2,2,2,2,2,2,2,2,2,3,3,3,3,3
```

E register, DE points to the precalculated weight which is roughly equal to 10/32nds of the error. This value is then added to the byte in CUMERR1 which corresponds to the pixel one to the right of the source pixel.

Incrementing D makes DE point to the appropriate weight in the 6/32nds table. This value is picked up, and added to the appropriate bytes in CUMERR1 and CUMERR2. This process is repeated to disburse the 3/32nds and 2/32nds weights in similar fashion.

After the line is dithered, CUMERR2 is copied to CUMERR1, where this accumulated error data will be used to modify the next line, and CUMERR2 is cleared to hold fresh next-line data from the next line to be dithered.

Next, at COMPRES, the high order bits of the 384 dithered luminences are packed into 48 bytes for writing to the hi-res board. This reflects the fact that 128 is the threshold for white. (Now you know why we converted from 9ths to 255ths back at CONVERT.)

At DISPLAY, the graphics board's X and Y registers are set up for the proper row and column numbers, and the 48 bytes are written to the hi-res board via an OTIR instruction. (The board was set up for auto clock X on write at the end of CLRGFX, remember?)

Now the line counter is popped off the stack, and examined to determine whether or not there is more MacPaint image data to process. If so, we make a jump to DCOD010 and do it. If not, the program falls through to FINISH.

Finally, at FINISH, the disposition menu is written to the video RAM, and the graphics board switched to text-only mode to display it. The code is such that the "V" key toggles between the image and the menu, and the "Q" key returns to DOS ready, with the additional provision that if the BREAK key is pressed while the menu is not visible, it will act the same as the "V" key.

MAC2HR is a demonstration program for the purpose of educating the reader about

some image processing techniques. The only practical value I can think of for it is to save the user some printer paper by letting him see what a MacPaint image will look like before he decides whether or not to print it out via a utility such as MACDMP.

The image quality will not be great, for two reasons.

First, in order to scale the image so it will display on the Model 4 screen, we've had to throw away 7/9ths of the resolution, and therefore 7/9ths of the detail.

Second, MacPaint images have already been dithered before being stored on disk, but we are forced to dither them again after scaling. Due to clipping, each dithering operation results in some loss of luminence data; which makes dithering an already dithered image roughly analogous to making a Xerox copy of a Xerox copy.

However, the techniques used in this program are the best currently available, and will produce the best possible rendition of the image under the limitations imposed.

J. F. R. Slinkman
1511 Old Compton Road,
Richmond, Va. 23233

```
        DB      3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4
        DB      5,5,5,5,5,5,5,5,5,5,5,6,6,6,6,6
        DB      6,6,6,6,6,6,7,7,7,7,7,7,7,7,7,7
        DB      8,8,8,8,8,8,8,8,8,8,8,9,9,9,9,9
        DB      9,9,9,9,9,9,10,10,10,10,10,10,10,10,10,10
        DB      11,11,11,11,11,11,11,11,11,11,11,12,12,12,12,12
        DB      -12,-12,-12,-12,-12,-12,-12,-11,-11,-11,-11,-11,-11,-
11,-11,-11,-11
        DB      -11,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-10,-9,-9,-
9,-9,-9
        DB      -9,-9,-9,-9,-9,-9,-8,-8,-8,-8,-8,-8,-8,-8,-8,-8
        DB      -8,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-7,-6,-6,-6,-6,-6
        DB      -6,-6,-6,-6,-6,-6,-5,-5,-5,-5,-5,-5,-5,-5,-5,-5
        DB      -5,-4,-4,-4,-4,-4,-4,-4,-4,-4,-4,-4,-3,-3,-3,-3,-3
        DB      -3,-3,-3,-3,-3,-3,-2,-2,-2,-2,-2,-2,-2,-2,-2,-2
        DB      -2,-1,-1,-1,-1,-1,-1,-1,-1,-1,-1,0,0,0,0,0
;
        DB      0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1
;2/32nds
        DB      1,1,1,1,1,1,1,1,2,2,2,2,2,2,2,2
        DB      2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,3
        DB      3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4
        DB      4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5
        DB      5,5,5,5,5,5,5,5,6,6,6,6,6,6,6,6
        DB      6,6,6,6,6,6,6,6,7,7,7,7,7,7,7,7
        DB      7,7,7,7,7,7,7,7,8,8,8,8,8,8,8,8
        DB      -8,-8,-8,-8,-8,-8,-8,-8,-8,-7,-7,-7,-7,-7,-7,-7,-7
        DB      -7,-7,-7,-7,-7,-7,-7,-7,-7,-6,-6,-6,-6,-6,-6,-6,-6
        DB      -6,-6,-6,-6,-6,-6,-6,-6,-6,-5,-5,-5,-5,-5,-5,-5,-5
        DB      -5,-5,-5,-5,-5,-5,-5,-5,-5,-4,-4,-4,-4,-4,-4,-4,-4
        DB      -4,-4,-4,-4,-4,-4,-4,-4,-4,-3,-3,-3,-3,-3,-3,-3,-3
        DB      -3,-3,-3,-3,-3,-3,-3,-3,-3,-2,-2,-2,-2,-2,-2,-2,-2
        DB      -2,-2,-2,-2,-2,-2,-2,-2,-2,-1,-1,-1,-1,-1,-1,-1,-1
        DB      -1,-1,-1,-1,-1,-1,-1,-1,-1,0,0,0,0,0,0,0,0
;
; This ends this portion of copyrighted code
;
        END     BEGIN
```


TRS-80 Hardware for Sale

We have available for sale, in good working condition, the following 128K computers:

1 **TRS-80 Model 4P**, 26-1080A, Ser. No. 005011
1 **TRS-80 Model 4P**, 26-1080A, Ser. No. 004803
1 **TRS-80 Model 4**, 26-1069, Ser. No. 0075484
1 **TRS-80 Model 4D**, 26-1070, Ser. No. 003634
1 **5 MB Hard Disk**
Cables for all equipment

Also available is a large selection of TRS-80 software with documentation.

Call or write for further information:

Javna Management Company
163 Engle Street
Englewood, NJ 07631
201-567-3032

New XLR8er patches for LS-DOS 6.3.1

by David Goben

DAVID GOBEN
417 Prospect Street, 1st Floor
Willimantic, CT 06226
Ph: 203-423-9312 (home)
203-486-2240 (day)

Dear Roy, Sorry it has been so long since I last spoke with you or wrote. Currently I am contemplating placing one of your 40Meg hard drives and controllers into my PC, and moving the 20Meg ST-225 from it into the Model 4 20Meg kit I got from you (as I did the 225). Hopefully, I'll make it a sister unit to the 3 1/2" 20Meg (QUIET) drive already in the kit, again hoping that this will not involve much more than setting jumpers on it to be selected as drive 2, and adding an interface cable which I'm sure you sell. The Model 4 hard disk kit is a real pleasure to use, and with the *XLR8er* it is an awesome machine. I finally got around to using the **DUAL** parameter in the *HD20INIT/JCL* hard disk initialization file. Wow, hiding LDOS 5.3 away sure helps me maintain filename sanity (adding 6 and 5 to them was driving me crazy).

Anyway, the reason I am writing is because I have managed to re-write the *LS-DOS 6.3.1* versions of Monsieur Michel Houdé's *XLR8er* patches so that they take up only 33 bytes of low RAM, as opposed to the original 121. This represents a savings of 88 precious additional free bytes in low memory. Though you or anyone else may do a double-take at this size difference (almost 1/4 the original), I think that if anyone really sat down and examined exactly what M. Houdé's excellent coding involved, they would agree that there was certainly room for compression.

I never realized until I had actually disassembled M. Houdé's *XLR8er* patches that what they were basically doing was forcing the computer to do double duty on the @BANK routines by the fact of providing two separate routines for similar functions; one set for the the standard 128K configuration, and another for the 256K mounted on the XLR8er board. In later reflection upon this, I wondered why we could not do both with a set of common routines? After all, retrieving the currently active bank, testing, reserving, and releasing banks did not actually access the bank switching code, as selecting a bank did. So why use different code? Also, when selecting a bank, there is much common code between setting up for and following the actual bank selection. For example, following is a sample of the actual bank switching code needed to support both the 128K standard memory extension and the additional 256K provided on the XLR8er:

By maintaining this routine set as the only diverse portion of the bank switching code, and setting up a 2-byte (16 bit) BUR$ table (Bank Use RAM) so that all 10

```
;----------
;SELECT BANK SETUP ROUTINE PRECEDES THIS CODE
;----------
        LD      A,C     ;get the desired bank
        AND     7FH     ;drop bit 7
        LD      B,A     ;save 7-bit code in case banks 0-2
        SUB     3       ;set XLR8er offset, or Carry if 0-2
        JR      C,SELLO ;was banks 0, 1, or 2
;----------
;SELECT BANKS 3-10 FROM 256K ON XLR8ER
;----------
        RLCA            ;mult by 8 for 4K unit offsets
        RLCA
        RLCA
        ADD     A,56    ;add base offset for start of addressing
        DB 0EDH,39H,38H ;OUT0 (38H),A - call in bank from 256K
SELLO:  CALL    C,SEL012 ;select banks 0,1,2 if Carry set
;----------
; COMMON FINISH UP TO GO HERE
;----------
.finish up code to end SELECT BANK function.....
.................
;----------
;SELECT BANKS 0,1,2
;----------
SEL012: XOR     A       ;disable 256K memory on XLR8er
        DB 0EDH,39H,38H ;OUT0 (38H),A
        OR      B       ;get bank # and set Z if bank 0
        JR      Z,SEL012B;was bank 0, so skip next instruction
        INC     A       ;else make 1 = 10 binary, or 2 = 11 binary
        RLCA            ;shift result over to bits 5 and 4 posn
        RLCA
        RLCA
        RLCA
SEL012B LD      B,A     ;save mask
        LD      HL,78H  ;point to OPREG$ in FLAG$ table
        LD      A,(HL)  ;get current OPREG$ mask
        AND     8FH     ;strip out bits 6,5,4
        OR      B       ;merge new mask
        LD      (HL),A  ;save new image to OPREG$
        OUT     (84H),A ;select bank 0, 1, or 2
        RET             ;return for finish-up code
;----------
```

extended banks and the standard upper 32K bank are maintained together, we would no longer have to depend upon separate code to process all the various non-bank switching functions accessible through the @BANK SVC. By testing for bank number limits (above 10) at the start of our new @BANK routine, we can also eliminate the need for a $BAR table (Bank Available RAM), and in fact use that 1-byte freed space as the needed 1-byte extension of the new BUR$. Of course these patches also assume (and require) that the user has both the standard 128K expansion, and the 256K of XLR8er RAM available, otherwise these patches should not be used. Considering the cheap price of RAM chips, there is little reason why you should not add the 128K expansion if you do not have it. This combination of the 128K and the 256K expansions gives us a total of 384K memory. This provides us with 11 addressable banks (0 through 10), and our base 32K of memory, which cannot be addressed through the @BANK SVC due to the fact that the computer's operating system must always be resident.

Of the original "Houdé" patches, the alterations to SYS0/SYS are relatively unchanged from his code, except for an additional initialization routine to set up our new BUR$ mask. I found no need to change tasker patches or math enhancements. By default, the code is set up to 1 memory wait state, 1 I/O wait state, and the interval of refresh cycle set to 40. However, I have mine boot up with an interval of refresh cycle of 80 and have experienced no problems at all (I am using it on a Model 4D with an RS HiRes board. Thus I must use a 6" ribbon for the *XLR8er*, and have shielded the ribbon with kitchen aluminum foil and masking tape — otherwise the *XLR8er* would refuse to run without locking up. I have also tested it on 2 of my other Model 4's and my Model 4P). If you want to change the patch for a refresh interval of 80, all you need to do is change the 02 in the line within the **XLSYS0B1/FIX** file beginning with D10,34= to 03 (12th data item from the left).

The most obvious difference comes in the patches to BOOT/SYS. Mainly, the bulk of my patches center on relative sector 6, over-writing the old @BANK routine code.

```
.*** XLR8B631/JCL ***
//.Procedure to patch an LSDOS 6.3.1 diskette for use with an XLR8er
//.Drive to be patched must be given on JCL command line.
//.If needed parameter (O=N) may be included, same meaning as PATCH
//IF -d
//.DO XLR8B631 (D=d [,O=N])
//QUIT
//END
//IF O
.DO XLR8B631 (D=#d#,o=#o#)
//ELSE
.DO XLR8B631 (D=#d#)
//END
.Diskette in drive #d# is about to be patched
. Press ENTER to continue, BREAK to abort
//ALERT (7,7)
//IF O
patch sys0/sys.system6:#d# using xlsys0b1/fix (o=#o#)
patch boot/sys.system6:#d# using xlbootb1/fix (o=n)
patch sys2/sys.system6:#d# using xlr8s2/fix (o=#o#)
patch sys12/sys.system6:#d# using xlr8s12/fix (o=#o#)
patch backup.utility:#d# using xlr8bu1/fix (o=#o#)
//ELSE
patch sys0/sys.system6:#d# using xlsys0b1/fix
patch boot/sys.system6:#d# using xlbootb1/fix (o=n)
patch sys2/sys.system6:#d# using xlr8s2/fix
patch sys12/sys.system6:#d# using xlr8s12/fix
patch backup.utility:#d# using xlr8bu1/fix
//END
//ALERT 0,0,1,5,0,2
.eop
```

```
.XLSYS0B1/FIX - To use XLR8er special features on LS-DOS 6.3.1 (09-
Sep-90)
.PATCH sys0/sys.system6 USING xlsys0b1/fix
.must also PATCH boot/sys.system6 USING xlbootb1/fix (O=N)
.new @MUL8 svc @x'190a'
d06,f0=57 ED 5C 7B C9;f06,f0=C5 57 AF 06 08
.tasker patch2. end of old @MUL8 @x'190f'
d06,f5=ED 29 38 7C 32 02 02 C9;f06,f5=87 CB 23 30 01 82 10 F8
.fix call to modified @bank instead of SVC @x'1a16' (disk I/O)
d09,74=CD 77 08;f09,74=3E 66 EF
.tasker (beginning) @x'1c11'
d0b,97=67 CD D6 06;f0b,97=F5 21 78 00
.tasker (end) @x'1c3c'
d0b,c2=E1 CD 0F 19;f0b,c2=F1 32 02 02
.special registers setup in initialisation @'1e79' & x'2194' &
x'20c4'
d0d,07=CD 94 21;f0d,07=00 00 00
d0f,60=21 00 02 36 F8 23 7E E6 07 77 C9
f0f,60=00 00 00 00 00 00 00 00 00 00 00
d10,34=3E 40 ED 39 32 ED 38 36 E6 FC F6 03 ED 39 36 AF
f10,34=00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
d10,44=ED 39 38 ED 39 39 3E 88 ED 39 3A C3 CA 20
f10,44=00 00 00 00 00 00 00 00 00 00 00 00 00 00
.eop
```

The 33 bytes of needed extended code deal basically with 12 bytes to finish up the new @BANK routine, and the remaining 21 bytes being the new @VDCTL setup as written by M. Houdé, and simply relocated into lower RAM. Being such a short piece of patching, I did away with the 9 byte header that M. Houde used, and tacked these 33 bytes onto the end of the $FD floppy disk driver.

As of this writing, I have been using these new patches for over a month now, and have absolutely no problem with them. I make heavy use of my ***XMEMDISK/DCT*** ram disk driver, as well as ***BUSYCALC***, my yet to be released fast disk file backup program, and my new ***BCX Business***

```
.XLBOOTB1/FIX - To use XLR8er special features (18-Sep-90)
.PATCH boot/sys.syatem6 USING xlbootb1/fix (O=N)
.must also PATCH sys0/sys.system6 USING xlsys0b1/fix
.fix lomem pointer @x'0206'
d00,06=15 10;f00,06=F4 0F
.bind XLR8er extended code to end of $FD to save room
d0c,3f=14 10;f0c,3f=f3 0f
.new @MUL16 svc @x'06c9'
d04,c9=57 5D 6F ED 6C ED 5C 7B 5A 16 00 19 C9
f04,c9=C5 EB 4F 21 00 00 7D 06 08 29 17 CB 01
.TASKER patch1, end of old @mul16 @x'06d6'
d04,d6=ED 28 38 E3 E5 AF ED 39 38 21 78 00 C9
f04,d6=30 03 19 CE 00 10 F5 4F 7D 6C 61 C1 C9
.@BANK beginning @x'0877'
d06,77=E5 79 E6 7F FE 0B 30 5D 3C
f06,77=E6 7F FE 03 D2 ED 0D 05 FA
d06,80=67 2E 02 3E 80 07 30 01 2D 25 20 F9 26 02 04 10
f06,80=B3 08 0E 86 28 19 0E 46 05 28 14 05 28 09 05 C2
d06,90=2C 21 05 80 39 38 45 79 E6 7F 47 D6 03 38 08 07
f06,90=ED 0D 3A 02 02 BF C9 47 CD 9F 08 C0 78 0E C6 E6
d06,a0=07 07 C6 38 ED 39 38 DC E1 08 21 02 02 46 79 E6
f06,a0=07 07 07 07 B1 32 B0 08 AF 3E 08 E5 21 00 02 CB
d06,b0=7F 77 A9 B0 4F A8 06 00 E1 C8 E3 BF C9 10 05 2F
f06,b0=46 E1 C9 E5 21 05 80 39 E1 DA ED 0D FE 01 17 47
d06,c0=A6 77 18 15 10 05 A6 3E 08 E1 C9 10 07 ED 34 20
f06,c0=3A 01 02 A0 20 C9 78 1F 3F CE 00 07 07 07 07 47
d06,d0=F6 B6 18 ED 10 06 3A 02 02 E1 BF C9 3E 2B B7 E1
f06,d0=3A 78 00 E6 8F B0 32 78 00 D3 84 3A 02 02 47 79
d06,e0=C9 AF ED 39 38 B0 28 05 3C 07 07 07 07 C3 F4 0F
f06,e0=E6 7F 32 02 02 A9 B0 4F CB 79 06 00 C8 E3 BF C9
.@VDCTL @x'0b9a' JP to fix
d09,9a=00 10;f09,9a=42 0D
.ROWCOL 2 ADDR @X'0ddf' multiply
d0b,df=2E 50 ED 6C 00 00 00 00
f0b,df=21 50 00 CD C9 06 65 6F
.change delay in FDC driver @x'0f26'
d0d,26=24;f0d,26=12
.add XLR8 module @x'0ff4', rest of @BANK and @VDCTL
d0d,f4=47 21 78 00 7E E6 8F B0 77 D3 84 C9 ED 38 38 B7
f0d,f4=00 00 00 00 00 00 00 00 00 00 00 00 6D B6 6D B6
d0e,04=CA 42 0D F5 AF ED 39 38 CD 42 0D E3 ED 21 38 E1 C9
f0e,04=6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D B6 6D
.eop
```

```
.XLR8S2/FIX - Patch to SYS2/SYS
.Apply via, PATCH SYS2/SYS.SYSTEM6 XLR8S2
D00,E4=12;F00,E4=09
D00,EE=40;F00,EE=20
D01,04=40;F01,04=20
.eop
```

```
.XLR8S12/FIX - Patch to SYS12/SYS
.Apply via, PATCH SYS12/SYS.SYSTEM6 XLR8S12
D03,7C=12;F03,7C=09
D03,86=40;F03,86=20
D03,9C=40;F03,9C=20
.eop
```

```
.XLR8BU1/FIX - Patch to BACKUP/CMD
.Apply via, PATCH BACKUP.UTILITY XLR8BU
D17,0A=12;F17,0A=09
D17,10=40;F17,10=20
D17,24=40;F17,24=20
.eop
```

Calculator for eXtended memory spreadsheet. All of these programs make strenuous use of extended memory, often doing long-term rapid-fire bank switching, and have yet to have one failure. They also work flawlessly with other bank switching programs, such as ***DoubleDuty***, ***SCRIPSIT Pro***, ***MemDISK***, ***Multiplan***, and ***VisiCalc***. As long as any program which uses bank switching follows the guidelines explained in either *The Programmer's Guide to LDOS/TRSDOS Version 6* or in *The Model 4/4P Technical Reference Manual*, absolutely no problems will be encountered.

In closing, it seems pretty funny to think that I did all this work just so I could maintain your *MSCSI* hard disk driver, *SubDISK* modules, and the *@EXMEM* handler all together in low memory, which I was previously incapable of doing. And it feels great!

The various FIX files associated with this software interface are XLSYS0B1/FIX, XLBOOTB1/FIX, XLR8S2/FIX, XLR8S12/FIX, and XLR8BU1/FIX. These patches are handled by XLR8B631/JCL for LS-DOS 6.3.1. It is used in the form DO XLR8B631 (D=n), where "n" is the drive which contains the system disk to apply the patches to. Also, one can use the O=NO (OH equals NO) parameter if you are patching the new patches over the old ones.

Just for reference, the @BANK code I used to develop my patches is named BANK/ASM.

By the way, I like the new format of ***TMQ***. Without all the downloaded BBS chatter it seems more accessible. Also, I'm very glad you decided to stay with ***TMQ***. Here's hoping that you'll now be able to manage to get enough time to catch a breath of fresh air (but of course, it was in the midst of one of these moments that I came up with this patching idea).

Best wishes to you and your family!

PS: I wrote this using one of my new soft font drivers for SuperSCRIPSIT and SCRIPSIT Pro, making use of 10 point proportional Times Roman normal up-

right, bold, and italics, with a touch of 14-point bold Helvetica for my name in the banner. Nice, huh?

Fm MISOSYS, Inc: Yes, your letter was nicely presented. Anyone in need of a SuperScripsit printer driver, check out David's drivers [see CN80 for details).

For *TMQ* readers, here's some before and after **MEMORY** command display data dumps so that you can see the effects of Goben's XLR8er interface versus the Houdé interface.

```
High = X'FED1'  Low = X'2FFF'
32K Banks avail = 11/11, In use = <------>

 Low Memory Directory     Start = X'08F0'     Length = 2244
Module          Start Address       End Address       Length
$KI                X'08F0'             X'0B87'            664
$DO                X'0B88'             X'0E00'            633
$PR                X'0E01'             X'0E3C'             60
$FD                X'0E3D'             X'0FF3'            439
XLR8               X'0FF4'             X'106C'            121
MSC0               X'106D'             X'11B3'            327


High = X'FED1'  Low = X'2FFF'
32K Banks avail = 11/11, In use = <------>

 Low Memory Directory     Start = X'08F0'     Length = 2156
Module          Start Address       End Address       Length
$KI                X'08F0'             X'0B87'            664
$DO                X'0B88'             X'0E00'            633
$PR                X'0E01'             X'0E3C'             60
$FD                X'0E3D'             X'1014'            472
MSC0               X'1015'             X'115B'            327
```

```
;*****************************************************************
;                NEW @BANK SVC
;*****************************************************************
LSDOS631 EQU    1                ;set to 0 for LS-DOS 6.3.0
LBANK$  EQU     0202H            ;current bank storage
BUR$    EQU     LBANK$-2 ;BUR$ storage area
LOWBNK$ EQU     0FF4H            ;END OF $FD DRIVER
@BNK    EQU     0877H            ;@BANK routine area
;-----------------------------------------------------------------
   IF   LSDOS631
LOWPTR$ EQU     20CAH            ;PATCH SPACE IN DOS 6.3.1
   ELSE
LOWPTR$ EQU     20C4H            ;PATCH SPACE IN DOS 6.3.0
   ENDIF
;-----------------------------------------------------------------
        ORG     @BNK             ;place over top of old @BANK code

@BANK   PUSH    HL               ;save used register
        LD      A,C              ;ensure we have bank #
        AND     7FH              ;strip off bit 7
        CP      10+1             ;bank # too high?
        JR      NC,PERR          ;yes, so error
        INC     A                ;else bump 1 to init for countdown
        LD      H,A              ;set counter register
        LD      L,.LOW.BUR$+2    ;init page offset pointer + 2
        LD      A,1<7            ;init with only bit 7 set
BLOOP   RLCA                     ;shift bit around
        JR      NC,$+3   ;skip next if not at bit 0 position
        DEC     L                ;else drop offset pointer if so
        DEC     H
```

FORTH

A Language for Every Application (but maybe not for every programmer)

Copyright 1990, Second Serial Rights
Bill Brandon
534 Via Corona
Mesquite, TX 75150

Originally published in slightly different form in Science Software, Vol. 4, No. 3 (1988)

Many professional programmers have a curious, even antagonistic, attitude about FORTH. On the other hand, contact with a FORTH enthusiast will quickly remind you that "fan" is short for "fanatic". This is a curious state of affairs, and might cause some speculation about its origin.

As it turns out, both sets of opinions about FORTH are probably justifiable. This article summarizes some of what I have learned about FORTH, and may help you decide whether it should be the primary language for your applications. You can solve any problem with FORTH, often more quickly than with other languages. As you will see, FORTH also presents some interesting challenges.

Most people don't realize how long FORTH has been around. The language was invented in the 1960's, originating as a set of software development tools. Since that time FORTH has found a solid niche in a number of settings, mostly associated with process control and productivity, but with some surprising uses, too.

The chances are good that you have seen or used items produced or controlled with the aid of FORTH. The baggage handling equipment used in some airports and sorting machinery in canneries were two early uses. The camera system that produced the spectacular effects in Star Wars was programmed entirely in FORTH. You

and your children play video games programmed in FORTH. Standalone FORTH programs control heart monitors and many kinds of hospital laboratory equipment. At least one large-scale expert system is written in FORTH, while artificial intelligence researchers use FORTH to emulate LISP and PROLOG.

So why should a busy person like yourself consider FORTH? Well, FORTH is able to support such a wide range of applications because it is a fully structured, high-level language which you, the user, can extend and modify at will. FORTH also gives direct access to primitive operations more readily than assembly language does, is both interpreter and compiler, and provides compact programs that run extremely fast. In addition, FORTH is a highly efficient programming environment thanks to the philosophy that underlies it. No memory hogs, most versions of the language fit into 8,000 to 16,000 bytes of memory.

In form, FORTH is a threaded, interpretive language whose elements are called "words" instead of "instructions" or "statements". The basic set consists of a dictionary of 100 to 150 words. The purpose of a word is to cause the computer to do something: perform a computation, display a message on the screen, and so on. Most of the words look like English words and carry similar meanings, such as EMIT, which causes a specified character to be output. Other words look like symbols; e.g., "@" (pronounced "fetch"), which retrieves the value of a variable or memory location.

Programming consists of using these words to define other words which carry out the application involved. All these new words are related ("threaded") by means of their definitions and are compiled into the FORTH dictionary, which may be further partitioned into specialized vocabularies, or lexicons. The highest level FORTH words in a program are the tasks that are to be performed. To execute a program, you simply invoke the word which is the

```
;count 1 done from bank counter
        JR      NZ,BLOOP ;not done yet, so shift some more
        LD      H,.HIGH.BUR$        ;else set page holding BUR$
        INC     B                   ;init for countdown
;------
;SELECT BANK
;------
SELBNK  DJNZ    RSTBUR              ;not function 0
        LD      HL,8005H ;else, first check stack space
        ADD     HL,SP
        JR      C,PERR              ;error if stack within bank area
        LD      A,C                 ;else get requested bank #
        AND     7FH                 ;strip bit 7
        LD      B,A       ;save 7-bit value in case banks 0-3
        SUB     3                   ;0-2?
        JR      C,SELLOW ;yes
        RLCA    ;else mult x 8 for 4K increments (4 x 8=32K)
        RLCA                        ;*4
        RLCA                        ;*8
        ADD     A,56      ;add base offset (bit 7 ignored, & NC set)
        DB      0EDH,39H,38H        ;OUT0 (38H),A ;select desired
XLR8er bank
SELLOW  CALL    C,SEL012 ;select banks 0,1,2 if Carry set
        LD      HL,LBANK$           ;now point to current bank storage
        LD      B,(HL)              ;save 'old' to B
        LD      A,C                 ;get new
        AND     7FH                 ;strip off bit 7
        LD      (HL),A              ;stuff new current bank #
        XOR     C                   ;keep only bit 7 value
        OR      B                   ;merge in new bank image
        LD      C,A       ;Set to bank select for user re-select
        XOR     B         ;set NZ if xfering to new bank
        LD      B,0       ;set B null for later possible reinvoke
        POP     HL        ;get saved register back
        RET     Z         ;return to caller if no Xfer to HL address
        EX      (SP),HL   ;else set XFER, and get return to HL
        CP      A         ;set Z for OK
        RET     ;and go to bank with HL & BC set for re-invoke
;------
;RESET BUR FLAG
;------
RSTBUR  DJNZ    TSTBUR              ;not function 1
        CPL                         ;else flip mask
        AND     (HL)                ;keep all other bits
SETMSK  LD      (HL),A              ;stuff new BUR$ mask
        JR      BANKX               ;then do OK exit
;------
;TEST BUR FLAG
;------
TSTBUR  DJNZ    SETBUR    ;not function 2
        AND     (HL)      ;else set NZ if bank in use
DNA     LD      A,8       ;init "device not available" in case NZ
        POP     HL        ;recover used register
        RET               ;return with Z for avail, NZ for in use
;------
;SET BUR FLAG
;------
SETBUR  DJNZ    GETBNK              ;not function 3
        DB      0EDH,34H ;TST (HL) ;allocated?
        JR      NZ,DNA              ;yes
        OR      (HL)                ;else set bit
        JR      SETMSK              ;and stuff new mask
;------
;GET CURRENT BANK
;------
GETBNK  DJNZ    PERR                ;not function 4
        LD      A,(LBANK$)          ;else get current bank
BANKX   POP     HL                  ;recover HL
        CP      A                   ;set Z for OK
        RET
;------
;PARAMETER ERROR
```

```
;-----
PERR    LD      A,2BH           ;set SVC parameter error
        OR      A               ;set NZ to altert user to error
        POP     HL              ;recover used register
        RET                     ;return with NZ set
;-----
;PROCESS BANKS 0-2 (called from below)
;-----
SEL012  XOR     A               ;turn off XLR8er bank selection
        DB      0EDH,39H,38H    ;OUT0 (38H),A
        OR      B               ;get bank # and set Z if bank 0
        JR      Z,SEL012X       ;don't bump, and no need to shift
if so
        INC     A
        RLCA                    ;shift over to bits 4 & 5
        RLCA
        RLCA
        RLCA
SEL012X JP      SEL012B         ;now process 2nd part (in append-
age to $FD)
LAST1   EQU     $
;-----------------------------------------
        ORG     LOWBNK$         ;end of $FD driver area
SEL012B LD      B,A             ;save bank select mask to B
        LD      HL,78H          ;point to OPREG$ in FLAG$ area
        LD      A,(HL)          ;get current byte there
        AND     .NOT.01110000B  ;strip out bits 4, 5, and 6
        OR      B               ;apply new mask
        LD      (HL),A          ;save new image
        OUT     (84H),A         ;and send out to OPREG$ to select
needed bank
        RET                     ;then return to cleanup procedure
;-----
;VIDCTRL ROUTINE
;-----
VDCTL   DB      0EDH,38H,38H    ;IN0 A,(38H) ;get current bank
        OR      A
        JP      Z,0D42H         ;go on with VDCTL if bank 0
        PUSH    AF              ;else save bank
        XOR     A               ;set bank 0
        DB      0EDH,39H,38H    ;OUT0 (38H),A
        CALL    0D42H           ;process
        EX      (SP),HL         ;save HL, get old bank to H
        DB      0EDH,21H,38H    ;OUT0 (38H),H
        POP     HL              ;recover HL
        RET
LAST2   EQU     $
;-----------------------------------------
        ORG     LOWPTR$         ;patch to add to SYS0/SYS ini-
tialization
;-----
;PATCH IN INITIALIZATION CODE
PATCH   LD      HL,BUR$         ;point to bank flag
        LD      (HL),0F8H       ;allow up to 10 banks (3 in this
byte)
        INC     HL
        LD      A,(HL)          ;get allocation for 2nd (old BAR$)
        AND     111B            ;keep only bank 0, 1, 2 allocation
                                ;   and allow banks 3-7 in this
byte
        LD      (HL),A          ;restuff with new image
        RET
;-----------------------------------------
        END     @BANK
```

program. There are no line numbers, GOTO statements, or subroutine CALLS, just words.

Here is a fairly typical FORTH word definition:

: TURN-RIGHT CORNERING-SPEED DECELERATE DEG CLOCKWISE TURN SAFE-SPEED ACCELERATE ;

This is taken from a lexicon, or task-specific vocabulary, that contains words to control the movement of a device. It appears almost like a description in English of the steps in the task. The word being defined is TURN-RIGHT. The FORTH word ":" begins the definition, and the word ";" ends it. All the words used to define TURN-RIGHT (DECELERATE, TURN, and so on) have already been defined, using lower level words. TURN-RIGHT uses three values, namely CORNERING-SPEED, DEG (how far to turn), and SAFE-SPEED. Notice the values appear before the words that use them. Also, notice how much information is hidden from the program, the computer, and the user. The values are hidden from TURN-RIGHT until they are needed. The nature of the device is unknown and makes no difference to the program. It could be a bulldozer, a submarine, a Porsche, or a graphic on a video screen. Details of the connection to the device's steering and speed control are hidden as well. As a result, this word can be tested independent of the rest of the program, given the antecedent words and the two values. It can even be tested independent of the device itself. The output could be in any form desired, from a printout to actual device movement. Once debugged, TURN-RIGHT is ready for use, directly or as part of a higher-level word (say, GYMKHANA).

There are four other features of FORTH that give it speed, power, and elegance. Paradoxically, these are the features that account for the antagonistic attitude toward FORTH, cited earlier, and tend to make learning FORTH more difficult.

To begin with, FORTH uses two memory areas, referred to as stacks, to pass parameters and as temporary storage. Because these are "Last-In-First-Out" struc-

tures, you must develop the skill of keeping track of what you have loaded onto them! In the example above, TURN-RIGHT looks for CORNERING-SPEED, DEG, and SAFE-SPEED on the stack, with CORNERING-SPEED on top. Having them in the wrong order will not give the results desired (consider what could happen if a cornering speed of 25 and a degree measure of 120 were reversed). The stacks are important to FORTH. They help the programmer hide information, i.e., to exclude from the procedures all arbitrary details and information that may change, and to minimize redundancy. Stacks also speed up applications by giving FORTH far better efficiency than other languages.

As you may have noticed in TURN-RIGHT, FORTH uses Reverse Polish Notation (RPN). That is, a value to be operated on appears before the operator. In other words, to add two plus two in FORTH, the expression is written 2 2 + (like the old HP calculators). This is done for the sake of speed in arithmetic computations and for ease of execution. Getting used to RPN is mostly a matter of time and discipline.

Third, and more serious for the new user of FORTH, there is almost no built-in error handling. Depending on the nature of the mistake, you may get off with a simple "?" or "Stack empty" message. Other errors have more disastrous results. FORTH may crash, or it may go off to La-La Land and take your data with it. It seems to be very easy to accidentally overwrite or erase vital system information. You have to build in your own error traps and messages — FORTH will not protect you. The absence of run-time error checking keeps FORTH simple and flexible and gives you absolute control of all parts of your system at all times. Indeed, FORTH makes you totally responsible for the efficiency and proper functioning of the system. The experienced programmer avoids problems by using FORTH's structure: each word is tested as it is built, so when the application is finished, debugging is complete. Until you achieve this level of skill, think of your confusion and system crashes as major opportunities to learn.

Finally, FORTH relies on integer arithmetic in most implementations (but not all — Misosys' HartFORTH and HartFORTH 86 are notable exceptions). Integer arithmetic has the advantage of being fast and flexible. If your application involves extensive, repetitive number-crunching, the small amount of time required to do the programming is rewarded by the execution speed gained.

I belive FORTH offers advantages which are not equalled by any other language or operating system. First, the philosophy behind FORTH complements the creative problem-solving process. FORTH encourages prototyping, rather than lengthy planning which may turn out to be counter-productive. After all, things change. In a similar vein, FORTH's emphasis on efficiency means that you get to spend your time on your work, not on trying to cope with the computer that is managing your application. FORTH is interactive, so you can still control things directly and don't have to be hindered by a tedious edit —> compile —> test sequence. Finally, FORTH's speed and flexibility help you to be as productive in the actual application as you were during the design, by enabling high-speed capture of data, precise and interactive control of equipment and processes, and rapid data reduction.

Not everyone wants to take the trouble to learn FORTH, and I don't expect to be able to influence any seriously negative attitudes about the language. However, if you would like additional information, I would recommend the following sources as a start:

Kelly, M., and Spies, N., FORTH, PC Tech Journal, September and October 1986. (A very thorough look at the architecture and character of the language.)

Brodie, L., Thinking FORTH, Englewood Cliffs, NJ: Prentice-Hall, 1984. (An excellent book on problem solving the FORTH way. The author assumes you have some knowledge of and facility in FORTH, however.)

Toppen, D., FORTH: An applications approach, New York: McGraw-Hill, 1985. (This is an incredibly practical book, written specifically to facilitate the introduction of FORTH into the environments of scientific and engineering applications programming. It may be the most accessible book on this list, even though the system used in the examples is the DEC PDP-11.)

Winfield, A., The Complete FORTH, New York: John Wiley & Sons, 1983. (Also a very accessible text written with a distinctly British accent.)

About the author...

Bill Brandon is the Manager, Employee Development, for Atmos Energy Corp., and one of two SYSOPs for the Computer Training Forum on CompuServe. He endeavors to learn FORTH while playing guitar and raising his kids in Mesquite, Texas.

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

A MAKE utility for MC

by Masa-aki Kitajima

Masa-aki Kitajima
Yamashou-Haitsu C-1-1-3-3-1
Namiki, Tsukuba 305
JAPAN

For a long time, BASIC was my major tool for program development, but now MC has come to be my favorite because of its well considered portability across different models of computers. The comprehensive standard functions in the MC library greatly encourages me to develop various programs because they could be recompiled with minimal changes in different computers in my company's office.

I appreciate the several supporting utilities of MC and MRAS, such as MC/JCL, MLIB/CMD and XREF/CMD, which made MC more useful as the comprehensive tool for program development. And one more such utility, MAKE, would also be useful, but unfortunately it is not contained in the current MC package.

Recently, I tried to write a MAKE for MC system. Included is the result of my struggle around MC and MC/JCL. I hope it is helpful for MC users.

MAKE is, of course you know, a famous utility which checks the file date/time stamp and recompiles the program if there is found any revision of source files. Fortunately LS-DOS supports file time-stamps, and then it is now possible to know whether the source file is younger than the object file or not. That is, the DOS environment is welcoming a MAKE, I suppose. Following is the explanation about my version of MAKE.

Invoking MAKE:

Under LSDOS Ready prompt, type,

```
MAKE filespec -e [ENTER]
```

where filespec is the name of MAKE data file (/MAK file) and it must have neither extension nor drivespec. {-e} is an optional parameter. If it is specified, MAKE writes a necessary JCL file but does not execute it. MAKE automatically executes it if the parameter is not specified.

MAKE data file:

MAKE reads a MAKE data file (which has an extension of "/MAK") to get to necessary information to follow. A MAKE data file has the format as follows;

```
FILE-A FILE-B — FILE-Z
COMMAND
FILE-a FILE-b — FILE-z
COMMAND
```

MAKE data file is consisted on two-line block configuration. The first line of a block is a list of files. A maximum of ten files are permitted. MAKE checks the files' date/time stamps and if the first file in the line is not the newest of all, MAKE writes the command in the second line to a JCL file (CC/JCL) with necessary modification in the format. The command is usually "DO MC (...)", but other commands including DOS library commands are permitted, except the "DO" command other than "DO MC".

MAKE data file can contain any number of blocks. This version of MAKE checks all the file time stamps in the every first lines of the blocks before executing CC/JCL.

CC/JCL:

MAKE writes a Job Control Language file as its output. Suppose a MAKE data file of the following exists:

```
MYPROG/CMD MYPROG/CCC
DO MC (N=MYPROG,C,A,T=0)
HISPROG/CMD HISPROG/CCC
DO MC (N=HISPROG,LIST)
```

The resulting CC/JCL generated by MAKE should look like the following job stream:

```
//set C
//set A
//assign N=MYPROG
//assign T=0
//include MMC
//reset C
//reset A
//reset N
//reset T
//set LIST
//assign N=HISPROG
//include MMC
//reset LIST
//reset N
```

MMC/JCL:

Some modification is desired on MC/JCL. MMC/JCL has the modification of,

P:	preprocess only
C:	compile only (without preprocessing)
T:	default is drive 0
N:	source file, which contains main()
N1, N2, N3:	separate source files, which do not contain main()
NOMAIN:	indicates that the file of the subject does not contain main()

Examples for usage of MAKE

Compilation of a single source file. MAKE data file should look like:

```
PROG/TOK PROG/CCC
DO MC (N=PROG,P)
PROG/CMD PROG/TOK
DO MC (N=PROG,C,A,L)
```

Separate compilation of two source files. MAKE data file should look like:

```
MAIN/REL MAIN/CCC
DO MC (N=MAIN,P,C,A)
MODULE/REL MODULE/CCC
DO MC (N=MODULE,P,C,A,NOMAIN)
MAIN/CMD MAIN/REL MODULE/REL
DO MC (N=MAIN,N1=MODULE,L)
```

Revision of libraries. MAKE data file should look like:

```
PROG/CMD LIBC/REL
DO MC (N=PROG)
```

CAUTION: Because MC is fairly a large system, it is recommended that a hard disk or a RAM disk is used for both the system drive and the drive for the temporary files. Because MAKE requires files on such drives to have the time stamps, all the drives should have DOS 6 format. Especially a RAM disk should be DATECONV-ed before use (except for Michel Houdé's ERAMDISK, which is my favorite. Thank you, Michel.).

Sample MAKE data file to make MAKE/CMD

```
make/cmd make/ccc
do mc (n=make,t=0,k)
```

```
/* MAKE version 3.1 - C program maintenance utility */
#include <stdio.h>
#include <malloc.h>
#include <stat.h>
#include <string.h>
#define  LINE    80
#define  MC_JCL  "MMC"           /* a replacement for mc/jcl */
#define  CC_JCL  "CC/JCL:0"      /* target file */
#define  SPACE   ' '
#define  NKEYS   16              /* number of MC/JCL parameters */
struct stat *modtime;            /* file mode-time */
char *args[3] = {"do", "cc", NULL};
FILE *mfp, *jfp;
struct mcopt {                   /* MMC/JCL options */
    char *keyword;   int flag;
} opt[NKEYS] = {
    {"P",FALSE},     /* preprocessor */
    {"C",FALSE},     /* compiler     */
    {"A",FALSE},     /* assember     */
    {"O",FALSE},     /* optimizer    */
    {"L",FALSE},     /* linker       */
    {"MP",FALSE},        /* multi-pass     */
    {"CC",FALSE},        /* list source with assember output */
    {"LIST",FALSE},      /* display assembler output */
    {"K",FALSE},     /* kill temporary files */
    {"N",FALSE},     /* source program module with main() */
    {"N1",FALSE},        /* source program module without main() */
    {"N2",FALSE},        /* source program module without main() */
    {"N3",FALSE},        /* source program module without main() */
    {"D",FALSE},     /* disk drive for /ccc and /cmd */
    {"T",FALSE},     /* disk drive for temporary files */
    {"NOMAIN",FALSE},    /* tell assember the module does not
contain main() */
};
/* error messages */
char *msg[5] = {
    "\nusage: MAKE mak-file(w/o ext) [-e]",
    "illegal parameter. should be -e or -E",
    "mak-file not found",
    "can't open CC/JCL",
    "illegal end of a block in the data file",
};
main(argc,argv)
int argc; char **argv;
{   int parm_e, revise;
    char line1[LINE], line2[LINE], makfile[15];
    hello();                     /* opening prompt */
    parm_e=revise=FALSE;
    if (argc<2 || argc>3)
        progend(msg[0],0);
    if (argc==3)
        if (strcmp(strupr(argv[2]),"-E"))
            progend(msg[1],0);
        else parm_e=TRUE;
    strupr(strcpy(makfile,argv[1]));
    if ((mfp=fopen(strcat(makfile,"/MAK"),"r"))==NULL)
        progend(msg[2],0);
    if ((jfp=fopen(CC_JCL,"w"))==NULL)
        progend(msg[3],0);
    fputs("* C program maintenance\n",jfp);
    while (TRUE) {
        fgets(line1,LINE,mfp);
        if (feof(mfp)) break;  /* no more block */
        fgets(line2,LINE,mfp);
        if (feof(mfp))               /* illegal end of a block */
            progend(msg[4],1);
        if (do_block(line1,line2)) /* //set parameters */
            revise=TRUE;
    }
    fcloseall();
    if (revise)
        if (parm_e)
```

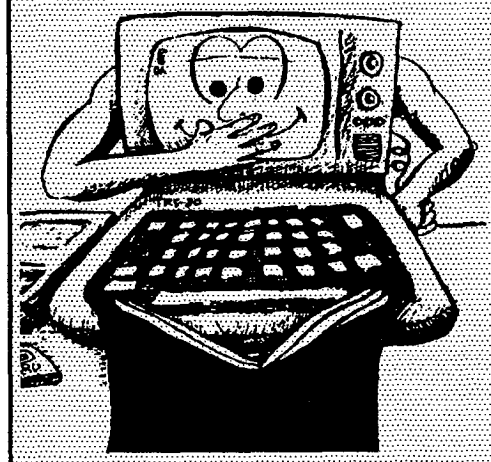
```
            printf("%s has been set.\n",CC_JCL);
        else
            execv(args[0],args);    /* issue DO CC */
    else puts("no revision detected.");
}
hello()
{   puts("MAKE version 3.1 - Jan. 1989 by Masa-aki Kitajima");
}
reset()
{   struct mcopt *p;
    for (p=opt; p<opt+NKEYS; p++)
        if (p->flag) {
            fprintf(jfp,"//reset %s\n",p->keyword);
            p->flag=FALSE;
        }
}
#define  MAXFILE  10
do_block(fline,command)
char *fline, *command;
{   int i, compile, max;
    time_t *f0_mod, *fi_mod;
    char files[MAXFILE][15];  /* array of filenames */
    char option[15], *p, *pf;
    for (i=0;i<MAXFILE;i++) {  /* set filenames to files[i] */
        pf=files[i];
        while (*fline!=SPACE && *fline!=EOL)
            *pf++=*fline++;
        *pf=NULL;
        if (*fline==EOL) break;
        else fline++;
    }
    max=i+1;                        /* i+1 files detected */
    compile=FALSE;
    if ((f0_mod=getmod(files[0]))==(time_t *)EOF)
        compile=TRUE;
    for (i=1;i<max;i++)
        if ((fi_mod=getmod(files[i]))==(time_t *)EOF) {
            printf("WARNING: %s not found\n",files[i]);
            compile=TRUE;
        }
        else if (*f0_mod < *fi_mod) {
            compile=TRUE; break;
        }
    if (!compile)
        return FALSE;           /* already compiled */
    if (strncmp(strupr(command),"DO MC (",7))
        fputs(command,jfp);         /* command is not DO MC */
    else {          /* if command is DO MC, handle parameters */
        command+=7;
        while (TRUE) {
            p=option;
            while (*command!=',' && *command!=')') {
                *p++=*command++;  /* set a parm to option buffer */
            }
            *p=NULL;
            setparm(option);  /* //set or //assign the parameter */
            if (*command==')')
                break;
            else
                command++;          /* skip the ',' */
        }
        fprintf(jfp,"//include %s\n",MC_JCL);
        reset();                    /* //reset parameters */
    }
    return TRUE;
}
setparm(parm)
char *parm;
{   char key[7], *pk, *orig; struct mcopt *p;
    pk=key; orig=parm;
    while (*parm && *parm!='=')
        *pk++=*parm++;
```

```
.MC/JCL - A revised version -
11/23/88 by Masa-aki Kitajima
//if -P&-C&-A&-L
//set P
//set C
//set A
//set L
//end
//if -N
//. ERROR: Must specify
"N=progname"
//quit
//end
//if -D
//assign D=1
//end
//if -T
//assign T=0
//end
//if P
.Preprocess #N#/CCC:#D# to #N#/
TOK:#T#
mcp #N#:#D# +o=:#T# ##>ERRORS/
CCC:#T#
//end
//if C
//if mp
mc1 #N#:#T#
mc2 #N#:#T# +o=:#T#
//else
//if CC
mc #N#:#T# +c +o=:#T# ##>ER-
RORS/CCC:#T#
//else
mc #N#:#T# -c +o=:#T# ##>ER-
RORS/CCC:#T#
//end
//end
//end
//if O
//assign EXT=OPT
.Optimize #N#/ASM:#T# to #N#/
OPT:#T#
mcopt #N#/ASM:#T#
//else
//assign EXT=ASM
//end
//if A
//if LIST
//assign LOPT="-nm-ne-nc-we"
//else
//assign LOPT="-nl"
//end
//if NOMAIN
.Assemble #N#/#EXT#:#T# to #N#/
REL:#T# (without main() func-
tion)
mras #N#/#EXT#:#T# #LOPT#
//else
.Assemble #N#/#EXT#:#T# to #N#/
REL:#T#
mras mc +i=#N#/#EXT#:#T#
+o=#N#:#T# #LOPT#
//end
//end
//if L
//if -N1&-N2&-N3
mlink -a=y,#N#:#T#-n=:#D#-e
//else
//if N3
```

```
mlink -
a=y,#N#:#T#,#N1#:#T#,#N2#:#T#,#N3#:#T#-
n=:#D#-e
//else
//if N2
mlink -
a=y,#N#:#T#,#N1#:#T#,#N2#:#T#-
n=:#D#-e
//else
//if N1
mlink -a=y,#N#:#T#,#N1#:#T#-
n=:#D#-e
//end
//end
//end
//end
//end
//if K
//if P
remove #N#/TOK:#T#
//end
//if C
remove #N#/ASM:#T#
//end
//if A
remove #N#/REL:#T#
//end
//if O
remove #N#/OPT:#T#
```

```
    *pk=NULL;    pk=strupr(key);
    for (p=opt; p<opt+NKEYS; p++)
        if (!strcmp(p->keyword,key)) {
            p->flag=TRUE;
            if (*parm)
                fprintf(jfp,"//assign %s\n",orig);
            else
                fprintf(jfp,"//set %s\n",orig);
            break;
        }
}
time_t *getmod(s)
char *s;
{   modtime=(struct stat *)malloc(sizeof(struct stat));
    if (stat(s,modtime))
        return (time_t *)EOF;              /* file not found */
    return &modtime->st_mtime;
}
quit()
{   fcloseall(); exit(0);
}
fcloseall()
{   fclose(mfp); fclose(jfp);
}
progend(s,i)
char *s; int i;
{   puts(s);
    if (i) fcloseall();
    exit(0);
}
```


TRSTimes magazine

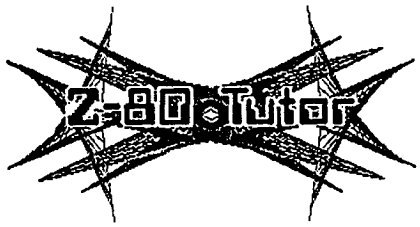


TRSTimes is the bi-monthly magazine devoted exclusively to the TRS-80 Models I, III & 4/4P/4D.

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



ASSEMBLY LANGUAGE TUTOR
VOLUME X: REFERENCE

by Chris Fara **NEW!**

Hot cakes! New from Chris Fara! Manual of reference to all Z-80 "op-code" instructions. For use in assembly programming on any Z-80 computer, including TRS-80 Models I, II, III and 4. It complements assembler packages which don't explain Z-80 op-code (most of them don't) and replaces unwieldy explanations in Tandy packages. Presented in Chris' famous common-sense style, neatly printed with bold headings and intelligent page layout for quick and easy reference. A must for all programmers and students of assembly language. Includes:

. Detailed descriptions of all instructions, their operands, sizes, timing, and effect on flags;
. Technical notes on all registers, flags, addressing modes, execution timing, plus many other interesting details;
. Complete alpha-numeric cross-reference listings of all mnemonics, operands and the corresponding machine codes;
. and the clever "Chris' Own" Quick Reference Table.

62 pages, bibliography, only $12.95
Add $3 S&H US - $4 Canada, AK, HI, PR, APO, FPO - $6 Overseas

PUBLISHED AND DISTRIBUTED BY COMPUTER NEWS 80 PO BOX 680 CASPER, WY 82602

FREE DISKS WITH YOUR SUBSCRIPTION ORDER TO
Computer News 80

Subscribe to Computer News 80 for one year, 12 Monthly issues, and we will send you a white box of 10 premium 5-1/4 double/single sided double density disks, with sleeves and labels. A $7.90 value absolutely FREE with your first issue.

Subscription rates are $24 in US for 12 issues. $35.50 in Canada, $36 all other.
Offer expires October 31, 1990, so hurry. Write FREE DISKS on your order.

Send Check or Money Order, Sorry no credit card orders, to
Computer News 80 PO Box 680, Casper, WY 82602

PowerSoft Products from MISOSYS, Inc

SUPER UTILITY PLUS: *The greatest utility ever written for the TRS-80. Every TRS-80 magazine has said so!*

Five-Star Excellent Reviews in *80-MICRO, 80-US, INFOWORLD, POPULAR ELECTRONICS, FAMILY COMPUTING, Creative Computing & more!*

If you use a TRS-80 with disk drives, then this is a must-have program that you will wonder how you did without for so long! Super Utility has won numerous awards, has received many 5-Star reviews and this could be your last chance to purchase a copy at this unheard of price. Super Utility does so many things, you will never use its full potential, but it isn't that hard to use since it is completely menu-driven with the most common defaults built right in. It is configurable for all the popular TRS-80 operating systems and will even allow you to set one drive for one system and another drive for a different operating system and copy files easily between the two. Even between Model I and III or 4, regardless of density, track number, number of sides, or system used. We have thousands of letters in our files over the years about how Super Utility has saved the user from various problems. Super Utility removes or decodes passwords (strips them right off a disk in one pass), reformats a disk without erasing the data, fixes problems, backs up most protected disks, etc. This was the very best utility ever written for the TRS-80 and now is the time to get your own copy. Super Utility has over 65 functions and features. Too many to describe! A fantastic buy. Does not work on hard disks. Our ToolBox or ToolBelt has similar features for hard drive use, as well as floppy. SU+ does not support Newdos/80 double-sided disks.

Specify Model I/III or Model 4/4D/4P: $4 S&H **$34.95**

LDOS ToolBox (Hard Disk Check, Repair, Modify, much more! Like a "SU+" for hard disk) $24.95

If you own a hard disk and use LDOS, this is the perfect insurance policy for your data. The LDOS TOOLBOX is like a Super Utility+ for hard disks. Features Disk Check and Disk Repair, Sector Modification, plus many, many other useful utilities that makes using a hard drive even easier. Each program contains a builtin help command, so many times you don't even need to look things up in the manual - just press <Enter> for help! A very wise buy for hard disk users.

Model 4 ToolBelt (same for Model 4 DOS 6 use. OK for 6.3. Like a "SU+" for hard disk) $24.95

This is similar to the LDOS TOOLBOX, except it is for the Model 4 TRSDOS 6 operating system (all versions).

Back/Rest - Super Fast Hard Disk Backup and Restore. Saves hours of time! For I, III or 4. $34.95

BACK/REST has proven to be a great time-saver for thousands of TRS-80 hard drive users. When reviewed by 80-MICRO, they gave it FIVE STARS - perfect! It saves hours of time and is very easy to use. BACK/REST can back up 10 megabytes in about 10 minutes and 20 meg in about 30-40 minutes. It also tells you how many disks to have ready. Works under LDOS or TRSDOS 6 (both versions on same disk). Great utility for hard disk users!

Superior Hard Disk Drivers for Tandy disk systems. Mix Model III and 4 easily. $49.95

These hard disk drivers out-perform the Tandy drivers in many ways. Our drivers allow you to combine LDOS and TRSDOS 6 on the same drive and boot from either system (with floppy disk). They run faster and take much less memory from the system. Only for use with Tandy Hard Drives. Available allso for Percom drives.

PowerMail Plus (Please specify Model 4, III/I.) 5 Star mailing list-data system! $4S&H $34.95

This program was written because all the other mailing list/data base systems couldn't keep track of all the types of data most folks wanted to keep track of. You needed speed, you needed hard drive support, and you needed a crash-proof data structure. PowerMail+ was top-rated (5 stars) in several publications and has never been topped. Works on floppies or hard disk under all popular TRS-80 operating systems. Allows importing of data from several other once popular mailing systems to avoid re-typing. Written in machine language by the author of Super Utility, this program is FAST and sorts up to 10 levels very quickly. If you keep track of names and addresses along with associated data for any situation, this is the one to use. Many churches, organizations and businesses use PowerMail+ for all the different kinds of lists they need to pull from. Each record has 24 user-definable "flags" to allow total customization for your exact needs.

Text-Merge Form Letter Module - Create customized "form letters" and Labels with PowerMAIL+! $15.00

This optional module for PowerMail allows you to create customized "form letters" or custom labels, lists, etc. with PowerMail Plus and any word processor that saves text in ASCII format. Very easy to use and really gets the effect you want. Allows completely definable report generating from your PowerMail+ data.

PowerDraw (animated TRS-80 screen graphics! Easy to use. Great for kids or adults!) $19.95

INFOWORLD, 80-MICRO and *80-US* magazines really loved this program when they reviewed it. It does many things and is fun to use as well. First, PowerDRAW allows you to create graphics (mixed with text if desired) and save them to disk. It also allows you to create up to 33 "frames" of animation and "play" them like a movie. It also allows you to save the graphics in several modes, including BASIC listings, CMD file format, and others. These can then be merged into your own programs, etc., either in BASIC or machine language! Many of PowerSoft's opening screens were created with PowerDraw. In fact, it even creates animated opening screens (like we use in Super Utility, PowerTool, etc.) to really pep up the program. It also allows you to print the screens on Epson-type and several other type of printers. Lastly, PowerDraw has the ability to load in many types of TRS-80 graphic's and convert them to BASIC listings like a BASIC program generator!

PowerDOT 2.0 for printers. Mix text with graphics - no problem. Build new fonts. $3S&H $19.95

This program is similar to PowerDraw, but quite different. It allows you to create "hi-res" type screen graphics combined with text, and allows you to create drawings much larger than your screen. The screen is a "window" to a much larger drawing area and you use the arrow keys to move about the drawing. In a way, it is similar to Macpaint for the Macintosh computer. It also allows you to create custom fonts for ads, etc. Many of our early ads were created with PowerDot. It creates the hi-res effect due to each TRS-80 block pixel being printed as a single dot. Please specify if EPSON, Okidata, Prowriter, or Radio Shack printer.

PowerDriver Plus for SuperSCRIPSIT and SCRIPSIT PRO and Epson printers $17.95

Allows EPSON or compatible printers to be fully utilized with SuperScripsit.
This is a replacement driver for the ones you got with SuperScripsit. It fully supports the various Epson and Epson compatible printers to the limits of their capabilities. Model I, III or 4 is supported in the same package. Easy to install. Once installed, works without any extra thinking. Thousands of happy customers!

Note: Technical support of these products is not included at these prices.

MISOSYS, Inc.
P. O. Box 239
Sterling, VA 22170-0239
800-MISOSYS [647-6797]

Unless otherwise noted, include $2S&H per program package in US; Canada is US+$1; Foreign is US*3. COD is $3.50 additional. Virginia residents add 4.5% sales tax, VISA/MC accepted.

ZOFAX™ 96/24

Send and receive fax 9600 bps + 2400 bps Data Modem

- Auto receive/print incoming fax
- Flip/rotate incoming fax picture
- Broadcast and distribution list
- Time schedule transmission
- Background receiving
- Fax mail merge
- No waiting in line
- No need of hard copy

The ZOFAX™ 96/24 Fax/Modem gives you the best value at an affordable price. It's a 2400 bps fully Hayes compatible modem with an integrated 9600 bps send/receive fax that is compatible with any Group III fax machine. Now you can turn your PC into a fax machine with the ZOFAX™ 96/24 Fax/Modem.

Zoltrix
Solely For Tomorrow's Technologies

Products available from MISOSYS, Inc.

Advanced Features:

- **Automatically receives and prints incoming fax messages.**
  Working like your regular fax machine, BitFax prints incoming fax messages directly to the printer and then saves it onto the disk.
- **Flips/rotates incoming fax pictures.**
  If the sender sends a page upside down, BitFax lets you flip the page so you can see it on the screen. If the page is sent sideways, you can rotate the page 90° so you can view it on the screen.
- **Time scheduled transmission.**
  You can schedule to send a fax at night or any time when the phone charges are less expensive.
- **Background receiving.**
  While receiving a fax message, BitFax lets you run other programs such as copying a file, viewing a file directory or running word processing software.
- **Fax mail merge.**
  Fax Mail Merge lets you fax customized letters to several people with their own name, address, or company information merged into a form letter. The names, companies, addresses or other information is pulled from the phone book database which is a dBase format. You can create a comprehensive database with dBase and use the data for the Fax Mail Merge. Doing so will save you time and get impressive results from your recipients.
- **Broadcast and distribution list management.**
  BitFaX lets you send a message to several people selected from the phone book. You can also assign names into groups and just send to the selected groups without having to key in or select their numbers each time you send. For example, you can put all the people in the sales department into a group and just send to the group rather than fax to each individual name each time you fax.
- **Optional BitPaint for editing**
  With an optional BitPaint program, you can view a received fax image, cut it, change it, edit it or write your own reply and fax it back to the senders.

Features and Benefits:

- Integrated data modem and send/receive Fax capability on a single board.
- Auto-answer: the modem will detect whether it's data or fax.
- Supports Multiple File Formats. You can send any of these file types directly from your PC to any Group III Fax machine: ASCII (text), PCX (pc Paintbrush), IMG (GEM Artline), TIFF (scanner) and FAX (received from proFAX PC Fax card). Color PCX Files will be converted to gray scaling.
- Auto redial.
- Sends Multiple Files. So you can combine more than one file to be sent to a fax machine. For instance, you might choose a picture file with graphs or logos, and combine it with text message, and then send every thing with one phone call. Doing so will reduce phone charges, as well as save the time needed to make multiple calls.
- Flexible phone number management. You can add, change, delete and sort phone numbers easily with BitFax. The BitFax Phone Book also allows you to store company names and addresses. You can group sets of phone numbers for distribution lists. The Phone Book Database is stored in a dBASE III type format (.DBF file name extensions), so you can access with dBASE or other utility program.
- Software selective for data modem and Fax mode - software automatically switches between data modem or fax.
- "AT" command set compatible for use by most communications software.
- Low transmission cost - schedule to send fax at night rate.
- Fax's are better quality - No degradation due to printer quality or scanner quality.
- Distribute fax messages to multiple destinations.
- Highest speed and lowest cost solution by combining 9600 bps fax and 2400 bps data modem on one board.

Technical Specifications:

- **Compatibility: Data modem**
  300 bps Bell 103
  300 bps CCITT V.21
  1200 bps Bell 212A
  1200 bps CCITT V.22
  2400 bps CCITT V.22
- **Compatibility: Facsimile**
  Send and receive Fax from CCITT Group III fax Machines. Fax format compatible with CCITT T.30 and T.4 Group III Fax 9600 bps CCITT V.29 ter with fall back to 7200, 4800, 2400, CCITT V.27 ter CCITT V.21 (Synchronous)
- **Command language.**
  "AT" command set compatible with extensions provided for fax operations.
- **Data format.**
  Asynchronous operation supports 7 or 8 data bits, plus start, stop and parity bits.
- **Operations.**
  Full or half duplex with pulse and tone dialing. Dial up or leased phone mode with auto answer mode.
- **Audio Monitor.**
  On-board monitoring with software speed control.
- **Line equalization transmit.**
  Adaptive equalization for PSK/QAM mode.
- **Test and diagnostics.**
  Install and Test program
  Self Diagnostics
- **Modular phone jacks.**
  Two RJ11C phone connectors for phone and line.
- **Approval.**
  FCC part 15, Class B
  FCC part 68
- **Warranty.**
  1 year warranty.

MISOSYS, Inc
PO Box 239
Sterling, VA 22170-0239
800-MISOSYS (647-6797)

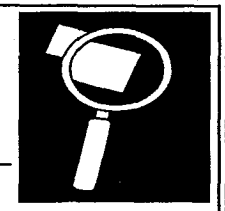
Are you still fussing with floppies for BACKUP? Our QIC40 DJ10 JUMBO tape drive from CMS is your solution!

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. An adaptor board can be used to connect a DJ10 when all FDC connectors are in use. Kits are available for XT/AT or PS/2 machines which convert Jumbo to external use.

- In about 5.5 minutes, a DJ10 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 DC2010 tape cartridges.
- The DJ10 plugs into your floppy disk controller to save cost, power, and a slot. Needs 5-1/4" (or 3.5" with faceplate) mounting slot.
- Optional adapter 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)**	Note: DJ10 price includes one DC2000 tape
Tape Adaptor	**$75 ($3S&H)**	
External Adaptor	**$110 ($5S&H)**	
DC2000 tape (40M)	**$20.00**	
DC2120 tape (60M)	**$25.00**	

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 ✓ Jump about subdirectories
- ✓ Keep sectors & clusters for writing to a file; Great for partial file recovery
- ✓ Edit bytes in hexadecimal or ASCII, zap in 0s; 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 + $4S&H

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

800-MISOSYS
or 703-450-4181

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 + $2S&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 SuperScripsit document files to DCA-RFT. Only **$89.95 + $3S&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 ✓ Background receiving
- ✓ Distribut Fax messages to multiple destinations ✓ Fax mail merge
- ✓ Time schedule transmission to take advantage of low nightime rates
- ✓ 2400 bps Fully Hayes Compatible Modem
- ✓ Includes powerful but easy to use BITCOM and BITFAX software

Special Introductory price: $225 + $6 S&H (price good through 01/31/91)

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

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 $179.10 + $6S&H.**

LS-DOS 6.3.1 has something for everyone

- ☆ 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** (+$2 S&H).

MISOSYS, Inc.
P. O. Box 239
Sterling, VA 22170-0239
703-450-4181 [orders to 800-MISOSYS (647-6797)]

DoubleDuty Version 2.6.0

DoubleDuty works with all expanded memory known to the DOS; a BANK parameter lets you use any two adjacent pair of memory banks for its "second 64K Model 4". If you have an XLR8er board installed with 256K additional memory (banks 3-11), you could install DoubleDuty using banks 10 and 11. In this way, you would have a 128K Model 4 partition and a 64K Model 4 partition still with six more memory banks available. Programs which use extra memory but are not "expanded memory literate", such as Multiplan or enhanced Visicalc, could be installed in one DoubleDuty partition and use 128K, still having another program installed in the other DoubleDuty partition. Have PRO-WAM and DoubleDuty both installed above bank 2 in expanded memory. What a machine!

If you already own DoubleDuty and have expanded memory known to the DOS (see the MEMORY command of LS-DOS 6.3.1), then you really ought to have this version of DoubleDuty. Just send your original DoubleDuty master disk to MISOSYS for a disk refresh; the fee is $10 plus S&H ($2 in U.S. & Canada, $3 elsewhere). It doesn't matter if your DoubleDuty was purchased from Radio Shack or direct from MISOSYS; I'll take it either way and convert it into a 2.6.0 "expanded memory literate" version.

If you don't already own it, maybe now is to time to think about acquiring DoubleDuty; it. turns a 128K (or greater) machine into two partitions with a third able to invoke DOS library commands. This lets you run two programs concurrently switching between either at the flick of a function key. It doesn't support multitasking, so only the foreground application receives CPU time. But DoubleDuty does let you run two programs such as Multiplan and Superscripsit without having two terminate one to run the other. **DoubleDuty is $49.95 + $2 S&H (U.S.).**

MISOSYS 20 or 40 MB Hard Drives for your Model III or 4

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/from 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.

1991 Prices currently in effect:

Complete Drive Kits:	
20 Megabyte kit:	$475
40 Megabyte kit:	$595
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	$95
Adaptec 4010 HDC	$95
Drive power Y cable	$5
XT drive cable set	$5

Note: freight charges are additional.
Prices subject to change without notice

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] **$74.95 + $5S&H**

LB Data Manager

A flexible data manager

LB is easily used by anyone for managing their data. It's menu driven for ease of use; absolutely no programming needed. Requires a Model 4 with 128K or a hard drive. LB86, an MS-DOS version is also available. Now activate PRO-WAM from newly compiled LB beta release (hardware restrictions apply)

- Store up to 65534 records per data base
- Up to 1024 characters (64 fields) per record
- Nine field types for flexibility
- Select and sort on up to 8 fields (multiple indexes)
- 10 input/update screens per data base
- 10 printout formats per data base
- Extensive on-line help available

LB [L-50-510] **$74.95 + $5S&H**

All Model I/III/4 Language software 30% off until March 31st

BSORT: high speed array sort for BASIC	$10.47 + $2S&H
DSM: high speed disk virtual sorting utility	$41.97 + $2S&H
DSMBLR: Disassemble CMD files to disk	$17.47 + $2S&H
EDAS: Macro assembler and full screen editor	$31.47 + $4S&H
EnhComp: BASIC compiler with in-line assembler	$41.97 + $4S&H
HartFORTH: 79Standard FORTH runs under DOS	$34.97 + $3S&H
MC: A complete K&R C compiler (requires MRAS)	$55.97 + $5S&H
MRAS: Advanced relocatable object code assembler	$41.97 + $4S&H
RATFOR (M4 only): Converts RATFOR to FORTRAN	$41.97 + $4S&H
UNREL: convert /REL object modules to /ASM	$20.97 + $2S&H

Specify Model I/III or Model 4

Floppy Drives and Accessories

5.25" 360K 1/2-height	$75+$4 S&H
3.5" 720K in 5.25" 1/2-height frame	$85+$3 S&H
2SV5 drive case & P/S	$60+$5 S&H
Single drive host cable *	$10
Dual floppy extender cable **	$15

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 can also provide internal FDC to floppy ribbon cables for 4/4D/4P computers.

MISOSYS, Inc.
P. O. 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