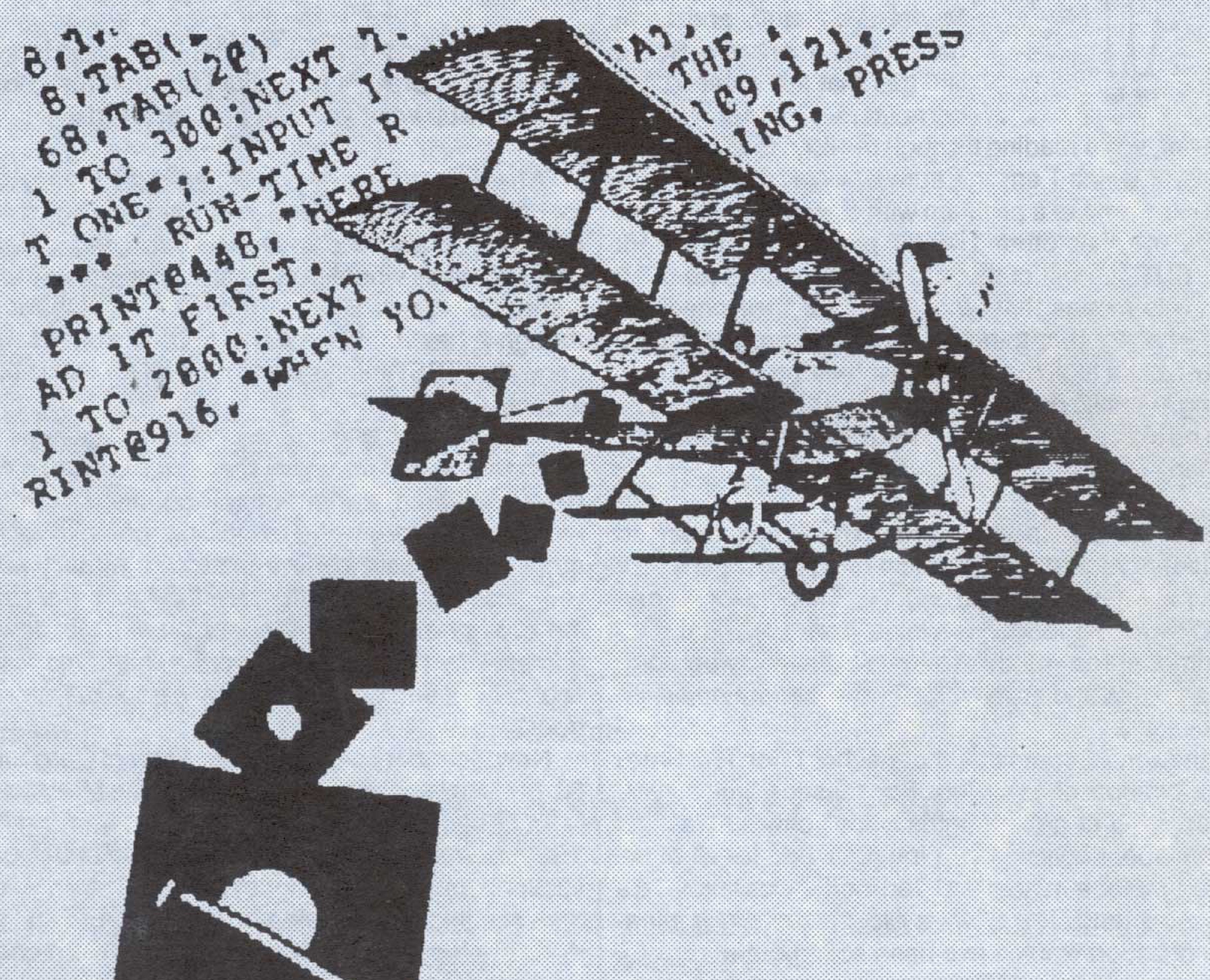
TRSTimes

Volume 4. No. 2. - Mar/Apr 1991 - $4.00



LITTLE ORPHAN EIGHTY

The other day I looked through some recently acquired TRSLINK disks and I came across a file called ALLFILES/ASC. This file is an advertisement from a bulletin board in Orlando, FL which lists all the files available for download. It certainly is an impressive list, containing file after file of pure TRS-80 Model I/III/4 stuff, and I am happy that there are BBS's like this that still cater to our computers.

However, while reading through the list, I found several programs of which I am the author. Now, don't get me wrong, I do not have a problem with that; to the contrary, it pleases me that other TRS-80 people find my work useful.... BUT.... most of these programs were sold to 80 MICRO. This means that when I cashed their checks, I transferred all rights to them. They own the programs, and they may have a legitimate beef that their property is now appearing in public domain forums. On the other hand, as they are no longer in business, they may not care one way or another.

This brings up another item. TRSTimes received several letters regarding the article by Ted Barnes in our Jan/Feb 1991 issue. Near the end, Ted offers to locate and share software no longer available from suppliers. Our letter writers perceived this to be an offer to swap commercial software and asked if TRSTimes condonded such blatant pirating.

The answer, of course, is that neither I nor TRSTimes condone pirating of any kind. For reasons which have been discussed at great lengths since software was first stored on tape, I believe that pirating is bad for all of us. Had I thought for a minute that Ted's offer was linked to a desire to accumulate software illegitimately, the paragraph in question would have been edited out.

There are legal means of swapping or selling commercial software. I checked with an attorney friend and received the following non-binding free advice:

- Selling, re-selling, swapping or giving away legally copyrighted commercial ORIGINAL MASTER disks and/or printed documentation is legal.

- Selling, swapping or giving away COPIES of legally copyrighted commercial software and/or printed documentation is illegal.

- You may make as many copies of your legally obtained, legally copyrighted commercial software and/or printed documentation as you wish, as long as it is for your own PERSONAL use.

- Making copies of your legally obtained, legally copyrighted commercial software and/or printed documentation for your own personal use, and then selling, re-selling, swapping or giving away the ORIGINAL MASTER disks and/or printed documentation is illegal.

There are additional points which are too lengthy (and technical) to present here, but I feel that the above pretty well sums up what you can and cannot do.

While I am on the subject of pirating, let me just mention that pirating does not end with the copying of disks and manuals; it also includes photocopying magazines. I know of cases where people have copied entire issues of Computer News 80 and the Misosys Quarterly, and, though it has not been done right in front of me, I also know that TRSTimes is being copied liberally and spread around.

The ones who participate in this practice must obviously find things of value in the magazines. To these individuals I direct the following thoughts:

Why go through the trouble of photocopying? How much are you saving? Five or six dollars per year? Surely your time must be worth more than that, not to mention the cost of the gasoline spent on the trip down to the printer! Why not simply subscribe to the magazine(s) of your choice and get your very own copy delivered to your door?

Now, I cannot speak for Stan Slater of Computer News 80, nor can I speak for Roy Soltoff of the Misosys Quarterly, but I can certainly tell you what the thinking is here at TRSTimes. The bottom line is the subscriber base. The subscribers pay their money, which in turn is spent on duplication costs, envelopes, labels, paper and postage. What is left over is spent on equipment maintenance and upgrade. There is no profit at TRSTimes. It was never the intension to make a profit. However, when the time comes that the subscriber base has shrunk to the point that the expendiatures are being paid out of my pocket, believe me, TRSTimes ceases to exist at the end of that year. Though I do love the TRS-80, I am not a masochist. I will simply go where there is still a market, and you all know where that is!

Don't misunderstand, TRSTimes is still going strong; I simply wanted to point out that pirating serves no one in the long run. Getting something for nothing just doesn't work.

Lance W.

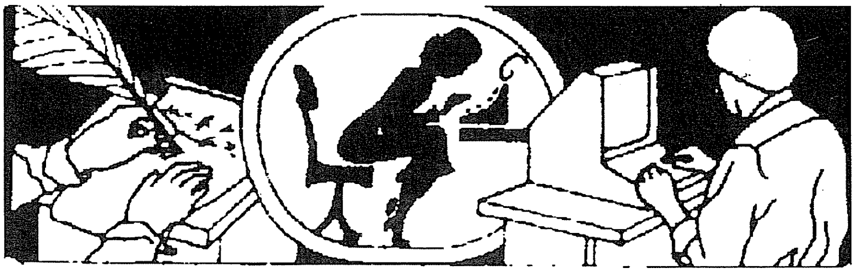
TRSTimes magazine

Volume 4. No. 2. - Mar/Apr 1991

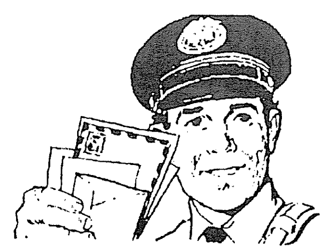
PUBLISHER EDITOR
Lance Wolstrup
CONTRIBUTING EDITORS
Roy T. Beck
Dr. Allen Jacobs
Dr. Michael W. Ecker
TECHNICAL ASSISTANCE
Members of:
San Gabriel Tandy Users Group
Valley TRS-80 Users Group
Valley Hackers' TRS-80 Users Group

TRSTimes magazine is published bi-monthly by TRSTimes Publications. 5721 Topanga Canyon Blvd, Suite 4. Woodland Hills, CA. 91367. (818) 716-7154.
Publication months are January, March, May, July, September and November.

1991 subscription rates (6 issues):
UNITED STATES & CANADA: $18.00 (U.S. currency)
EUROPE, CENTRAL & SOUTH. AMERICA: $23.00 for surface mail or $29.00 for air mail. (U.S. currency only)
ASIA, AUSTRALIA & NEW ZEALAND: $25.00 for surface mail or $$32.00 for air mail. (U.S. currency only)
Article submissions from readers are welcomed and encouraged. Anything pertaining to the TRS-80 will be evaluated for possible publication. Please send hardcopy and, if at all possible, a disk with the material saved in ASCII format. Any disk format is acceptable, but please note on label which is used.



THE MAIL ROOM



NEWDOS/80

Serendipity! With your #6 disk you enclosed the PDRIVE settings to let NEWDOS/80-V2 read the 35 track SD disk. Out of curiosity and respect for the old 'hacker' DOS I got a copy of it a while ago, but the manual is tough sledding. To copy files to it from TRSDOS 1.3 I used SU 3.2, but would not want to make a habit of it. What a pleasant surprise to find COPYAID on your first disk; now I can make backups of the DOS right in drive :0 where it will be used. On one copy of it I left SD settings for drive :1; mine is a 4 drive system, and now I can copy files from SD on :1 to a NEWDOS data disk in drive :2. NOTNAMED is really not a very inspired name for any kind of data disk.

Your article with the COPYAID/BAS listing in the May 1988 issue is almost a tutorial on how to manipulate the most important functions of NEWDOS.

Unfortunately there is no local GURU left in this city of a half million people who I could turn to on matters of the TRS-80 or NEWDOS. For too long I stuck it out with a tape based Model I that was upgraded to 48K in the computer (no expansion interface)! Our local TRS-80 club fell apart when invariably, one by one, members went to MS-DOS. How I marvel about the things people in the Alternate Source did with NEWDOS, and then in Northern Bytes later on. Now even Codeworks is folding; it seems that, more and more, computers are used as business appliances or toys to play games. The genuine enthusiast and hacker is a dwindling species. Just look at the magazine rack in the local drug store or book seller; maybe one or two computer magazines where there used to be shelves full of them.

It is comforting to have individuals like you to keep the TRS-80 flame burning. Sometimes I am embarrassed that all I can offer are more questions and not some hints, programs, or penetrating articles. I have no sense of rhythm or musical abilities; qualities that correlate with creative programming or Morse code (which I have tried in the past, but failed). Can you maybe, sometimes as fillers in TRSTimes, give as examples the EXACT syntax of the NEWDOS commands to format a data disk with a name, the various copy formats and what they will do, DO files and what they can and cannot do. The SYSTEM command has let me change the cursor. What really amazes me with NEWDOS is that it reboots instantly to DOS Ready with date and time correct, and that it does not stumble over passwords that really are not there, like the new dating scheme in LDOS or LS-DOS. Even the much patched TRSDOS 1.3, when used to CONVERT, still chokes on it. The only way out is SU 3.2 to strip all passwords and that scrambles the date and time stamp. It baffles me that TRSDOS 1.3, a Radio Shack product, takes by far the most time to boot or reboot on my Model 4. But I can live with that. What I like about 1.3 is that it a standard and that it puts no SYSTEM drivers or filters in HIGH$.

Keep the TRS-80's humming.
Willi Wald
Hamilton, Ontario
Canada

I am pleased that you find COPYAID useful. It was originally used as a class project in my Advanced Basic class some years ago. The idea was for the students to form teams to study the manual, and then for each team to write a module of the program. I merely took their work, cleaned it up a bit and added a couple modules. Incidentally, the assembly language version of COPYAID is on the same disk. It works identically, but is much faster (especially the mass delete module).

NEWDOS/80 is, in my opinion, the finest Model I/III Dos. Unfortunately, most users were completely overwhelmed by all the capabilities. The manual, written in 'programerese' for programmers by a programmer, also scared many people back to TRSDOS 2.3 or TRSDOS 1.3. Too bad, this Dos does it all, if you know how, or where to look in the manual. Your idea about presenting NEWDOS commands is good. We will begin immediately by taking a look at the dreaded PDRIVE command elsewhere in this issue.

I was not aware that Codeworks were having trouble, but in checking around, I have confirmed that they have, indeed, published their last issue. That is a real shame. They had a good idea, but I feel they should have stayed with the TRS-80 crowd. Presenting type-in programs for the MS-DOS user will not work, at least that has been my experience.

I am glad that the TRS-80 people still have the pioneer spirit and hacker mentality.

Ed.

REMOVE vs. KILL

I read the article "TRSDOS 6.3.1 Custom Patches" in the Nov. issue of TRSTimes and decided to check it out. I do not yet have a copy of LS-DOS 6.3 and decided to see how it differed from my copy of TRSDOS 6.2.1 (Level XT).

The most useful of the listed patches for me is the replacement of REMOVE by KILL. As an old Model I user, I still like the KILL command. Patching SYS1/SYS with the listed patch didn't work, giving the usual "FIND LINE

MISMATCH". So, I used Super Utility 4 and found the problem. The address for the patch is 0DH for TRSDOS 6.2, instead of the 15H for LS-DOS 6.3. It is located between PURGE and RENAME as shown on the right side of the display of sector 2. By using the zap procedure, I was able to make the patch directly. I was also able to use the following patch to do the job, after removing the password. (I imagine that using the password of LSIDOS might work.)

```
PATCH SYS1/SYS:d (D02,0D=4B 49 4C 4C 20 20;
F02,0D=52 45 4D 4F 56 45)
```

The ":d" is the drive number where the disk to be patched resides.

The DIR/D, CAT/C, BOOT/UPPERCASE, BOOT/CLOCK, and EXTENDED ERROR patches appear to work as listed, apparently the same addresses are used in both DOS versions.

When used, the KILL command results in the statement "Removing filename" displayed during the operation. KILL is listed in the LIB in place of REMOVE when the patch is made. The reverse patch works if one desires to go back to REMOVE.

Incidentally, for those who want to use "REMOVE" in both Model III (TRSDOS 1.3) and Model 4, the address for the "KILL" command in TRSDOS 1.3 is 81H, sector 3, of SYS1/SYS. I was able to make the switch using SUPER UTILITY, but was unable to do it as a patch.

TRSTimes has sure helped me a lot to understand more about the TRS-80, and to use it better.

Leonard A. Falevitch
Phoenix, AZ

PROGRAM LISTINGS

The last issue (Jan/Feb 1991) was the best yet. I enjoyed all the articles, but the standouts were 'The Movie Librarian' and 'Puzzle Exchange'. I am retired and have a lot of time on my hands, so I am spending it trying to learn Basic. Typing in your program listings is a lot of fun and has been very helpful, teaching me some tricks and techniques. I also enjoyed the 'Setdate' article, but I had a lot of trouble with the DEF FN lines. I finally got them to work, but have no clue to what they do. Guess I'll have to study some more. Keep your fine magazine coming.

Daniel T. Mangione
Miami, FL

Don't feel bad...I don't really understand the DEF FN's either; I simply take Mr. Rosenfelder's word for their validity. Thanks for the nice words.

Ed.

THE MOVIE LIBRARIAN

I have been struggling with PROFILE to create a database to organize my video tape collection. What a relief to find Art Molz's MOVIES/BAS. This program does just about everything I could not get PROFILE to do for me. Please publish more of Art's stuff. Two questions: What is TRASHMAN? and what is a Model IP?

J.A. Lipschutz
Muncie, IN

While PROFILE is a good program, I must admit that it is not my preferred database. Art's program does exactly what it is supposed to do, which just goes to show you that there's still talent left in the TRS-80 world.

TRASHMAN is a program that will virtually eliminates Basic's 'garbage collection' of strings. It was written and published by PROSOFT, the same people who did ALLWRITE. Unfortunately, they are no longer in business.

A Model IP is a portable Model I. Anyone who owns, or has owned, a Model I knows that carrying it around is not an easy task. Art has solved that problem. Maybe we can talk him into submitting an article, telling us exactly what he did.

Ed.

NEED HELP

Frank Carbone is looking for a spreadsheet program to run on his Model I. if anyone can help, please contact him at 11110 Deerpath Ct., Charlotte, NC 28262.

MODEL 1000 FOR SALE

Tandy 1000, dual drives and 20 MB hard drive, 640K, DWP 220 printer, monochrome monitor. WordPerfect on hard drive. Includes new desk. $950.00
Call Laura Aman (213) 697-2781

EDTASM FOR MODEL 4

Somewhere in one of the articles in TRSTimes magazine I saw an article where you spoke of a program that would patch EDTASM to work on the Model 4. If you still have this available, I would sure appreciate a copy.

Your magazine is the only real help for the Model 3 and 4 that I can get here in Columbia, SC. It seems that all of the people who were interested have given up and gone on to something else.

I have all your TRSTimes magazines from the very first to the latest edition, and many times I go back through when something really stumps me.

Your last two were really right on time. I just bought a used 15 Meg Hard Drive and didn't know anything at all about it, but the articles from Roy Beck gave me a lot of the basic things that I needed to know. I still haven't gotten it up and running, but I think something is wrong inside the drive.

Thanks for all the help you have given me with my computer.
William R. Salisbury
Columbia, SC

By the time you read this, you should already have received the file EDIT4/CMD, which will patch Model III Radio Shack EDTASM to run perfectly in Model 4 mode. Hope you enjoy it, and please send us all the great assembly language programs you develop with it.

Ed.

BASIC BUG?

I understand from the TRSDOS 6.x manual that I may define up to 10 USR statements to be in effect at one time, numbered 0 to 9. Consider the following program, which uses a simple machine language routine to display chr$(191) 23 times on the screen.

```
1'LSDOS 6.3.1 with Basic 01.01.02 enhanced by LSI
10 DEFINT A-Z:DIM J(7)
240 CLS:PRINT@(5,43),"READING LASTS APPROX. 5
SECONDS:PRINT CHR$(15);
246 FOR I=0 TO 7:READ J(I):NEXT I
247 DATA 3902.1536,2,3584,191,8448,1332,-13841
248 DEF USR0=VARPTR(J(0))
249 INPUT" press <ENTER> to start ",A$
262 FOR I=0 TO 22 '23 chrs to print
264 J(6)=J(6)+1     'move 1 print position
265 X=USR0(0)       'call the SVC 15 via USR
266 FOR K=1 TO 550:NEXT 'delay 5 seconds
272 NEXT
```

This works fine when the USR digit is 0. It also works correctly when the USR digits are 2 to 9. However, when the USR digit is 1, the program crashes. What am I doing wrong?
Fred Cornet
Amsterdam, Holland

This certainly presents an interesting problem. Let me clarify just what the program is trying to do. To demonstrate the power of using machine language routines from Basic, the program uses the SVC 15 DOS call to display 23 CHR$(191) starting at line 5, column 52. Fred uses a technique known as a 'magic array', undoubtably borrowed from Lewis Rosenfelder, to set up the machine language routine in the J array. When broken down, the routine is this:

3E, 0F,00,06,02,00,0E,BF,00,21,34,05,EF,C9

This translates to:

```
LD      A,15
NOP
LD      B,2
NOP
LD      C,191
NOP
LD      HL,0534H
RST     28H
RET
```

The NOP's are there to balance the array correctly (see Basic Faster & Better for further explanation of 'magic arrays').

This machine language routine (without the NOP's) is right out of the Model 4 technical manual and should work without a hitch. It does work when called with USR0 and USR2 to USR9; but, as Fred correctly states, it crashes when called with USR1.

As far as I can tell, Fred is doing everything right. The Basic code is correct; the machine language code is correct. If either were bad, the program would crash when called by any USR. The only thing I can think of is that Fred may have uncovered a bug in Basic (in which case it goes all the way back to TRSDOS 6.2.1 Basic).

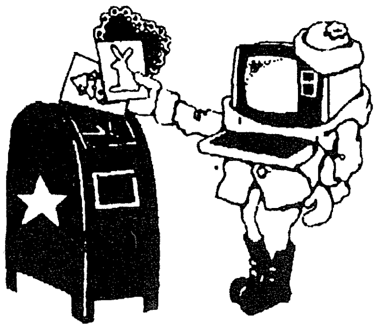
Now, I can get the program to execute correctly using any USR digits, but I will have to change the code a bit.

I have never been a fan of storing ML routines in arrays; instead, my personal preference is to store them in strings, so the code would look something like this:

```
1 'LSDOS 6.3.1 with Basic 01.01.02 enhanced by LSI
10 CLS
20 FOR I=0 TO 10:READ A:J$=J$+CHR$(A):NEXT
30 DATA 62,15,6,2,14,191,33,52,5,239,201
40 J=PEEK(VARPTR(J$)+2)*256+
PEEK(VARPTR(J$)+1)
50 DEF USR0=J
60 INPUT" press <ENTER> to start ",A$
70 FOR I=0 TO 22
80 POKE J+7,PEEK(J+7)+1
90 X=USR0(0)
100 FOR K=1 TO 550:NEXT
110 NEXT
```

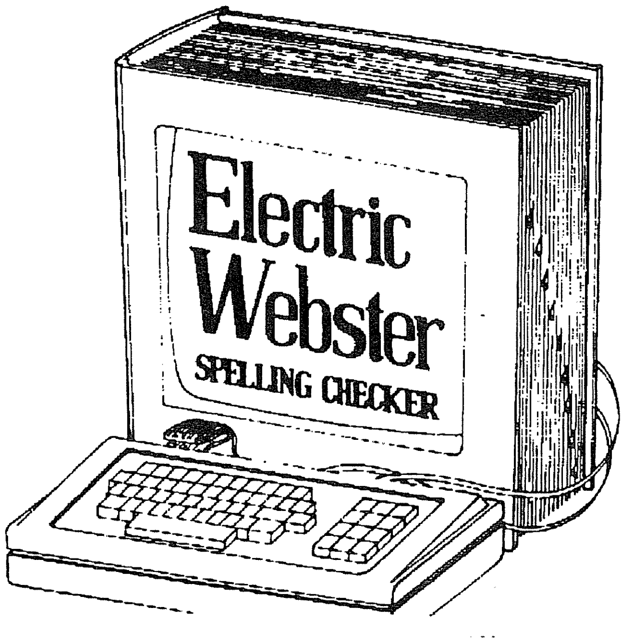
So the question of why does USR1 crash, when used with the integer array, remains unanswered. If anyone can come up with a reasonable explanation, both Fred and I will be extremely interested.

Ed.



Electric Webster
The "Cadillac" of spelling checkers!

By Roy T. Beck



INTRODUCTION

As many of you know, there are several spelling checkers available for the TRS family of machines. I don't pretend to know much about the others, but since I have and use Electric Webster, I will talk about it.

Basically, Electric Webster (EW) is a stand-alone spell checker which can work with any ASCII file. It is available in several different versions, tailored to work with different word processors. Mine is, of course, tailored to match AllWrite.

EW operates by itself, and therefore is unavailable in the editing mode. With AllWrite, I can call EW from the editing mode. This causes Allwrite to SAVE the file, and then transfers control to EW. When EW is all through, AllWrite and the working file are reloaded and you are returned to the EDITing mode.

EW can also be called from DOS Ready, and will return there when it finishes. My version of EW expects a straight ASCII file. I would have trouble if I loaded a Scripsit file with embedded formatting characters. There is a version of EW for use with Scripsit, but I don't have it.

There are actually two versions of EW, one identified as the "Standard" version and the other one as the "Self-Correcting" version. The latter costs more. I don't see the point of the Standard version, as it only calls your attention to a suspect word; you have to make note of it and come back later in your spell checker to fix it. The Correcting version allows you to fix it during the spell-checking phase, which makes more sense to me. Anyway, I use the Self-Correcting version.

EW does not have the largest dictionary in the micro world; I think it is limited to about 30K words. These are included in two files which are a part of the EW package. A third dictionary file is on the disk when you receive it, but it really is a kind of a blank, formatted file. It contains only (as I remember) Aardvark and some word beginning with a "Z". As you add additional words of your own, they are evidently stored alphabetically in this third file. I say "evidently", because EW uses a compression algorithm to store its dictionaries, which means the dictionaries themselves are unreadable by zapping programs. The compression algorithm greatly reduces the space required to hold the dictionaries, and also speeds up disk access. I am spoiled by having a hard drive available on each of my 4P's, so for me EW moves right along.

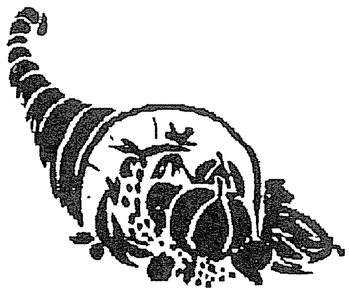
I believe EW is not case-sensitive; at any rate its suspect words always show up in lower case.

To make EW more usable, there are some things which definitely need to be added to the dictionary. One item is to add all official two-letter abbreviations for state names, such as CA for California, and MO for Missouri. Others would be your own name and home address (street and city). You could also consider adding names and addresses of people you write to frequently.

MY VIEWPOINT

I run AllWrite and EW from my hard drive, so the speed of response is pretty good, much better than from floppies, although that will also work. Since EW has to scan its entire dictionary each time it runs to insure matching all the words in your text file, speed of operation is very important.

I just now ran EW on this file, which at the moment consists of 10663 bytes or 1772 words. In just 45 seconds, it saved the file, loaded EW, loaded my text, and scanned it for unrecognized words. By the one minute mark it had



allowed me to correct and/or add to the dictionary the roughly 10 unrecognized words it found. By one minutes and 30 seconds total elapsed time I was back to typing this sentence. Obviously it would be slower with floppies. It would be somewhat faster with Peter Ray's (Anitek) speedup, or with the XLR8er board, neither of which I have working.

It is possible to edit the 3rd file, which holds your added words. This is useful if you have mistakenly entered an incorrect word and only recognize this fact later on. The method of removing such words is a bit tedious, but it can be done. I suspect a clever hacker could even edit the two original files, but I haven't yet got around to trying that. As far as I know, the size of the additional (3rd) file is only limited by your DOS and your disk space.

RESPONSES TO UNKNOWN WORDS

After checking a file, EW displays each suspect word in turn, and offers a list of options to the user.

- 1. Typing a question mark (?) will cause EW to later display the word in context so as to better understand the reason for the "suspect" flag.

- 2. Typing an @ sign will cause a portion of the dictionary to be alphabetically listed on the screen, with a gap where the suspect word would fall. The dictionary list may be scrolled up or down to find the correct word if it is in the dictionary. By placing the correct word at the bottom of the screen and hitting ENTER, the the correct word is positioned for easy insertion.

- 3. The correct word will be seen at the top of the screen. Hitting the period key (.) will replace the suspect word with the word previously positioned at the bottom of the screen in #2 above.

- 4. To add a suspect word to the dictionary, Hit the plus (+) key.

- 5. If the word is OK but should not be added to the dictionary, hit ENTER. The word will be left alone. This is appropriate for "jargon", place names, surnames etc.

EW ignores all single letters and punctuation, and therefore will not bother you with people's initials and the like. By the same token, if you typed "U" instead of "I" as a word in a sentence, EW won't find it. You MUST still read your own text after EW operates, but it will pick up a large share of your clinkers.

OTHER FEATURES

As an option at purchase time, you can buy Hyphenation and Grammar checking modules. I have both, but, having used them, I am not sure I would recommend them to others.

The hyphenation module inserts possible "soft" hyphens into all long words in your file. AllWrite then uses the soft hyphens as it sees fit to right justify your text, making up lines with the minimum amount of inserted spaces. This works well, but since most of my essays end up being formatted by someone else's editor or publisher, I don't make much use of it. If you do the final formatting and printing of your material, this would be very helpful.

Occasionally I do two column work, and here the hyphenation is a boon, as right justified text in multicolumn form can get to be mostly white space, especially if you tend to use long words, as I do. Hyphenation also improves the appearance of ragged right formatting, as the variation of line lengths is reduced.

REACTIONS

I have some comments and complaints about EW. First off, it seems not to include plural forms of words in its built-in dictionary, and so it flags every plural word as a suspect word. You have to add these to the third file as you meet them. Apparently there is no good algorithm for forming plurals, so the problem wasn't solved by that route.

Another item is that EW treats hyphens and apostrophes as breaks between words, so contractions such as "aren't" get segregated into two words. Since EW ignores all single letters, the "t" is discarded in this case and EW comes back with "aren" as a suspect word. This should be improved, in my opinion, but at this state of the TRS market, the author is not doing any fixing; he is just selling the product.

If you add hyphens to a file in preparation for formatting and printing by AllWrite, the file then looks very strange when you return to the edit mode. Almost every word has an accent grave mark or two in it! Accent grave is the back-slanted accent, not the regular one. This corresponds to ASCII value 96D or 60H. It can be typed as the SH @ (shifted @ sign).

If you want to remove these from a file to facilitate further editing, use the global search and replace command of AllWrite. To replace all the accents grave with "nothing", place the cursor at the top of the file, then type the command string Break c/SH@//*. Presto, all the accents grave are gone! AllWrite IS powerful!

Grammar checking is interesting, and has its place. One of its good points is that it will point out homonyms (words which sound alike but are spelled differently, such as to, too and two), thus reminding you to be sure you used the correct form. EW can't tell you which to use, just that you are using one of them. It gives examples which help you to decide which word is correct.

It also does such things as reporting when you write a sentence of greater length than its built in style values recommend. Again, it just reminds you the sentence is long; it is up to you to decide if you want to do anything about it. The style limits can be easily changed, so if you tend to be wordy, you can tell EW to accept longer sentences and to stop complaining! In fact, you can turn off any of its checking features (there are over a dozen) as you please.

I have increased the setting on my sentence length, and then usually ignore its squawks when I occasionally exceed the increased limit. If you don't like my style, don't read my stuff!

EW also complains about the use of "passive voice". I never learned about this in high school, and evidently I do it all the time as a result of my learning to write engineering and other reports. Since I feel comfortable with what I write, and most other people seem to also, I also ignore the "passive voice" criticisms.

Another item that I have turned off is the convention that there must be two blank spaces between sentences. I personally object to that old typing rule, as I feel it is not valid. Since all my published stuff gets right justified anyway, the argument is moot because the formatter will fill the line with extra spaces as required.

AVAILABILITY

Electric Webster is still available, either from its publisher, Cornucopia Software, Inc at 1625 Beverly Place, Berkeley, CA 94707 (415-528-7000). The price varies, depending upon the features you select. I think the version with all bells and whistles goes for about $150, but check with the publisher.

They are also still advertising AllWrite (without support) for $65. That's a good deal. An even better one, if you don't have either is AllWrite plus full EW for $200. I wish mine had cost so few dollars when I bought them!

OTHER PRODUCTS

I am certainly no expert on any other word processors and spell checkers, but I do know that LeScript has a spell checker built in. As for its capabilities, I cannot help there. A great advantage of LeScript is that its author, Peter Ray of Anitek still actively sells and supports the program. I understand he is very willing to help you if you get stuck on something, and I am also told that LeScript is very "user friendly" to operate.

TIPS FROM JACK

Will just a little EMI hurt you?

Hardware by Jack Eich

This may be more than a "tip", it is more of a small project, particularly when it comes to hunting for parts (I found mine at swap meets). The objective is to build what is technically a low pass (60 cycle) filter to protect your computer from spikes and surges of voltage from the commercial power lines.

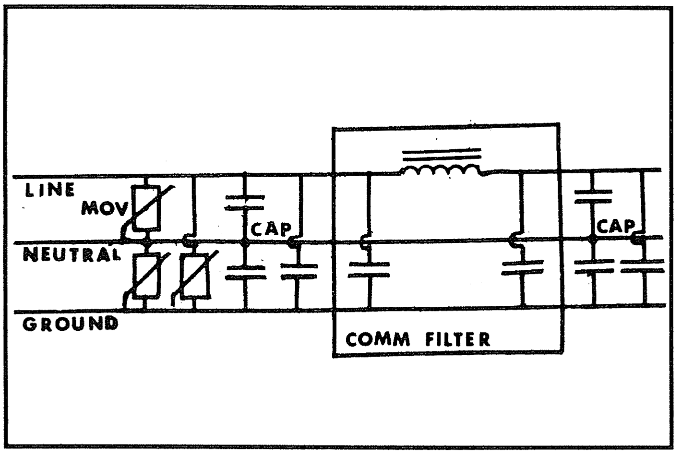
Perhaps I should give my motivating reason to get into the EMI problems. About 4 or 5 years ago I cared for four Model 4 computers in Long Beach - about a 30 or 40 mile drive from my home in Orange, CA. Things were normal with the machines but after about 2 years the customer moved to a large building only about a mile further away. Almost immediately he started having failures. I got mighty tired of those long trips (particularly since I sold him the computers, I didn't charge him for maintenance). So I came up with the filter idea. I installed them and haven't seen the man since! I called him on the phone a couple of months later and he reported no problems. Similar case with a man that had an arc welder next door. Remember your basic science, when the big machine (air conditioner, etc) turns on voltage drops momentarily but when the machine is turned off, - inductance tries to maintain the current flow -so here comes a huge voltage spike - BANG!

First, lets review a plea I made a while back on this same subject: you must have what amounts to true 3 wire 110V single phase power wiring in or at the place you use your computer. This consists of:

1. A ground wire which connects to a water pipe or metal stake in the ground and is also connected to the bottom round hole in the 3 circuit outlets mounted in the walls of your home. This wire is not connected to the power sources to your home or business.

2. The neutral wire which is the "center" wire of the 3 wire 220 volt line that supplies electric power to your home or business. This wire is connected directly to the center taps of all the stepdown transformers (there will be one within a few hundred feet of your home) that reduces the perhaps hundreds of thousands volts at your power source (steam or hydro-electric plant) to the 220 volt 3 wire line to your home. The neutral (center) wire is not connected directly to the ground wire in your home, but is grounded separately at every users location , at every stepdown transformer center tap and probably every power pole along the way. This neutral wire is connected to the upper left flat blade (the wide one) hole in the wall socket.

3. The line wire is the "hot" one. If you are standing on a wet floor or yard, touching the neutral or ground wire will have no effect. Touch any connection to the line wire and you will quickly learn why it is called "hot". This wire is connected to the upper right flat blade (the narrow one) hole in the wall socket.



Attending the above schematic illustration, note that the source of power comes from the left. The three wires encounter a trio of MOVs (metal oxide varistors), with one MOV between each pair of wires. The MOV between the line and neutral is likely the most important part of the filter system. It should be rated at 130 volts (I use 150 volt ones when I can't find the 130's) and 20 amps (fall back to 10A if necessary).This one MOV is the only protection available within almost all $10 to $20 multiple outlet boxes. I've had good luck using 10 amp MOVs in the line to ground and neutral to ground locations (120 or 130 V of course). Available MOVs are made by GE with part number V130LA20A, painted red and about 3/4" diameter and are flat like disc capacitors. Obviously the "130" is the voltage rating and "20" is current. The purpose of MOVs is to clip off spikes above their rated voltage but when a MOV clips spikes it really splatters noise.

Thus far we have covered the protection portion to our line filter, now on to noise removal:

Since most encapsulated noise filters are also made for use in aircraft (which use 400 cycle alternators), and some parts of the world still use 50 cycle AC power (even Los Angeles used to be 50 cycles) (I just exposed my age) the filters are mostly rated 50-400 cycles. That is, they pass all frequencies between 50 and 60 or 400 cycles. I like to pad these filters to a lower frequency cutoff, so I add three 0.1 mfd 600 VDC caps to the input and output sides of the boxed in illustration of a commercial noise suppressor.

No scientific research, I guess I based my selection of 0.1 mfd on the fact that I have seen fairly expensive equipment protected with no more than one 0.1 mfd cap across the power input terminals. If I could find inexpensive 3 mfd 600 VDC (or 150 VAC) caps and they weren't too big physically, I'd use them. In a fairly "bare bones" Model 4, 2 amps seem satisfactory and has the possible advantage of mounting under the key board. If any 'question' use 5 amp filters.

That's my spiel for the added caps to the "comm filter" in the drawing. I should add a few notes about the most available EMI filters:

CORCOM (most common) 50-400 cycle pass band (2 amp, 7/8"h x 1 3/4" x 2"; 5 amp 1 1/8"h x 1 1/8" x 2")

Cornell-Dubilier 5 amp, 50-400 cycle pass band (same size as 5 amp above)

Schaffner (Swiss made) 50 to 60 cycle pass band with schematic diagram on case, 10 amp, 1 1/2"h x 2" x 2" with 4 internal inductors and 4 caps. This one needs no external caps. (Who knows what's in the others)

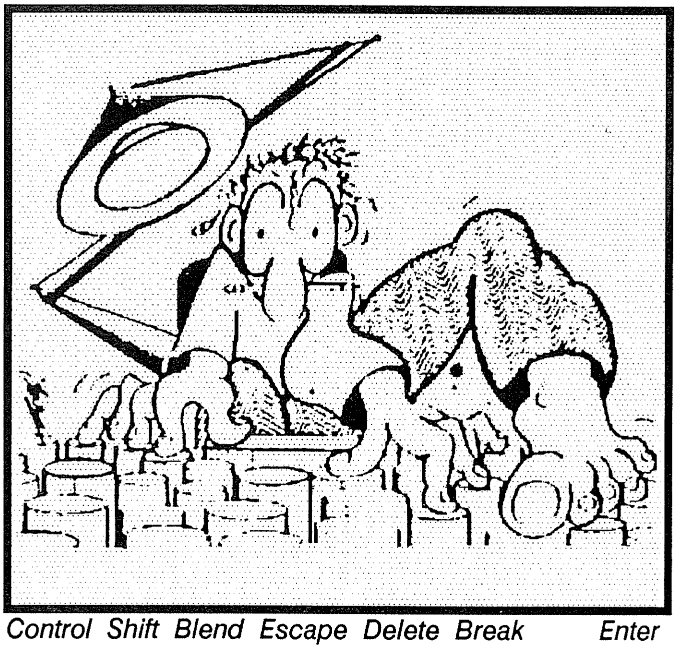
Assembly and Installation

You'll have to find a commercial filter unit with a bandpass of 50 to 60 cycles or to 400 cycles. (Ignore the additional 0.1 mfd caps if you are using 50 to 60 cycle bandpass filters). Attach the 3 MOV's and three of the 0.1 mfd capacitors in parallel with the 3 MOV's to the input side of the commercial filters. Note the schematic drawing. Don't solder yet. One MOV and cap go to the ground lug of the case (or the case itself) and to the line terminals and the MOV and cap from the case to the neutral terminal. There may be no difference between the two inputs, so designate them now. Those two inputs (also two outputs) must be insulated from the case. Connect the third MOV and cap from the line terminal to the neutral terminal. Now decide where you are going to mount the unit. I prefer below the lower drive (D:0). Cut the two wires from the power switch that go to the power supply long enough to make connection with the filter input later. Now we need a piece of terminal strip with 3 isolated terminals, the center one such that the setup can be attached to the computer base near to the commercial filter outputs. Mound the remaining 3 0.1 mfd caps on this strip, 1 between each of the end terminals and the center one and the remaining cap connects the two end terminals.The 2 end terminals connect to the 2 outputs of the filter, which in turn are connected to the power supply. Connect these 2 wires to the power supply such that the 2 colored output wires colors match the position of the 2 inputs. The common centers of the output side of the 2 capacitors needs a 14 gauge (stranded and insulated) piece of wire connected to the neutral connection at the input end of the filter. Continue the same wire and connect to the common ground connection over the cassette where the neutral wire from the power cord is attached to the bracket.

It would be easier if the mounting holes were drilled and tapped before all these things are tied together but only after you have a good idea where things go. Set the boxed commercial filter close to the barrier between the keyboard area and the space under the drives. Leave enough area for the MOVs and caps. Once in place, mark the holes in the mounting lugs and also the terminal strip mounting lug.

With an ice pick indent the center of these holes and drill each all the way through with a size 39 bit. Now center a #6 by 1/2" tapered sheet metal screw and use it to tap the holes driving with a hex nut driver; then remove.

Lastly, assemble the pieces together using #6 x 1/4" screws, and solder all necessary wires. That's it!



Control Shift Blend Escape Delete Break Enter
Space Return Dodge Parry Thrust

Using Your TRS-80 With an HP LaserJet II

Model I/III/4/4P

By Fred Blechman

I make no bones about being "trailing edge" in my computer mentality. I really despise new things, after all the time and trouble it takes to learn the "old" things. So, when I was recently faced with printing out some TRS-80 Model III files on an HP LaserJet II printer, I assumed I would have a problem. Boy, was I ever right!

Trying to blend the old with the new is bound to be a problem with today's technology, where nothing is compatible with anything that has come before - or will come later.

This is not a new problem, since every manufacturer and programmer tries "one-upsmanship" by adding features nobody needs - and dropping all the simple stuff that has been tried and proven.

Let me get to the point. From the first TRS-80 Model Is to the Model 4P, it was standard that the computer, when sending data to a printer, produced a carriage return at the end of every line, to return the printhead to the beginning of the line. However, for whatever reason, no line feed was automatically issued.

Frankly, it looks like Radio Shack screwed-up. With only a carriage return and no line feed, the next printed line simply prints on top of the previous printed line.

To solve this little problem, printers made by Radio Shack automatically added a line feed after every carriage return. Fine - for awhile. But when IBM produced the PC, its operating system automatically issued a line feed after every carriage return at the computer, so the printer didn't have to. Now if you used a TRS-80 with a printer expecting IBM PC input, you would get no line feed, and lines would print on top of each other. Furthermore, if you used an early Radio Shack printer with an IBM PC, you would get DOUBLE SPACING, since the printer was adding an extra line feed!

Printer manufacturers (even Radio Shack) realized that there should be a way to issue a line feed after a carriage return, but only if it was needed. Most printers made in the last five years have a CR/LF toggle switch setting. One setting does nothing, the other setting adds a line feed after every carriage return.

However, Hewlett-Packard, in their infinite wisdom, decided to ignore all past computers, and their LaserJets do not provide a switch to add a line feed after a carriage return! You can't even command this function from the pushbutton control panel on the LaserJet. Bah, humbug!

I discovered this omission recently when trying to print a BASIC LISTing and a word processing file from my TRS-80 Model III to a LaserJet II. Unless I used double line spacing (and I don't know how to get a BASIC LISTing to do this!) I got one line printed on top of another. I couldn't believe that HP had ignored all the many older computers that did not automatically issue a line feed with each carriage return!

I dug into the LaserJet documentation (there are two thick manuals) and found nothing to help. I turned to a 514-page book I had purchased not long ago, "LaserJets Unlimited" (by Ted Nace and Michael Gardner, Peachpit Press.) There, on pages 305 and 444, I found the answer: a special control code that instructs the LaserJet to add a line feed after every carriage return. The code is the ESCAPE character (ASCII 27) followed by the characters &k1G, with a small k and a capital G.

But how do you get this code to the printer? You send it with your computer, using either a short one-line BASIC program, or from your word processor.

The BASIC program is this simple:

```
10 LPRINT CHR$(27);"&k1G"
```

Don't forget the semicolon and the quotes exactly as shown. When you RUN this program, making sure the LaserJet is on-line after self-test and warmup, the printer will act like a TRS-80 expects - a line feed after each carriage return.

If your word processor can issue control codes to the printer, send the same code sequence. You'll also want to set your text and page lengths to 60 lines, which is all a LaserJet can handle. If you don't do this, the LaserJet forces a page at 60 lines and your page numbering can be all out of whack!

With ZORLOF, my favorite TRS-80 word processor (re-reviewed in the next issue) I simply use the following command line at the beginning of the document:

```
C1B266B3147
```

The C indicates a contol code follows, and the rest is equivalent, in hexadecimal, to ESC&k1G. The codes for 60 lines of text and a 60 line page in ZORLOF are E60 and Y60.

It's all so easy - once you find out how......

PDRIVE WITHOUT TEARS

NEWDOS/80 v2 for Model III

By Lance Wolstrup

In the beginning Tandy created the Model I. And the Model I was without form, and working only with a cassette interface; and slowness was upon the cassette interface: and the spirit of Tandy moved upon the face of the ROM.

And Tandy said, Let there be disk drives: and there were disk drives.

And Tandy saw the disk drives, that they were good: and Tandy divided the cassette interface from the disk interface.

And Tandy called the cassette interface Basic, and the disk interface TRSDOS.

This was the evening and the morning of the first day.

In retrospect, Tandy should have spent a lot more time developing their Model I TRSDOS. It was notoriously bad, riddled with errors, and even the quick succession of TRSDOS upgrade releases did not improve matters significantly. To this day, my TRSDOS 2.3 or 2.3B will hang up, or reboot, at the most inconvenient times.

The weakness of TRSDOS became an opportunity for independent programmers. The 'crack team of one' from Apparat provided the first alternative DOS. It was called NEWDOS.

Originally NEWDOS was nothing more,than an extensive series of patches to TRSDOS. But the patches firmed up the weaknesses and made an otherwise intolerable DOS acceptable. The system no longer crashed, and the disk drives now actually worked as promised. The TRS-80 crowd was ecstatic.

While NEWDOS was riding this wave of success, Apparat completely rewrote the Model I DOS, making it better and adding new features. This was called NEWDOS PLUS. The TRS-80 crowd approved, and by now NEWDOS was fast becomming the Model I standard.

Apparat did not rest on its laurels; rather they developed a brandnew DOS called NEWDOS/80. Though still compatible with TRSDOS, it sported features and utilities previously only dreamed about.

Meanwhile Tandy brought out the Model III. It also came with TRSDOS. Tandy seemed to have learned a little something from the failures of the Model I DOS, and by the time TRSDOS was up to version 1.3 it was a reasonable and reliable piece of software. However, it suffered from some of the same inefficiencies as its Model I counterpart: it locked the user in to a particular type of disk drive format.

TRSDOS on the Model I was only capable of handling 35 track, single-sided, single density disks, and Model III TRSDOS could only handle 40 track, single-sided, double density disks. You couldn't 'mix and match' formats. While Model III TRSDOS did have a utility to copy files from Model I TRSDOS, it was simply impossible for the Model I TRSDOS people to copy Model III files back. The DOS'es themselves were not compatible, and since 1.3 was unable to write single density disks, Model I TRSDOS was not capable of reading them.

Apparat had been hard at work. Not only had they upgraded Model I NEWDOS/80 to version 2, but they had also written version 2 to run on the Model III.

The rub with Model III NEWDOS/80 v2 was that it was not compatible with TRSDOS 1.3, nor was it directly compatible with the Model I world.

However, Apparat had anticipated this, as well as realized that there were now a multitude of different disk drives installed or connected to the TRS-80's. Hence they incorporated a DOS command that would allow the user to read and write just about any TRS-80 disk format. This command is probably the single-most feared and misunderstood command in the entire TRS-80 world. It is called PDRIVE.

If you type **PDRIVE,0** <ENTER> from the NEWDOS/80 READY prompt, the screen will display a list of 10 (0-9) PDRIVE settings from the system disk in drive :0. Notice that 0 - 3 have an asterisk placed immediately next to the drive number. Since NEWDOS/80 is only capable of handling 4 disk drives, the asterisk next to 0 - 3 indicate the current setting of this particular drive (if it is hooked up). The remaining PDRIVES (4 - 9) are spares. That is, they are not active, but can be copied to an active drive to change the way that drive will work, should you wish to do so.

If you are looking at the PDRIVE table from a standard Model III (two 40 track, single-sided, double density drives), the first four PDRIVE settings will most likely look this way:

```
0*  TI=A,TD=E,TC=40,SPT=18,TSR=3,GPL=2,DDSL=17,DDGA=2
1*  TI=A,TD=E,TC=40,SPT=18,TSR=3,GPL=2,DDSL=17,DDGA=2
2*  TI=A,TD=E,TC=40,SPT=18,TSR=3,GPL=2,DDSL=17,DDGA=2
3*  TI=A,TD=E,TC=40,SPT=18,TSR=3,GPL=2,DDSL=17,DDGA=2
```

These settings mean that your drive :0 and :1 (and drive :2 and :3 should you add them to your system) will read and write to standard NEWDOS 40 track, single-sided, double density disks.

Each PDRIVE setting has 8 parameters - TI, TD, TC, SPT, TSR, GPL, DDLS, and DDGA. Each of these parameters can be changed individually, or in groups, to create a PDRIVE that will read and write to a different TRS-80 disk format.

TI (type interface) tells DOS which disk interface is to be used.

=A is the standard for both Model I and III.
=B is for Model I with an OMIKRON mapper interface.
=C is for Model I with a PERCOM type interface.
=D is for Model III with an Apparat disk controller.
=E is for Model I with an LNW type interface.
=H is for 8" drives only. Head settle delay.
= I means that sector # start with 1, rather than 0.
=J means that track # start with 1, rather than 0.
=K means that track 0 is formatted in density opposite to that of the diskettes other tracks.
=L allows you to read a 35 or 40 track diskette in an 80 track drive (do not write).
=M indicates that the disk is a standard TRSDOS 1.3 diskette. This is used in combination with A. Setting TI=AM allows transfer of files between NEWDOS/80 and TRSDOS 1.3. Note that flag I is not needed. Also note that only the COPY function will work.

TD (type drive) tells DOS the type of disk the drive will be expected to work with.

=A is 5 inch, single-sided, single density
=B is 8 inch, single-sided, single density
=C is 5 inch, double-sided, single density
=D is 8 inch, double-sided, single density
=E is 5 inch, single-sided, double density
=F is 8 inch, single-sided, double density
=G is 5 inch, double-sided, double density
=H is 8 inch, double-sided, double density

TC (track count) tells DOS how many tracks to work with on the disk. This is normally the total number of tracks on the disk. However, do keep in mind that if TI has been flagged to use one of the weird Model I disks that has track 0 formatted in density opposite that of the rest of the disk, the track count must be set to one less that the total track count.

SPT (sectors per track) tells DOS how many sectors there are on each track.

If TD=A then SPT=10
If TD=B then SPT=17
If TD=C then SPT=20
If TD=D then SPT=34
If TD=E then SPT=18
If TD=F then SPT=26
If TD=G then SPT=36
If TD=H then SPT=52

PDM/SRC

```
00010 ; pdm/src for newdos/80 v2
00020 ; for model iii
00030 ; (c) copyright 1991 by Lance Wolstrup
00040 ; all rights reserved
00050 ;
00060 ;
00070 ;
00080 ;
00090 ;
00100               ;set equates
00110 CLS           EQU         1C9H
00120 CURSOR        EQU         4020H
00130 DISPLY        EQU         4467H
00140 DOSCAL        EQU         4419H
00150 DSP           EQU         33H
00160               ;clear screen and display name of program
00161               ;and the copyright
00170               ORG         7000H
00180 START         CALL        CLS
00190               LD          HL,15361
00200               PUSH        HL
00210               LD          DE,MSG1
00220               CALL        CHROUT
00230               POP         HL
00240               PUSH        DE
00250               LD          DE,64
00260               ADD         HL,DE
00270               POP         DE
00280               PUSH        HL
00290               CALL        CHROUT
00300               POP         HL
00310               PUSH        DE
00320               LD          DE,64
00330               ADD         HL,DE
00340               POP         DE
00350               CALL        CHROUT
00360               LD          HL,15390
00370               LD          DE,MSG2
00380               CALL        CHROUT
00390               LD          HL,15454
00400               CALL        CHROUT
00410               LD          HL,15518
00420               CALL        CHROUT
00421               ;draw line to seperate name & copyright
00422               ;from the working portion of the screen
00430               LD          HL,15552
00440               PUSH        HL
00450               POP         DE
00460               INC         DE
00470               LD          BC,63
00480               LD          (HL),131
00490               LDIR
00491               ;display left portion of menu
00500 START1        LD          B,6
00510               LD          HL,15688
00520               LD          DE,MENU
```

TSR (track stepping rate) tells the disk controller how fast to move from track to track.

TSR = 0 steps at 5ms
TSR = 1 steps at 10 ms
TSR = 2 steps at 20 ms
TSR = 3 steps at 40ms

GPL (granules per lump) is the way NEWDOS/80 handles granules. Other Model I and III DOSes use the words 'track' or 'cylinder' rather than 'lump'. Model I uses 2 granules of 5 sectors each per lump (track or cylinder). Model III normally uses 3 granules of 6 sectors each per track or cylinder. NEWDOS/80 allows you to alter the number of granules assigned to a lump from 2 all the way up to 8. Using a low number gives you more disk space, but less space for directory entries. Using a high number gives less disk space, but room for more directory entries.

DDSL (default directory starting lump) tells DOS where to find the first sector of the directory. Standard for both Model I and III is 17.

DDGA (default directory granule allocation) tells DOS how many granules to allocate to the directory during format. Like GPL, this is a number from 2 to 8. The smaller the number, the more data space will be available, but less space for directory entries, and vice versa.

As mentioned earlier, these parameters can be changed at will. For example, if you installed a 40 track, double-sided, double density drive as drive :2, you could take advantage of its capabilities by changing the PDRIVE settings for drive :2 with the following command:

PDRIVE,0,2,TD = G,SPT = 36 < ENTER >

It would be necessary to reboot before the new PDRIVE setting would take effect. However, if you do not wish to reboot, you can add a parameter to the end of the command to tell DOS to use the new setting immediately. This parameter is A.

PDRIVE,0,2,TD = G,SPT = 36,A < ENTER >

If the setting is correct (does not produce an incompatible PDRIVE setting), it will take immediate effect.

There are many more nuances and 'gotchas' built into the PDRIVE command, and I'll be the first to tell you that I always have the manual next to me when I try to do something exotic. Since I normally only use NEWDOS/80 on my Model 4s (in Model III mode), and I have 40 track, double-sided, double density drives, I have written a program that will take most of the drudgery and confusion out of switching between disk formats.

The program is called PDM (PDrive Menu). It was written in assembly language, using the EDTASM that is included with the NEWDOS/80 v2 DOS disk.

```
00530 MENLP1  CALL    CHROUT
00540         DJNZ    MENLP1
00541         ;display right portion of menu
00550         LD      B,6
00560         LD      HL,15718
00570 MENLP2  CALL    CHROUT
00580         DJNZ    MENLP2
00581         ;display last two menu entries
00590         LD      B,2
00600         LD      HL,16087
00610 MENLP3  CALL    CHROUT
00620         DJNZ    MENLP3
00621         ;display 'drive type' prompt
00630 ASK1    LD      HL,16276
00640         LD      (CURSOR),HL
00650         LD      HL,ASKTYP
00660         CALL    DISPLY
00670         LD      A,14
00680         CALL    DSP
00681         ;wait for keypress
00690         CALL    49H
00700         PUSH    AF
00710         LD      A,15
00720         CALL    DSP
00721         ;translate keypress to upper case
00722         ;and display it
00730         POP     AF
00740         AND     223
00750         CALL    DSP
00751         ;is it X (exit to newdos) - jump if X
00760         CP      'X'
00770         JP      Z,EXIT
00771         ;check for keypresses A through M
00772         ;and jump to appropriate routines if so
00780         CP      65
00790         JR      Z,GETA
00800         CP      66
00810         JR      Z,GETB
00820         CP      67
00830         JR      Z,GETC
00840         CP      68
00850         JR      Z,GETD
00860         CP      69
00870         JR      Z,GETE
00880         CP      70
00890         JR      Z,GETF
00900         CP      71
00910         JR      Z,GETG
00920         CP      72
00930         JR      Z,GETH
00940         CP      73
00950         JR      Z,GETI
00960         CP      74
00970         JR      Z,GETJ
00980         CP      75
00990         JR      Z,GETK
01000         CP      76
01010         JR      Z,GETL
```

PDM is a simple menu of the various PDRIVE settings that I have had occasion to use over the years. Having the PDRIVEs in a menu has saved me an enormous amount of time. not having to struggle with the manual each time I need to read or write a different disk format. So, type in the program listing and assemble it to a CMD file. I think you will find it as handy as I have.

When the program is executed it presents a menu with 14 options. The screen will look like this:

```
PDM   PDRIVE MENU for NEWDOS/80 v2
      (c) Copyright 1991 Lance Wolstrup
      All rights reserved
_____________________________________________

A. 1S SD - 35 TRK            G. 1S DD 35 TRK
B. 2S SD - 35 TRK            H. 2S DD 35 TRK
C. 1S SD - 40 TRK             I.  1S DD 40 TRK
D. 2S SD - 40 TRK             J.  2S DD 40 TRK
E. 1S SD - 80 TRK            K. 1S DD 80 TRK
F. 2S SD - 80 TRK            L. 2S DD 80 TRK
                M. TRSDOS 1.3
                X. EXIT TO NEWDOS

         SELECT DRIVE TYPE (A-M) _
```

As can be observed from the above, all combinations of 35, 40 and 80 track formats are covered, along with the oddball TRSDOS 1.3. Absent from the menu are the eight inch drive formats, as well as the exotic Model I PDRIVE settings that handle double density, but with the boot sector formatted in single density. I never use them, so they are left out.

To make a selection, simply type the letter associated with the desired format. Pressing X, exits the program and returns to NEWDOS. Any other valid choice, brings up another prompt directly underneath the 'select drive' prompt:

```
         SELECT PDRIVE SLOT (1-9) _
```

Answer this prompt with the number of the drive to be the target of the new PDRIVE setting. Pressing <ENTER> returns the program to the 'select drive' prompt. Any other valid selection changes the appropriate PDRIVE setting on the system disk in drive :0, and then displays the entire system PDRIVE table for inspection. Pressing any key at this point brings the program back to the menu.

Do keep in mind that you will be able to change any and all PDRIVE slots except slot 0. However, if you choose a disk format that is beyond the capabilities of the disk controller (such as selecting the 80 track format with a 40 track drive, or selecting double density with a single density drive), you will not be successful. PDM will change the PDRIVE software settings, but, obviously, it cannot make modifications to the hardware.

```
01020           CP          77
01030           JR          Z,GETM
01031           ;keypress was not A through M, so go
01032           ;back and get another keypress
01040           JR          ASK1
01050           ;A was pressed - move command to
01051           ;buffer and go get drive prompt
01060 GETA      LD          HL,TYPEA
01070           CALL        MOVTYP
01080           JR          GETDRV
01081           ;B was pressed - move command to
0182            ;buffer and go get drive prompt
01090 GETB      LD          HL,TYPEB
01100           CALL        MOVTYP
01110           JR          GETDRV
01111           ;C was pressed - move command to
01112           ;buffer and go get drive prompt
01120 GETC      LD          HL,TYPEC
01130           CALL        MOVTYP
01140           JR          GETDRV
01141           ;D was pressed - move command to
01142           ;buffer and go get drive prompt
01150 GETD      LD          HL,TYPED
01160           CALL        MOVTYP
01170           JR          GETDRV
01171           ;E was pressed - move command to
01172           ;buffer and go get drive prompt
01180 GETE      LD          HL,TYPEE
01190           CALL        MOVTYP
01200           JR          GETDRV
01201           ;F was pressed - move command to
01202           ;buffer and go get drive prompt
01210 GETF      LD          HL,TYPEF
01220           CALL        MOVTYP
01230           JR          GETDRV
01231           ;G was pressed - move command to
01232           ;buffer and go get drive prompt
01240 GETG      LD          HL,TYPEG
01250           CALL        MOVTYP
01260           JR          GETDRV
01261           ;H was pressed - move command to
01262           ;buffer and go get drivempt
01270 GETH      LD          HL,TYPEH
01280           CALL        MOVTYP
01290           JR          GETDRV
01291           ;I was pressed - move command to
01292           ;buffer and go get drive prompt
01300 GETI      LD          HL,TYPEI
01310           CALL        MOVTYP
01320           JR          GETDRV
01321           ;J was pressed - move command to
01322           ;buffer and go get drive prompt
01330 GETJ      LD          HL,TYPEJ
01340           CALL        MOVTYP
01350           JR          GETDRV
01351           ;K was pressed - move command to
01352           ;buffer and go get drive prompt
01360 GETK      LD          HL,TYPEK
```

Before ending this article, I have to give credit to an unknown programmer from Holland for creating the font used to draw the letters PDM. Peter Plomp submitted a disk with a program called AGCAP. This program will be featured as soon as I can plow through the code. It is a clever utility that will automatically figure out what the PDRIVE setting should be for a particular drive. Anyway, the programmer used the font to draw the letters AGCAP. I was impressed with the style, so I shamelessly stole the letter P, modified C to make D, and finally made M from two modified As. Thank you, unknown programmer.

```
01370          CALL    MOVTYP
01380          JR      GETDRV
01381          ;L was pressed - move command to
01382          ;buffer and go get drive prompt
01390 GETL     LD      HL,TYPEL
01400          CALL    MOVTYP
01410          JR      GETDRV
01411          ;M was pressed - move command to
01412          ;buffer and go get drive prompt
01420 GETM     LD      HL,TYPEM
01430          CALL    MOVTYP
01440 ;
01441          ;display drive number prompt
01450 GETDRV   LD      HL,16340
01460          LD      (CURSOR),HL
01470          LD      HL,ASKDRV
01480          CALL    DISPLY
01490          LD      A,14
01500          CALL    DSP
01501          ;wait for keypress
01510          CALL    49H
01520          PUSH    AF
01530          LD      A,15
01540          CALL    DSP
01550          POP     AF
01551          ;if <enter> was pressed - go back
01552          ;to the drive type prompt
01560          CP      13
01570          JP      Z,ASK1
01571          ;check if keypress is in range og 0-9,
01572          ;jump if not
01580          CP      31H
01590          JR      C,GETDRV
01600          CP      3AH
01610          JR      NC,GETDRV
01611          ;keypress is in range, so store it
01612          ;in buffer
01620          LD      (DRVBUF),A
01630 ;
01640          ;erase menu
01650          LD      HL,15616
01660          PUSH    HL
01670          LD      (CURSOR),HL
01680          LD      A,31
01690          CALL    DSP
01691          ;perform PDRIVE command from buffer
01700          LD      HL,PDBUF
01710          CALL    DOSCAL
01711          ;display 'press any key' prompt
01720          LD      HL,16345
01730          LD      (CURSOR),HL
01740          LD      HL,ANYKEY
01750          CALL    DISPLY
01760          LD      A,14
01770          CALL    DSP
01771          ;wait for key press
01780          CALL    49H
01781          ;turn off cursor
01790          LD      A,15
01800          CALL    DSP
01801          ;erase PDRIVE table from screen and
01802          ;display menu
01810          POP     HL
01820          LD      (CURSOR),HL
01830          LD      A,31
01840          CALL    DSP
01850          JP      START1
01860 ;
01870          ;exit routine - clear screen and
01871          ;return to newdos
01880 EXIT     CALL    CLS
01890          RET
01900 ;
01901          ;subroutines
01902 ;
01903          ;chrout takes the bytes pointed to by DE
01904          ;and stuffs it in the screen location pointed
01905          ;to by HL. The terminating byte is 0
01910 CHROUT   PUSH    HL
01920 CHROU1   LD      A,(DE)
01930          OR      A
01940          JR      Z,CHROU2
01950          LD      (HL),A
01960          INC     DE
01970          INC     HL
01980          JR      CHROU1
01990 CHROU2   INC     DE
02000          POP     HL
02010          PUSH    DE
02020          LD      DE,64
02030          ADD     HL,DE
02040          POP     DE
02050          RET
02060 ;
02061          ;movtyp moves the command pointed
02062          ;to by HL into the buffer pointed to
02063          ;by DE. The terminating byte is 0
02070 MOVTYP   LD      DE,TYPBUF
02080 MOVE1    LD      A,(HL)
02090          OR      A
02100          RET     Z
02110          LD      (DE),A
02120          INC     DE
```

```
02130           INC      HL
02140           JR       MOVE1
02150           ;the actual PDRIVE commands:
02152           ;A was selected
02160 TYPEA     DEFM     'TI=A,TD=A,TC=35,SPT
=10,TSR=3,GPL=2,'
02170           DEFM     'DDSL=17,DDGA=2,A'
02180           DEFB     13
02190           DEFB     0
02191 ;
02192           ;B was selected
02200 TYPEB     DEFM     'TI=A,TD=C,TC=35,SPT
=20,TSR=3,GPL=2,'
02210           DEFM     'DDSL=17,DDGA=2,A'
02220           DEFB     13
02230           DEFB     0
02231 ;
02232           ;C was selected
02240 TYPEC     DEFM     'TI=A,TD=A,TC=40,SPT
=10,TSR=3,GPL=2,'
02250           DEFM     'DDSL=17,DDGA=2,A'
02260           DEFB     13
02270           DEFB     0
02271 ;
02272           ;D was selected
02280 TYPED     DEFM     'TI=A,TD=C,TC=40,SPT
=20,TSR=3,GPL=2,'
02290           DEFM     'DDSL=17,DDGA=2,A'
02300           DEFB     13
02310           DEFB     0
02311 ;
02312           ;E was selected
02320 TYPEE     DEFM     'TI=A,TD=A,TC=80,SPT
=10,TSR=2,GPL=6,'
02330           DEFM     'DDSL=20,DDGA=6,A'
02340           DEFB     13
02350           DEFB     0
02351 ;
02352           ;F was selected
02360 TYPEF     DEFM     'TI=A,TD=C,TC=80,SPT
=20,TSR=2,GPL=6,'
02370           DEFM     'DDSL=20,DDGA=6,A'
02380           DEFB     13
02390           DEFB     0
02391 ;
02392           ;G was selected
02400 TYPEG     DEFM     'TI=A,TD=E,TC=35,SPT
=18,TSR=3,GPL=2,'
02410           DEFM     'DDSL=17,DDGA=2,A'
02420           DEFB     13
02430           DEFB     0
02431 ;
02432           ;H was selected
02440 TYPEH     DEFM     'TI=A,TD=G,TC=35,SPT
=36,TSR=3,GPL=2,'
02450           DEFM     'DDSL=17,DDGA=2,A'
02460           DEFB     13
02470           DEFB     0
```

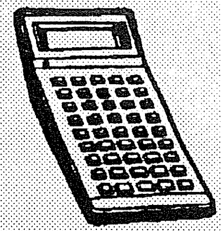
```
02471           ;I was selected
02480 TYPEI     DEFM     'TI=A,TD=E,TC=40,SPT
=18,TSR=3,GPL=2,'
02490           DEFM     'DDSL=17,DDGA=2,A'
02500           DEFB     13
02510           DEFB     0
02511 ;
02512           ;J was selected
02520 TYPEJ     DEFM     'TI=A,TD=G,TC=40,SPT
=36,TSR=3,GPL=2,'
02530           DEFM     'DDSL=17,DDGA=2,A'
02540           DEFB     13
02550           DEFB     0
02551 ;
02552           ;K was selected
02560 TYPEK     DEFM     'TI=A,TD=E,TC=80,SPT
=18,TSR=2,GPL=8,'
02570           DEFM     'DDSL=35,DDGA=6,A'
02580           DEFB     13
02590           DEFB     0
02591 ;
02592           ;L was selected
02600 TYPEL     DEFM     'TI=A,TD=G,TC=80,SPT
=36,TSR=2,GPL=8,'
02610           DEFM     'DDSL=35,DDGA=6,A'
02620           DEFB     13
02630           DEFB     0
02631 ;
02632           ;M was selected
02640 TYPEM     DEFM     'TI=AM,TD=E,TC=40,SPT
=18,TSR=3,GPL=6,'
02650           DEFM     'DDSL=17,DDGA=2,A'
02660           DEFB     13
02670           DEFB     0
02680 ;
02681           ;name and copyright. PDM, is drawn
02682           ;over three screen lines, thus making it
02683           ;necessary for each character to have a
02684           ;top, middle, and a bottom
02585 ;
02686           ;top part of P
02690 MSG1      DEFB     137
02700           DEFB     147
02710           DEFB     191
02720           DEFB     131
02730           DEFB     139
02740           DEFB     155
02750           DEFB     180
02760           DEFB     32
02770 ;
02771           ;top part of D
02780           DEFB     137
02790           DEFB     147
02800           DEFB     191
02810           DEFB     183
02820           DEFB     139
02830           DEFB     155
02840           DEFB     189
```

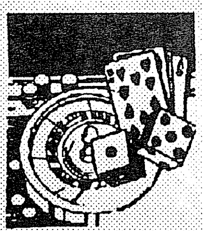
```
02850           DEFB     32
02860 ;
02861           ;top part of M
02870           DEFB     32
02880           DEFB     152
02890           DEFB     185
02900           DEFB     171
02910           DEFB     155
02920           DEFB     180
02930           DEFB     184
02940           DEFB     167
02950           DEFB     151
02960           DEFB     182
02970           DEFB     164
02980           DEFB     0
02990 ;
03000           ;middle part of P
03010           DEFB     32
03020           DEFB     149
03030           DEFB     191
03040           DEFB     140
03050           DEFB     142
03060           DEFB     142
03070           DEFB     129
03080           DEFB     32
03090 ;
03091           ;middle part of D
03100           DEFB     32
03110           DEFB     149
03120           DEFB     191
03130           DEFB     32
03140           DEFB     32
03150           DEFB     149
03160           DEFB     191
03170           DEFB     32
03180 ;
03181           ;middle part of M
03190           DEFB     170
03200           DEFB     170
03210           DEFB     191
03220           DEFB     32
03230           DEFB     137
03240           DEFB     155
03250           DEFB     167
03260           DEFB     134
03270           DEFB     32
03280           DEFB     191
03290           DEFB     149
03300           DEFB     149
03310           DEFB     0
03320 ;
03330 ;bottom part of P
03340           DEFB     160
03350           DEFB     165
03360           DEFB     191
03370           DEFB     144
03380           DEFB     32
03390           DEFB     32
```

```
03400           DEFB     32
03410           DEFB     32
03420 ;
03421           ;bottom part of D
03430           DEFB     160
03440           DEFB     165
03450           DEFB     191
03460           DEFB     183
03470           DEFB     184
03480           DEFB     185
03490           DEFB     159
03500           DEFB     32
03510 ;
03511           ;bottom part of M
03520           DEFB     154
03530           DEFB     186
03540           DEFB     149
03550           DEFB     32
03560           DEFB     32
03570           DEFB     32
03580           DEFB     32
03590           DEFB     32
03600           DEFB     32
03610           DEFB     170
03620           DEFB     181
03630           DEFB     165
03640           DEFB     0
03650 ;
03651           ;the rest of the name and copyright text
03660 MSG2      DEFM     'PDRIVE MENU for
NEWDOS/80 v2'
03670           DEFB     0
03680 MSG3      DEFM     '(c) Copyright 1991 Lance
 Wolstrup'
03690           DEFB     0
03700 MSG4      DEFM     'All rights reserved'
03710           DEFB     0
03711 ;
03712           ;the PDRIVE menu
03720 MENU      DEFM     'A. 1S SD - 35 TRK'
03730           DEFB     0
03740           DEFM     'B. 2S SD - 35 TRK'
03750           DEFB     0
03760           DEFM     'C. 1S SD - 40 TRK'
03770           DEFB     0
03780           DEFM     'D. 2S SD - 40 TRK'
03790           DEFB     0
03800           DEFM     'E. 1S SD - 80 TRK'
03810           DEFB     0
03820           DEFM     'F. 2S SD - 80 TRK'
03830           DEFB     0
03840           DEFM     'G. 1S DD - 35 TRK'
03850           DEFB     0
03860           DEFM     'H. 2S DD - 35 TRK'
03870           DEFB     0
03880           DEFM     'I. 1S DD - 40 TRK'
03890           DEFB     0
03900           DEFM     'J. 2S DD - 40 TRK'
```

```
03910           DEFB      0
03920           DEFM      'K. 1S DD - 80 TRK'
03930           DEFB      0
03940           DEFM      'L. 2S DD - 80 TRK'
03950           DEFB      0
03960           DEFM      'M. TRSDOS 1.3'
03970           DEFB      0
03980           DEFM      'X. EXIT TO NEWDOS'
03990           DEFB      0
03991 ;
03992           ;drive type prompt
04000 ASKTYP    DEFB      31
04010           DEFM      'SELECT DRIVE '
04011           DEFM      'TYPE (A-M) '
04020           DEFB      3
04021 ;
04022           ;drive number prompt
04030 ASKDRV    DEFB      30
04040           DEFM      'SELECT PDRIVE '
04041           DEFM      'SLOT (1-9) '
04050           DEFB      3
04051 ;
04052           ;prompt to press any key
04060 ANYKEY    DEFM      'PRESS ANY KEY '
04070           DEFB      3
04080 ;
04081           ;buffers to hold PDRIVE commands
04090 PDBUF     DEFM      'PDRIVE,0,'
04100 DRVBUF    DEFB      1
04110           DEFB      ','
04120 TYPBUF    DEFS      64
04130 ;
04140           END       START
```

"BUT IT WAS SO FRIENDLY
IN THE STORE!"





ALL ABOUT ALLWRITE pt. 3
(well - almost)

By Dr. Allen W. Jacobs

When I write an article, I often find it helpful to outline my thoughts in an informal manner. While I don't manipulate numbered and lettered headings around, although this can be done, I do like to have some consistent and convenient method of subordinating one thought to another. Usually, these outlines do not exceed the limits of memory, even if the final article does. Since space in the file is not at a premium, I would rather sacrifice space for the benefit of clarity. Thus was born my outlining technique, implemented largely in soft keys.

We all know that Allwrite is not a dedicated outline editor, but it does have enough "stuff" to do the job. I was stumped on one feature I needed until I worked out an undocumented method to bring the technique "to life". From inspection of the soft key file that appeared near the end of part 2 and its name, OUTLINE/DEF, it is apparent that my outlining technique involves the liberal use of indents. That is how I maintain subject subordination.

The technique is relatively straightforward. Main headings are at the left margin. Subordinate headings are indented 5/10 inch. More subordinate headings are indented yet another 5/10 inch. I do this by using relative indents. When I return to a more major heading, I vacillate between indenting negative 5/10 inches or, for greater clarity, I turn indenting off and then indent as many relative half (5/10) inch increments as necessary to properly subordinate the next heading. This can be done during the writing of the outline or afterwards since all the soft keys I use share the easily insertable structure I have previously described. The technique is wordy but clear, even in an unprinted document. While the character volume used to attain this type of formatting can be relatively large, the number of keystrokes used to achieve it is rather small, due to the use of softkeys.

My outlines lacked one more feature that I needed for clarity. There was no means of indenting subsequent lines in a heading without separating the second and all lines thereafter by hand and indenting them farther to the right. I used to do this. It might seem to look more "wysiwyg" and use less space in a file but this is not true. Remember that spaces at the beginning of a line use up as many characters in a file as does an equal amount of text. Knowing that, it remains a matter of your personal preference.

The only other method I know of is to use the offset command. In normal text, after issuance, all lines after the first are indented until the next text break. In normal text, the offset command works well. However, as described, the relative offset command had problems with repetitive use. If a relative offset is issued such as +2/10 inch, it works correctly the first time. The second time it is issued, it is cumulative and thus the text will be offset 4/10 inch. The third time, the subsequent text lines will be offset by 6/10 inch, and so on. If an absolute offset command is issued, such as 2/10 inch, the text will be offset 2/10 inch from the left margin, independent of the indent. Issuing an offset "0" or "off" command and a relative +2/10 inch offset each time an offset is used is almost as ungainly as making hand indents.

In reading on other commands, I decided to try an undocumented offset command. It is offset +0. I discerned that placing a sign ("+" or "-") in front of a command tells the formatter that it is about to receive a relative rather than an absolute value. The numerical value tells the formatter by what amount to increase the offset. In this case, the offset is "increased" by ZERO. It's undocumented but it works. To use it, issue an offset of +2/10, or whatever amount you like immediately after an indent command. The second and subsequent times, issue an offset of +0. After issuing indent off and offset off commands, reissue an offset of +2 just once and an offset of +0 thereafter.

The best way to learn about all of this is to try it out and please let all of us know if there is a better way of doing any of it. Issue some of these soft key commands on a copy of something you wrote or make up a test text. You will not hurt a thing. Remember, becoming familiar with the actual key sequences in this article is not nearly as important as becoming more familiar with the process of constructing and using soft keys and experimenting in general.

It appears that there are two ways in which block operations are handled by word processors. Since I don't know the proper names for each of the two techniques I have encountered or whether these two techniques even HAVE official names, I will use what I believe to be differentiating terms for these schemes. Let us refer to them as "named" or "PERSISTENT" block operations versus "IMMEDIATE" block operations. If you have dabbled in any number of word processors or text editors, you know what I am talking about.

To do any block operation, you have to define the block or blocks you want to work with. The major difference is whether the defined blocks persist after the operation or whether the block markers "go away". There are advantages and disadvantages to both schemes.

In a PERSISTENT block operation scheme, a block is defined at its start, at its end, and it is named. Named blocks can be moved, copied, deleted, exchanged, printed, sent to another file, hyphenated, spell checked, "thesaurused", placed in upper or lower case, imbedded, made into a graphical image, mathematically manipulated, sorted, archived, etc. Any or all of these operations may be available to act on a block, in a given word processor. After the block is defined, none of these operations needs to be done immediately, since the block identifier and block markings PERSIST, until they are removed by the user. Good old Scripsit is an example of a named block operation scheme.

In an IMMEDIATE block operation scheme, a block is defined at its start, and at its end. This places the processor into a block operation mode wherein the next action of the editor will operate directly on the block defined at that time. Once the block has been acted upon, all markings are removed and the processor returns to its default text input mode. You will notice that the block did not have to be named and it didn't have to be unmarked after the operation. That's because we're only dealing with one block at a time. Allwrite works in this manner.

If you are doing a lot of input of original text and you want to do fast efficient block operations, immediate blocks are the way to go. It is much more convenient than having to search through text to unmark a block after you no longer need it, which can be irritating.

On the other hand, if you are "boilerplating" multiple blocks of text, it becomes difficult to repeatedly re-mark a block of text, in order to copy or insert it into a number of places. Fortunately, Allwrite has single line block operation commands which I use all the time. Were it not for these commands, the search and replace commands would still be of limited use for short blocks. The problem is that the use of search and replace is limited to the length of the search/replace arguments. If the block is longer, the operation becomes harder to perform, but not impossible. You can always "get" a predefined block of text from a file. It will even work with an entire file, but it takes a lot of keystrokes. Well, admittedly, so does my solution; but since those keystrokes are in soft keys, that cuts their number to a manageable amount. The most advantageous feature is that the entire scheme can be carried out within a single file, in memory.

This persistent block technique takes advantage of a number of features of the editor that it is helpful to understand. While it does not look or work like Scripsit, I acknowledge that the block marking scheme I am using was inspired by good ol' Scripsit. Since this is an alternate way of manipulating blocks, I have only implemented it essentially for block copies. It seems to me that most other block operations are better performed through the use of immediate blocks. While block exchanges can only be done with persistent blocks, I have not bothered to implement this feature because I never found much use for it. If you previously used Scripsit and you don't use it now, do you miss this feature? Do you remember it? I didn't think so, but if that feature is a "must have", use the following techniques to develop it yourself. There will be room in the BLOCK/DEF file we will create.

Are you familiar with named points in Allwrite?...No?..I was afraid of that.... Here is a brief explanation.

Hitting: <CLEAR><,><_>, where <_> is any letter from A-Z, defines the point in text your cursor is currently at.

Later, when you hit: <CLEAR><.><_>, where <_> is any of the letters from A-Z you have previously defined, you are instantly returned to the previously defined location, providing you have not inserted or deleted text before the marked point. If you have, the point will "drift" by the amount of text inserted or deleted, on a line by line basis.

You can always redefine your point with another: <CLEAR><,><_>, thereby reassigning that named point to the curfent location. All named point references are lost when the editor is exited. Play around with this feature if you are not familiar with it.

Since named points are apparently just stored in a simple lookup table of line numbers and positions within the line, we will use the feature by not depending on its accuracy for any time longer than the instant before it is to be referenced. We can do that because the actual locations of the block we want to act upon can be located with Allwrite's "Find" command. The way we find it is to give the beginning of each named block a unique signature. That is, a character sequence that is not likely to be found any other place in the file. Since the <CLEAR><=> soft key is "hard programmed" to generate a hard space (ASCII "127"), we can always depend on it being there. Also, it is unlikely that there is a frequent need for the hard space character to be repeated three successive times, in most text. Placing a non-hard-space character in the sequence of three should give us about as unique a block start identification signature as we will need: <HARD SPACE><HARD SPACE><_><HARD SPACE>, where <_> is any of the letters from A-Z you want to use to identify a given block. Now we can have a "NAMED" blocks feature to go along with named points.

If all blocks end with the same signature, we should not have a uniqueness problem if we have already established the beginning of each unique block, with the exception that blocks can not be nested. If that is too much of a restriction for you, you can always go to a different labeling scheme. I don't think that it is generally worth it. If you do, the technique I describe here may help serve as a basis for you to develop your own system.

With these naming conventions "established", let's outline the soft keys we will need to manage our NAMED or PERSISTENT block system.

We will need soft keys to:

- 1. Identify the starting point of a block and to name it.
- 2. Identify the ending point of a block.
- 3. Find the current location of a uniquely named block.
- 4. Load or initialize the current starting and ending locations of the uniquely named block into the named point table.
- 5. Initialize the point in text to place the copied block.
- 6. Act upon the block (most likely copy it).
- 7. Un-mark it.
- 8. Delete it.
- 9. Put it into some other permanent file (ie. save it).

Correspondingly, I have modified my OUTLINE/DEF and OUTLINE/TMP files to form the files, BLOCK/DEF AND BLOCK/TMP. If you want to do it another way, go ahead. It is just a matter of deciding what keys you want where. It's almost like designing your own keyboard. Look out Mr. Dvorak!

My BLOCK/TMP file thus far looks like this:

```
;cm Start the paper at the very top of the page, due to
the title.
;pl 80
;li 8
;ls 6
;pi 13
;fo off
strtbloc ! = ;sk
2 = endbloc      "= RU ALF/CMD
3 = findbloc     # = ;cm
4 = initbloc     $= unmrkbloc
5 = blk2here     % = deletbloc
6 = copybloc     & = putbloc
7 = copy2end     ' = ;fo o
8 = VA           ( = del spc
9 = STATUS       ) = del spc +
0 = SR           * = capslock
: = chr nmbr     * = print scrn
- = ____          = = hard space
x = ins spc      y = split line
z = del line
```

I should just interject at this point that increasing the page length as I did is just the lazy way of eliminating the top and bottom margin defaults with cut sheets. It may not work well with continuous feed paper as the next sheet will start too far down the page.

The named block management soft keys are presented below:

Soft Key 1: <CLEAR><O><CLEAR><=> <CLEAR><I><CLEAR><=><SPACE> <CLEAR><=><LEFT ARROW> <LEFT ARROW><CLEAR><I><F1><Q>

Soft Key 2: <CLEAR><O><CLEAR><=> <CLEAR><I><CLEAR><=><CLEAR> <=><CLEAR><I><F1><Q>

Soft Key 3: <CLEAR><T><BREAK><F> <SPACE><CLEAR><=><CLEAR><=> <SPACE><CLEAR><=><LEFT ARROW> <LEFT ARROW><F1><Q>

Soft Key 4: <SPACE><SPACE><SPACE> <SPACE><CLEAR><,><Y><BREAK> <SPACE><CLEAR><=><CLEAR> <=><CLEAR><=><ENTER> <LEFT ARROW><CLEAR><,><Z><ENTER>

Soft Key 5: <CLEAR><,><X><CLEAR><3> <F1><Q>

Soft Key 6: <CLEAR><.><Y><CLEAR><B> <CLEAR><.><Z><CLEAR><C<CLEAR> <.><X><CLEAR><H><F1><Q>

Soft Key 7: <CLEAR><C><CLEAR><E> <CLEAR><H><F1><Q>

Soft Key 8: UNCHANGED

Soft Key 9: UNCHANGED

Soft Key 0: <BREAK><S><R><*><ENTER> <F1><Q>

Soft Key !: <CLEAR><ENTER><;><s><k> <ENTER><F1><Q>

Soft Key ": UNCHANGED

Soft Key #: <CLEAR><O><;><CLEAR><I> <c><m><SPACE><CLEAR><I<ENTER> <F1><Q>

Soft Key $: <BREAK> <R><SPACE><ENTER> <BREAK><F><SPACE><CLEAR><CLEAR> <=><CLEAR><=><ENTER> < CLEAR><R><F1><Q>

Soft Key %: <CLEAR><B><BREAK><F> <SPACE><CLEAR><=><CLEAR><=> <CLEAR><=><ENTER><RIGHTARROW> <RIGHT ARROW><CLEAR><D><F1><Q>

Soft Key &: <CLEAR><.><Y><CLEAR><B> <CLEAR><Z><CLEAR><P><F1><Q>

Soft Key ': <CLEAR><ENTER><;><f><o> <SPACE><o><F1><Q>

Soft Key (: <CLEAR><@><CLEAR><ENTER> <CLEAR><SPACEBAR><CLEAR><J> <LEFT ARROW><DOWN ARROW><F1><Q>

Soft Key): <CLEAR><@><CLEAR><ENTER> <CLEAR><SPACEBAR><CLEAR><J> <LEFT ARROW><+><LEFT ARROW> <DOWN ARROW><F1><Q>

An explanation of how these keys are used is in order:

Soft Key 1 puts a persistent beginning-of-block marker into the text. It allows you to name the block with a letter from A to Z, 0 to 9, or virtually any single character other than a hard space (ASCII "127"). Be careful of control characters (anything below ASCII "32"). Just about any printable character should be acceptable but avoiding truly special characters is just plain prudent. If you are going to use upper and lower case identifiers, make sure that your search is case sensitive. The default is case insensitive. Avoiding a hard space prevents the block initializing soft key from identifying the beginning of this block as the end of a previous block. Avoiding the overlaying or nesting of blocks is imperative, without us changing the block identification scheme.

To use this soft key, just put the cursor over the beginning of the block you want to name, activate the soft key and just type in the single character block name.

Soft Key 2 puts a persistent end-of-block marker into the text. It is the same marker for all blocks. That is why blocks can not be nested. The end of any block will be defined by the first end-of-block marker encountered. If you fail to place it, you will get erroneous results from a named block action. To use it, just place the cursor over the first character or space to the right of the last character you desire in the block and activate the soft key.

Soft Key 3 locates the block you desire to act upon. Because Allwrite's "Find" command is being used to search from the top of text, activating this key appears to disrupt the entire file. Don't let this worry you. All you have to do is to activate the key, place the single character block designator into the command line, and press <ENTER>. The block, if previously marked and named, will appear.

Soft Key 4 is used in conjunction with Soft Key 3 in that it initializes just the contents of the block (ie. not the markers). Soft Key 4 happens to be 21 keystrokes long. Since the "end-of-soft-key-sequence" is 2 characters long, I had to pad the soft key with one benign keystroke in order to cause the 22 character sequence length to automatically end the soft key. Thus, I selected the <ENTER> key. That's why its presence doesn't seem to make sense.

Soft Key 5 locates the destination of the block to be copied. Just put the cursor at the exact location to which you wish to copy a named, initialized, and located block. You may do this before or after you find and initialize the named block to be copied. Soft Keys 3, 4, and 5 should be used in close conjunction with each other to prevent their locations from getting corrupted. Soft Key 5 and 6 were originally combined. However, on a suggestion from Roy Beck, it is more convenient to separate their functions. This allows you to specify the block to be copied AFTER you select its destination, or before, at your option. That flexibility is worth the cost of another soft key.

Soft Key 6 actually performs the named block copy. Using it looks like an earthquake has struck your computer. But, when the dust settles, your block should have been placed into the location you selected. Keep this soft key, as all others, out of the reach of very young children.

The next three soft keys are for the maintenance of persistent blocks. What they do is to save them, un-mark them, and delete them.

Soft Key $ removes the persistent markings around a named block that has first been found and initialized, by using Soft Keys 3 and 4. This key preserves the text in place.

Soft Key % removes both the text and the block markings of the specified block, after it has been found and initialized, by using Soft Keys 3 and 4. The block must be properly marked, most importantly at its end. If it is not, a great amount of text can be lost. This is especially true if the first of two named blocks is at the beginning of a file, lacking an end-of-block marker, and the second block is properly terminated, at the end of the file. If you deleted what you thought was the first block, you could theoretically lose the entire file. That is, of course, a worst-case condition.

There is a remedy for it. Just keep most of your blocks near the beginning of a file and make sure that they are terminated properly. This practice also provides an additional benefit. If persistent blocks are placed near the beginning of the file and you are going to insert the same block into a number of locations, you don't have to re-initialize it each time. Don't place the block at the first character position in the file, however, as the find routine will not be able to locate it. Just put a non printing ";cm" or some other character before the block marker.

The lack of need to reinitialize the block holds true as long as you haven't inserted any text before its occurrence. Of course, this also assumes that you haven't used any search and replace commands, initialized other blocks, or other such thing in the meantime. Make sure that you are comfortable with the system before trying out this technique.

Finally, Soft Key & allows you to nondestructively save a located and initialized block to a separate file on disk. Just use Soft Keys 3 and 4. Then "put" your block to a file as you would if it was an immediate block. It will still be in its original position in your text and still marked. The copy in the disk file will not be marked. You can change this condition to a marked copy if you desire. I will leave that up to you after you have seen how I "programmed" this key, as it is. If you are having any problem, you can get in touch with me through TRSTimes.

Some More About Allwrite

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

Allen Jacobs' series "All About Allwrite" began with the confession that it really wouldn't be ALL about Allwrite because the author had never used Allwrite with a laser printer. I have used Allwrite with such a machine -- actually, a DeskJet Plus printer, which uses the same HP LaserJet driver and prints at 300 dots per inch. George Madison said something about setting up Allwrite with one of these printers a few issues ago; what better way to describe what Allwrite can do with a laser printer than to provide you with an actual printout? Here it is...

As should be obvious, Allwrite will allow you to use any number of typefaces in the same document (though, as should be equally obvious from this example, a large mix of faces is not very esthetic). Two features make this possible. One is the printer driver, which can include support for up to 64 different fonts and can be customized for whatever combination you want. The other feature is the way Allwrite handles line margins. It measures lines in terms of tenths of an inch, not in characters. In many word processors, changing the pitch changes the margins: a line of 60 characters in pica type will be longer than the same 60 characters in elite type. *If you are mixing fonts and want all the margins to line up, every time you change type you have to manually change the margin settings until, through a trial-and-error process, you get them even. This generally means you can't have more than one typeface on the same line.* *Allwrite, however, automatically adjusts the margins for whatever size type you're using, even within lines. You simply type your text, switching fonts wherever you wish, and then print out the result.* Though <u>TRSTimes</u> is printed on an IBM-compatible machine. most of it could just as easily be printed with Allwrite and a TRS-80. LaserJet and DeskJet drivers for Allwrite are currently available through <u>Computer News 80</u>, as are laser drivers for a number of other TRS-80 word processors.

And now you really do know ALL about Allwrite...

UNASSIGNED MEMORY BOARD

Hardware by Skip May

This memory board, when connected to your TRS-80 through the expansion slot, will provide you with free memory to use. Let me explain further.

If you examine the TRS-80 memory map, you will notice a portion of memory between 3000 and 37D0 hex that is not used. Possibly it is saved for expansion of hardware, but more likely it is left over for software "ROM" reasons. What this board does is to provide a usable read/write RAM for you to store a variety of software projects.

One might also address and "burn" an Eprom for reading a dedicated program. I've used this "porthole" for various hardware projects, including I/O for my external hardware.

The beauty of this memory is that no current or former software talks to this address as it was "never there". The originator of the idea, my friend Jim, uses it to store the disk directory and a quick patch to his DOS, makes a directory "dir" without disk access and a considerable time savings. I use it for rewriting programs which before crashed my DOS, requiring reboot. This board causes no problems with the normal hard drive operation, although a "Y" cable is required for installation. An auto function to load and execute programs placed here are quick and efficient. The possibilities are endless.

About the board:

There are only three chips required. I use a small PC board from Radio Shack and older 22ga. telephone wire for "point to point" wiring. There is an option for a Nicad Battery backup, this would require an occasional powerup to charge the battery if the supply line was tied to the internal 5 volt Supply. You may also just leave it out but you will lose the data as you power down.

The pin numbers are shown for a model I, but the line functions are shown so you able to cross them to the equivalent model III/4 pins.

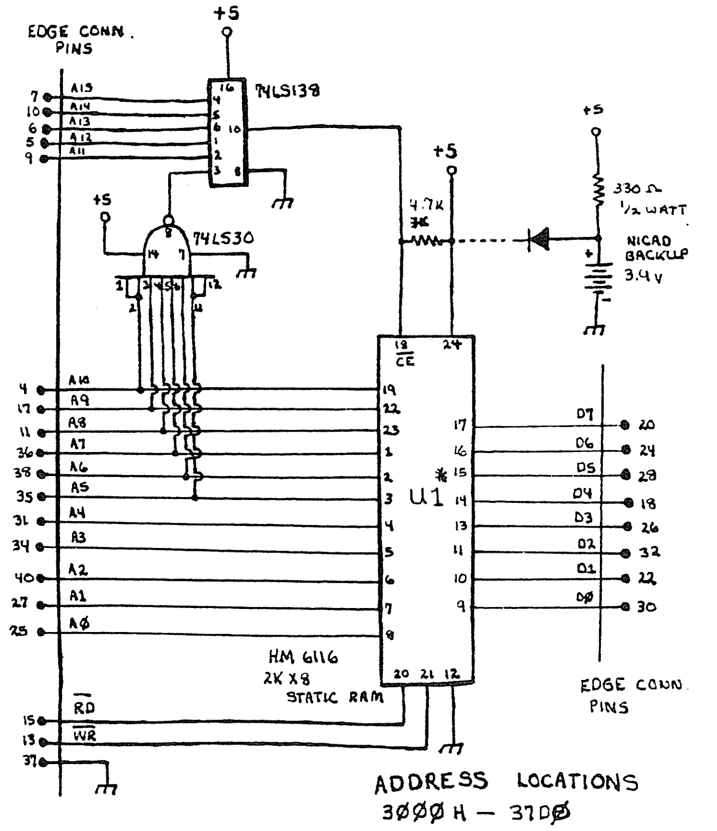
I have not examined the model 4 memory map, so before you build this project you may wish to look for available memory which may be at different locations.

For any of the machines it is practical to have a service or technical manual. U1 is a Harris 2K X 8 static ram available at most serious electronic part houses. Other types may be used with comparable functions taking pin and supply voltages into consideration.

The 74LS30 and 74LS138 are used as address decoders and are tied to the chip enable (active low). The whole assembly is placed on the small PC board which has the connector fixed at one end. Unless you have other items on your expansion bus port the unit hangs right off the side and down if so built.

Once the board is built, check it for solder bridges and problems as best you can, then install it on your expansion port with the power off. Power the computer up which should operate normally; if not, immediately power the computer down, remove the board and check your wiring. A good board will allow you to poke a value into memory and peek it back. From that point each person may use it for their own requirement. I would enjoy hearing from anyone who builds this project or others based on this idea. Best of luck.

TRS-80 MODEL I UNASSIGNED MEMORY BOARD

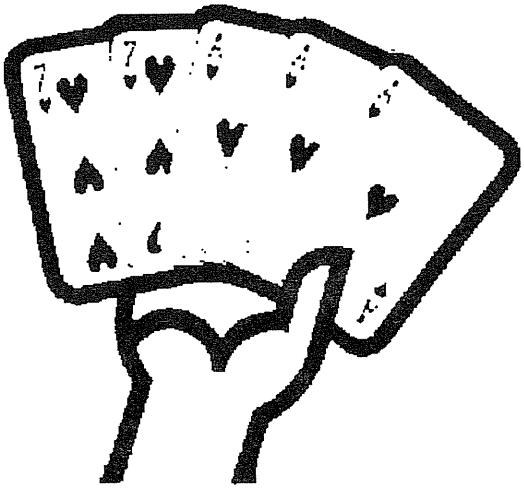


* U1 may be configured as RAM or ROM - drawn by Skip May

Recreational & Educational Computing

Mind-Reading and other Recreations

By Michael W. Ecker, Ph.D.



The 21 Cards

This trick is an oldie, so let me describe this briefly.

The BASIC program displays three columns of numbers from 1 to 21. Think of one such number. The program will ask you three times which column houses it. At the end, the program reveals your chosen card!

```
10 CLS:PRINT"The 21 Cards - Program by Sally Frazza"
20 PRINT"adapted by Dr. M. Ecker for REC"
100 DEFINT A-Z
110 DIM V(21),X(7,3)
120 ITER=0
130 FOR N=1 TO 21:V(N)=N:NEXT N
140 PRINT:PRINT"Pick a card, please...":PRINT
150 IF ITER=3 THEN PRINT:
PRINT"Your card is ";V(11):END
160 ITER=ITER+1
170 N=0
180 FOR I=1 TO 7:FOR J=1 TO 3
190 N=N+1
200 X(I,J)=V(N)
210 NEXT J: NEXT I
220 PRINT: PRINT
221 FOR I=1 TO 7
222 PRINT USING "## "; X(I,1); X(I,2); X(I,3)
223 NEXT I
230 PRINT
240 INPUT "Column of card (1, 2, or 3)"; C
250 IF C<1 OR C>3 THEN 240
260 FOR K=1 TO 3: O(K)=K: NEXT K
270 IF C< >2 THEN TEMP=O(C):
O(C)=O(2): O(2)=TEMP: 'SWAP
290 N=0
300 FOR K=1 TO 3: FOR I=1 TO 7
310 J=O(K)
320 N=N+1
330 V(N)=X(I,J)
340 NEXT I: NEXT K
350 GOTO 150
```

Mystery Program

It's a little late for New Year's, and maybe too early for Easter/Passover, but what the heck. This program is good any time of year, I figure.

```
10 CLS: PRINT: PRINT
20 FOR A=1 TO 2
30 FOR B=1 TO 4
40 X = 2 - ABS(SGN(B-3))
50 FOR C=1 TO X
60 PRINT CHR$(84-7*A+5*B-8*X);
70 NEXT: NEXT
80 PRINT CHR$(A+31);: NEXT
90 PRINT: END
```

(Thanks to Bob Lodge of Minnesota for this one.)

TRS-80 Resources: Casino 21, Loan Amortization

Everybody outside our small group says that the TRS-80 is dead, that users don't buy, and all the rest. Maybe true; maybe not.

But, I actually received one - count it: one - honest-to-goodness inquiry right around New Year's Day about a loan amortization program for the TRS-80. Can any reader supply any names of products in this category still available other than the one I have?

The only one I know about is Recreational Mathemagical Software's Fastloan 2.5. Okay, I won't snow you. It's my program. I took advantage of my math background, wrote the first version of the program years ago, upgrade it every year or so, and I still think it's pretty neat. It's $24.95

for TRS-80 version, $29.95 for PC version, with free shipping. Its basic function is to calculate monthly payments for mortgages, automobiles, etc., given amount of loan (principal), interest rate, and period of loan. Output is to screen, printer, or both, as you choose. It also handles the inverse question of the maximum amount you can afford to borrow based on amount you can afford as a monthly payment.

The TRS-80 version comes on a self-booting disk using a licensed DOS. For orders (or more information if you send SASE), contact Recreational Mathemagical Software, 909 Violet Terrace, Clarks Summit, PA 18411. (Phone (717) 586-2784.

In a more playful spirit, there is Casino 21. No, I did not write it, but I wish I were clever enough to, because it plays a nice game of blackjack. Features include multiple players, or just one, against the computer; random deal of cards, hitting, standing, doubling, splitting, insurance, and betting.

This, too, is available exclusively from Recreational Mathemagical Software, same address. CASINO 21, written expressly and solely for the TRS-80, is available for TRS-80 Model 3 (tape or disk), Model 1 (cassette tape), Model 4, 4P, 4D (disk format, Model III mode). All disk versions come complete on a self-booting disk with a licensed DOS that is TRSDOS 1.3-compatible.

Fair warning: Though there are no commands to memorize, special keys to press, or special computer things to know, you should know the basics of the game of 21 itself (blackjack). It's $15.95 plus $1 shipping/handling.

Incidentally, in fairness to anybody else left in this market, I used to know of one other commercial program. It was for Model 4 only, but it was a great one by Eric Husted. Unfortunately, I've lost touch with him. Can any reader enlighten me as to any other offerings available or Eric's whereabouts?

That's it for this issue. If you wish to contact me, please write to:

Dr. Michael W. Ecker
Recreational & Educational Computing
909 Violet Terrace
Clarks Summit, PA 18411

Dr. Michael W. Ecker is a Penn State University math professor as well as a computer writer-reviewer and columnist with 300 publication credits. Mike is also Editor/Publisher or Recreational & Educational Computing.

REC, from which "The 21 Cards" and "Mystery Program" pieces have been adapted, is in its sixth year and is available for $27 per calendar-year of 8 issues, prepaid. It focuses on "mathemagic" and computer recreations. readers are invited to try a trial subscription of three issues for $10, fully creditable toward subscription.

ATTENTION TRSDOS 1.3. USERS!

ANNOUNCING "SYSTEM 1.5.", THE MOST COMPREHENSIVE 1.3. UPGRADE EVER OFFERED!

MORE SPEED!! MORE POWER!! MORE PUNCH!!

While maintaining 100% compatibility to TRSDOS 1.3., this DOS upgrade advances TRSDOS 1.3. into the 90's!
SYSTEM 1.5. supports 16k-32k bank data storage and 4MGHZ clock speed (4/4P/4D).

DOUBLE SIDED DRIVES ARE NOW 100% UTILIZED! (all models).

CONFIG=Y/N	CREATES CONFIG BOOT UP FILE	DATE=Y/N	DATE BOOT UP PROMPT ON or OFF
TIME=Y/N	TIME BOOT UP PROMPT ON or OFF	CURSOR='XX'	DEFINE BOOT UP CURSOR CHAR
BLINK=Y/N	SET CURSOR BOOT UP DEFAULT	CAPS=Y/N	SET KEY CAPS BOOT UP DEFAULT
LINE='XX'	SET *PR LINES BOOT UP DEFAULT	WP=d.Y/N (WP)	WRITE PROTECT ANY or ALL DRIVES
ALIVE=Y/N	GRAPHIC MONITOR ON or OFF	TRACE=Y/N	TURN SP MONITOR ON or OFF
TRON=Y/N	ADD an IMPROVED TRON	MEMORY=Y/N	BASIC FREE MEMORY DISPLAY MONITOR
TYPE=B/H/Y/N	HIGH/BANK TYPE AHEAD ON or OFF	FAST	4 MGHZ SPEED (MODEL 4'S)
SLOW	2 MGHZ SPEED (MODEL III'S)	BASIC2	ENTER ROM BASIC (NON-DISK)
CPY (parm,parm)	COPY/LIST/CAT LDOS TYPE DISKS	SYSRES=H/B/'XX'	MOVE/SYS OVERLAY(s) TO HI/BANK MEM
SYSRES=Y/N	DISABLE/ENABLE SYSRES OPTION	MACRO	DEFINE ANY KEY TO MACRO
SPOOL=H/B.SIZE	SPOOL is HIGH or BANK MEMORY	SPOOL=D.SIZE='XX'	LINK MEM SPOOLING TO DISK FILE
SPOOL=N	TEMPORARILY DISABLE SPOOLER	SPOOL=Y	REACTIVATE DISABLED SPOOLER
SPOOL=RESET	RESET (NIL) SPOOL BUFFER	SPOOL=OPEN	OPENS, REACTIVATES DISK SPOOLING
SPOOL=CLOSE	CLOSES SPOOL DISK FILE	FILTER *PR.ADLF=Y/N	ADD LINE FEEDS BEFORE PRINTING 0DH
FILTER *PR.IGLF	IGNORES 'EXTRA' LINE FEEDS	FILTER *PR.HARD=Y/N	SEND 0CH to PRINTER (FASTEST TOF)
FILTER *PR.FILTER	ADDS 256 BYTE PRINTER FILTER	FILTER *PR.ORIG	TRANSLATE PRINTER BYTE TO CHNG
FILTER *PR.FIND	TRANSLATE PRINTER BYTE TO CHNG	FILTER *PR.RESET	RESET PRINTER FILTER TABLE
FILTER *PR.LINES	DEFINE NUMBER LINES PER PAGE	FILTER *PR.WIDTH	DEFINE PRINTER LINE WIDTH
FILTER *PR.TMARG	ADDS TOP MARGIN to PRINTOUTS	FILTER *PR.BMARG	ADDS BOTTOM MARGIN to PRINTOUT
FILTER *PR.PAGE	NUMBER PAGES, SET PAGE NUMBER	FILTER *PR.ROUTE	SETS PRINTER ROUTING ON or OFF
FILTER *PR.TOF	MOVES PAPER TO TOP OF FORM	FILTER *PR.NEWPG	SET DCB LINE COUNT TO 1
FILTER *KI.ECHO	ECHO KEYS to the PRINTER	FILTER *KI.MACRO	TURN MACRO KEYS ON or OFF
ATTRIB:d.PASSWORD	CHANGE MASTER PASSWORD	DEVICE	DISPLAYS CURRENT CONFIG INFO

All parms above are installed using the new LIBRARY command SYSTEM (parm,parm). Other new LIB options include DBSIDE (enables double sided drive by treating the "other side" as a new independent drive, drives 0-7 supported) and SWAP (swap drive code table #s). Dump (CONFIG) all current high and/or bank memory data/routines and other current config to a disk data file. If your type ahead is active, you can (optional) store text in the type buffer, which is saved. During a boot, the config file is loaded back into high/bank memory and interrupts are recognized. After executing any active auto command, any stored type ahead data will be output. FANTASTIC! Convert your QWERTY keyboard to a DVORAK! Route printer output to the screen or your RS-232. Macro any key,even F1, F2 or F3. Load *01-*15 overlay(s) into high/bank memory for a memory only DOS! Enter data faster with the 256 byte type ahead option. Run 4MGHZ error free as clock, disk I/O routines are properly corrected! Spool printing to high/bank memory. Link spooling to disk (spooling updates DCB upon entering storage). Install up to 4 different debugging monitors. Print MS-DOS text files, ignoring those unwanted line feeds. Copy, Lprint, List or CATalog DOSPLUS, LS-DOS, LDOS or TRSDOS 6.x.x. files and disks. Add top/bottom margins and/or page numbers to your hard copy. Rename/Redate disks. Use special printer codes eg: LPRINT CHR$(3); toggles printer output to the ROUTE device. Special keyboard codes add even more versatility. This upgrade improves date file stamping MM/DD/YY instead of just MM/YY. Adds optional verify on/off formatting, enables users to examine *01-*15, DIR, and BOOT sectors using DEBUG, and corrects all known TRSDOS 1.3. DOS errors. Upgrade includes LIBDVR, a /CMD driver that enables LIBRARY commands, such as DIR, COPY, DEBUG, FREE, PURGE, or even small /CMD programs to be used within a running Basic program, without variable or data loss.

By special arrangement with GRL Software,
SYSTEM 1.5. is now distributed exclusively by TRSTimes magazine.

ORDER YOUR COPY TODAY!

Send $39.95 (U.S. funds) to:

TRSTimes - SYSTEM 1.5.
5721 Topanga Canyon Blvd., Suite 4
Woodland Hills, CA. 91367

TRS-80 PUBLIC DOMAIN SOFTWARE BONANZA

We have bought collections of software from people leaving the TRS-80 world. As fast as we can, we are weeding out the good Public Domain and Shareware from the Commercial programs and the junk. So far, we have come up with 28 disks for the Model I & III, and 10 disks for the Model 4.

Model I & III

PD#1: binclock/cmd, binclock/doc, checker/bas, checker/doc, chomper/bas, cls/cmd, dduty3/cmd, driver/cmd, driver/doc, drivtime/cmd, mazeswp/bas, minibase/bas, minitest/dat, mx/cmd, piazza/bas, spdup/cmd, spdwn/cmd, vici/bas, vid80/cmd, words/dic.

PD#2: creator/bas, editor/cmd, maze3d/cmd, miner/cmd, note/cmd, poker/bas, psycho/cmd, supdraw/cmd, vader/cmd

PD#3: d/cmd, trsvoice/cmd, xmodem/cmd, xt3/cmd, xt3/txt, xthelp/dat

PD#4: cobra/cmd, disklog/cmd, flight/bas, flight/doc, narzabur/bas, narzabur/dat, narzabur/his, narzabur/txt, othello/bas, vid80x24/cmd, vid80x24/txt

PD#5: eliza/cmd, lu31/cmd, sq31/cmd, usq31/cmd

PD#6: clawdos/cmd, clawdos/doc, cocoxf40/cmd, dskrnam/bas, menu/cmd, ripper3/bas, sky2/bas, sky2/his, space/cmd, stocks/bas, trs13pat/bas, vidsheet/bas

PD#7: cards/bas, cities/bas, coder/bas, eye/bas, heataudt/bas, hicalc/bas, life/bas, moustrap/bas, ohare/bas, slots/bas, stars/cmd, tapedit/bas

PD#8: craps/bas, fighter/bas, float/bas, hangman/bas, jewels/cmd, lifespan/bas, varidump/bas, xindex/bas, xor/bas

PD#9: bublsort/bas, chess/bas, finratio/bas, homebudg/bas, inflat/bas, mathdril/bas, midway/bas, nitefly/bas, pokrpete/bas, teaser/bas

PD#10: ltc21/bas, ltc21/ins, lynched/bas, match/bas, math/bas, message/bas, message/ins, portfol/bas, portfol/ins, spellegg/bas, storybld/bas

PD#11: alpha/bas, caterpil/cmd, cointoss/bas, crolon/bas, cube/cmd, dragon/cmd, fastgraf/bas, fastgraf/ins, lunarexp/bas, music/bas, music/ins, planets/bas, volcano/cmd

PD#12: baccarat/bas, backpack/bas, backpack/ins, doodle/bas, dragons/bas, dragons/ins, king/bas, sinewave/bas, snoopy/bas, wallst/bas, wallst/ins

PD#13: atomtabl/bas, boa/bas, chekbook/bas, conquer/cmd, dominos/bas, morse/bas, mountain/bas, quiz/bas, signbord/bas, sketcher/bas

PD#14: autoscan/bas, checkers/bas, craps/bas, ducks/bas, isleadv/bas, nim/bas, rtriangl/bas, sammy/cmd, typing/bas, wordpuzl/bas

PD#15: budget/bas, corp/bas, corp/ins, fourcolr/bas, fullback/bas, grapher/bas, illusion/bas, jukebox/bas, ledger/bas, maze/cmd, reactest/bas, shpspree/bas, states/bas, tapecntr/bas, tiar/bas, tiar/ins

PD#16: amchase/bas, constell/bas, filemastr/bas, foneword/bas, geometry/bas, heartalk/bas, hidnumbr/bas, lgame/bas, marvello/bas, powers/bas, scramble/bas, speed/bas, subs/bas

PD#17: conundrm/bas, eclipse/bas, esp/bas, esp/ins, hustle/bas, jacklant/bas, mindblow/bas, othello/bas, pleng/bas, rubik/bas, trend/bas, ufo/bas, veggies/bas

PD#18: backgam/bas, chess/cmd, cosmip/cmd, distance/bas, hexpawn/bas, music/cmd, stokpage/bas, texted/bas, texted/ins, trex/bas, twodates/bas, wanderer/bas

PD#19: banner/bas, cresta/cmd, lander/bas, medical/bas, moons/bas, par/bas, parchut/bas, pillbox/bas, readtrn/bas, replace/bas, ship/cmd, solomadv/bas, space/cmd, survival/bas

PD#20: bomber/bas, bumbee/cmd, ciaadv/bas, dice31/bas, dice31/ins, diskcat1/bas, firesafe/bas, flashcrd/bas, hitnmiss/bas, mazegen/bas, mazescap/cmd, roulette/bas, seasonal/bas

PD#21: aprfool/bas, catmouse/bas, d/cmd, escape/bas, header/bas, kalah/bas, mathwrld/bas, nameit/bas, note/cmd, photo/bas, read/cmd, syzygy/bas, timeshar/cmd, timeshar/doc, trace80/cmd, trsdir/cmd, worm/bas, yatz80/bas

PD#22: arcade/bas, cube/cmd, eclipse/bas, lcd/bas, leastsqr/bas, medical/bas, million/bas, pwrplant/bas, round/bas, subway/bas, tapeid/bas

PD#23: artil/bas, artil/ins, baseconv/bas, crushman/bas, dissert/bas, huntpeck/bas, jungle/bas, jungle/ins, messages/bas, monitor/bas, monster/bas, moons/bas, ohmlaw/bas, stockpage/bas, tictacto/bas

PD#24: baslist/asm, baslist/cmd, baslist/doc, cleaner3/cmd, cleaner3/doc, difkit1/bas, difkit1/doc, dirpatch/asm, dirpatch/cmd, e/cmd, ei/doc, i/cmd, newmap/bas, newmap/doc, varlst/asm, varlst/cmd, varlst/doc

PD#25: copy/bas, copy/doc, dirpw/asm, dirpw/cmd, dirpw/doc, dskfmt/bas, dskfmt/doc, himap/asm, himap/cmd, huricane/bas, hv/bas, hv/doc, keydemo/bas, keyin/bas, keyin/doc, lazyptch/asm, lazyptch/doc, salvage/bas, salvage/doc,wpflt/asm, wpflt/flt

PD#26: constell/bas, divisor/bas, frame/bas, heatfus/bas, heatfus/doc, hicalc/bas, mathlprt/bas, mathquiz/bas, molecule/bas, morscode/bas, phyalpha/bas, phyalpha/doc, remaindr/bas, usa/bas, wiring/bas

PD#27: engine/bas, fraction/bas, geosat/bas, grades/bas, julian/bas, lunarcal/bas, mailist/bas, metaboli/bas,musictrn/bas, perindex/bas, potrack/bas

PD#28: chainfil/bas, citoset/bas, convnum/bas, cursors/bas, cursors/doc, datamkr/bas, deprec/bas, gmenuii/bas, ledger12/bas, menui/bas, menuii/bas, minives/bas, ninteres/bas, refinanc/bas, regdepo/bas, rembal/bas, rndbordr/bas

Model 4

M4GOODIES#1: day/cmd, day/txt, gomuku/cmd, llife/cmd, llife/doc, writer/cmd, writer/doc, writer/hlp, yahtzee/bas

M4GOODIES#2: arc4/cmd, arc4/doc, cia/bas, etimer/cmd, index/cmd, index/dat, mail/bas, mail/txt, trscat/cmd, trscat/txt, util4/cmd, xt4/cmd, xt4/dat, xt4hlp/dat

M4GOODIES#3: convbase/bas, dates/bas, dctdsp/cmd, dmu/cmd, dmu/doc, dskcat5/cmd, dskcat5/doc, editor/cmd, editor/doc, fedit/cmd, fkey/asm, fkey/cmd, fkey/doc, hangman/cmd, m/cmd, m/src, membrane/bas, miniop2/cmd, miniop2/src, move/cmd, move/doc, othello4/bas, scroll4/cmd, scroll4/src, setdate6/cmd, setdate6/doc, setdate6/fix, spaceadv/bas, taxman/bas, utilbill/bas, utilbill/doc

M4GOODIES#4: word wizard - disk 1

M4GOODIES#5: word wizard - disk 2

M4GOODIES#6: word wizard - disk 3

M4GOODIES#7: calendar/cmd, castladv/bas, civilwar/bas, crimeadv/bas, dctdsp/cmd, ed6/cmd, ed6/doc, edittext/bas, fedit/cmd, mail/bas, mail/txt, scramble/bas, states/bas, textpro/cmd, time4/bas, wizard/bas, wizard/doc, worldcap/bas

M4GOODIES#8: books/bas, books/doc, dmu/cmd, dmu/doc, hamcalc/bas, hamhelp/bas, network/bas, network/doc, pirate/bas, pirate/doc, vmap/bas, vmap/doc, vmap2/bas, vmap2/doc, zork1/doc, zork2/doc, zork3/doc

M4GOODIES#9: ft/cmd, ft/doc, pterm/cmd, pterm/doc, r/cmd, r/doc, scrconv/bas, scrconv/doc, video4/asm, video4/cmd

M4GOODIES#10: checker/cmd, crossref/cmd, crossref/doc, ddir/cmd, diskcat/cmd, diskcat/doc, division/bas, division/doc, getput/bas, getput/doc, host/cmd, hv/bas, hv/doc, maszap4/cmd, maszap4/doc, park/cmd, profile4/doc, protect/bas, protect/doc, rename/bas, replace/bas, restore/bas, rm/bas, scrndump/bas, scrndump/doc, super/hlp, vers/cmd

Each disk is $5.00 (U.S.)
or get any 3 disks for $12.00 (U.S.)
please specify the exact disks wanted.

TRSTimes PD-DISKS
5721 Topanga Canyon Blvd., Suite 4
Woodland Hills, CA. 91367

PRICE LIST effective March 1, 1991

Prices subject to change without notice

TRS-80 Software

Product Nomenclature	Mod III	Mod 4	Price S&H
AFM: Auto File Manager data base	P-50-310	n/a	$49.95 D
BackRest for hard drives	P-12-244	P-12-244	$34.95
BASIC/S Compiler System	P-20-010	n/a	$29.95 B
BSORT / BSORT4	L-32-200	L-32-210	$14.95
CON80Z / PRO-CON80Z.	M-30-033	M-31-033	$19.95
diskDISK / LS-diskDISK	L-35-211	L-35-212	$29.95
DISK NOTES from TMQ (per issue)			$10.00
DoubleDuty		M-02-231	$49.95
DSM51 / DSM4	L-35-204	L-35-205	$49.95
DSMBLR / PRO-DUCE	M-30-053	M-31-053	$29.95
EDAS / PRO-CREATE	M-20-082	M-21-082	$44.95 D
EnhComp / PRO-EnhComp	M-20-072	M-21-072	$59.95 D
Filters: Combined I & II	L-32-053	n/a	$19.95
GO:Maintenance	n/a	M-33-100	$49.95 B
GO:System Enhancement	n/a	M-33-200	$49.95 B
GO:Utility	n/a	M-33-300	$49.95 B
Hardware Interface Kit	n/a	M-12-110	$24.95
HartFORTH/PRO-HartFORTH	M-20-071	M-21-071	$49.95 B
LB Maintenance Utility-M4	n/a	L-50-515	$19.95
LDOS 5.1.4 User Manual	L-40-020	n/a	$15.00 D
LDOS 5.3 Mod3 Upgrade Kit	M-10-033	same	$34.95
LED / LS-LED	L-30-020	L-30-021	$19.95
Little Brother-M4 (Ver 1.0)	n/a	L-50-510	$74.95 D
LS-DOS 6.3.1 Upgrade Kit - M4	n/a	M-11-043	$39.95
LS-DOS 6.3.1 Diskette - M4	n/a	M-11-243	$15.00
LS-DOS 6.3.1 Upgrade kit - M2/12/16		M-11-002	$39.95 B
LS-Host/Term	n/a	L-35-281	$39.95
LS-UTILITY	n/a	L-32-150	$24.95
MC / PRO-MC	M-20-064	M-21-064	$79.95 D
Mister ED	n/a	M-51-028	$39.95 B
MRAS / PRO-MRAS	M-20-083	M-21-083	$59.95 D
PowerDot (Epson or Tandy)	P-32-21?	n/a	$19.95
PowerDraw	P-32-220	n/a	$19.95
PowerDriver Plus (Epson).	P-50-200	P-50-200	$17.95
PowerMail Plus	P-50-003	P-50-004	$39.95 D
PowerMail Plus TextMerge	P-50-100	P-50-100	$15.00
PowerScript	P-50-142	P-50-142	$24.95
PRO-WAM	n/a	M-51-025	$74.95 D
PRO-WAM Toolkit	n/a	M-51-225	$29.95
Programmer's Guide DOS 6.	n/a	M-60-060	$20.00 B
QuizMaster	L-51-500	n/a	$19.95
RATFOR-M4		M-21-073	$59.95 D
RSHARD - R/S HD driver	M-12-013	same	$29.95
ST80-III	P-35-300	n/a	$39.95
SuperUtilityPlus	P-32-132	P-32-104	$44.95 D
SuperUtilityPlus CMD file diskette	P-32-832	P-32-804	$20.00
Supreme HD Driver (PowerSoft-RS)	P-12-113	P-12-113	$34.95
TBA / LS-TBA	L-21-010	L-21-011	$19.95 D
THE SOURCE 3-Volume Set	n/a	L-60-020	$40.00 D
Toolbox/Toolbelt	P-32-203	P-32-245	$24.95
UNREL-T80	same	M-30-054	$29.95
UTILITY-I	L-32-070	n/a	$19.95

TRS-80 Game Programs

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

Hardware

TeleTrends TT512P modem (M4P)	H-4P-512	$74.95	E
XLR8er e/w 256K RAM (M4)	R-MB-004	$182.00	F
Floppy drives (5.25" 360K 1/2 ht)	H-FD-360	$75.00	D
Floppy drives (3.5" 720K 1/2 ht)	H-FD-720	$85.00	B
Floppy Drive Case (2-1/2 ht drives)	H-FD-2SV	$60.00	F
Hard drive kit e/w clock, 20Meg M3/M4	H-HD-020	$475.00	?
Hard drive kit e/w clock, 40Meg M3/M4	H-HD-040	$595.00	?
Hard drive joystick port option	H-HD-JSO	$20.00	
Hard drive: Kalok KL320	R-HD-020	$200.00	F
Hard drive: Seagate ST251-1	R-HD-040	$320.00	F
Hard drive: Seagate ST157A (16B IDE)	R-HD-A40	$320.00	F
Hard drive: Seagate ST-157N (SCSI)	R-HD-S40	$350.00	F
Cable: dual floppy extender	H-FD-2EX	$18.00	
Cable: 4Ft floppy (1 34EDC each end)	H-FD-C04	$12.50	
Cable: 4Ft M3/M4 printer	H-RC-PM4	$20.00	
Cable: 4Ft Radio Shack hard drive	H-HD-CT4	$20.00	
Cable: 4Ft MISOSYS hard drive	H-HD-C04	$22.50	
Cable: 26-1069 internal floppy	H-FD-2NG	$20.00	
Cable: 26-1069A/26-1080 internal floppy	H-FD-2GA	$20.00	
Cable: 26-1080/A internal floppy	H-FD-24P	$20.00	
Cable: drive power Y	H-HD-CPY	$5.00	
Cable: XT hard drive set	H-HD-CXT	$5.00	
Cable: Custom IDC ribbon (M3/M4/M2)	?-??-???	varies	
Standby Power System: 200VA	R-PS-200	$250.00	?
Standby Power System: 450VA	R-PS-450	$399.00	?
HD Controller: Adaptec 4010A	H-HD-CA4	$95.00	D
HD Controller: Xebec S1421A	H-HD-CX2	$95.00	D
T80 to SCSI host adaptor	H-HD-MHA	$75.00	D
Zoltrix ZOFAX 96/24 (PC XT/AT)	R-Z1-FAX	$225.00	F

The Fine Print

Freight codes: A = $3.50; B = $4.00; C = $4.50; D = $5.00; E = $5.50; F = $6.00; G = $8.50; H = $12.00; ? = varies; All unmarked are $3.00 each; Canada/Mexico add $1 per order; Foreign use US rates times 3 for air shipment. Virginia residents add 4.5% sales tax.

Virginia residents add 4.5% sales tax. We accept MasterCard and VISA; Checks must be drawn on a US bank. COD's are cash, money order, or certified check; add $4 for COD.

MISOSYS, Inc.
P.O. Box 239
Sterling, VA 22170-0239
703-450-4181; Orders only: 800-MISOSYS (800-647-6797)