

# NORTHERN BYTES



## Volume 7 Number 2

GREETINGS! These opening remarks are going to be very short this month, and that's because our letters column is very LONG. The letter from Roy Soltoff in Volume 6, Number 8 prompted a number of readers to take pen in hand (would you believe "word processing disk in drive?") and we have their letters, plus the normal volume of mail which is always heavy at this time of year, anyway (must be the "cabin fever" season...).

But before I go, I've got to pass this one along: I got an exchange newsletter from the Tandy Hobart Users' Group in Hobart, Tasmania (that's in Australia, in case you didn't know) recently. It was only two sheets so they mailed it in a regular airmail envelope. It then took the long way around the world to get to me. Would you believe that there were postmarks on the back of the envelope indicating that it had passed through Johannesburg, South Africa (some farsighted postal clerk probably read "R.S.A." instead of "U.S.A."), Van Nuys, California, and Indianapolis, Indiana before it got to me here in Sault Ste. Marie? The funny part is that two of the stamps on the envelope were commemoratives for Australia Post's new electronic mail service. If this is Australia Post's method of showing how fast electronic mail is in comparison with conventional Airmail, it certainly is an effective demonstration (as to the snail's-pace of regular mail, anyway!)

### THE EXTERMINATOR

Spring is sprung, the grass is ris, I wonder where the BUGS is? Right here in NORTHERN BYTES, unfortunately...

In NORTHERN BYTES Volume 6, Number 8, page 22 we published a zap to the Model I version of Alan Johnstone's NEWDOS/80 modifications. We said that the place to zap was SYS0/SYS.23.30. Well, it may have been byte 30 in an earlier version, but in the version that's on the TAS Public Domain Library disks, the area to zap begins at byte 5A.

Then, on the very same page and just across the column, we ran Bob Seaborn's instructions on adapting Model III INFOCOM games to run on the Model I. The second paragraph ends with '... change it to the new sequence "21 18 42".' The "42" should be 43. Then, in the next paragraph a reference is made to the DOS command buffer at 4218H in the Model I, but that address should be 4318H. Thanks to Greg Small for catching this one.

### LETTERS DEPARTMENT

Reminder: Persons sending letters intended for publication should send them on magnetic media or via CompuServe [72167,161], Delphi [TASIO], or MCI Mail [109-7407] (especially if longer than a couple of paragraphs). If you are NOT using Allwrite (or Newsprint) and your word processor offers the option to save your file in ASCII format, please do so (especially if using SuperScript!). Your cooperation in this matter will help us to bring you a better newsletter!

[Message received via CompuServe:]  
Date: 30-Jan-86 19:46 CST  
From: Jim Gaffney [70515,1036]  
Subj: Newdos80 vs LDOS

To: Jack Decker [72167,161]

Jack,

I feel compelled to comment on the exchange of letters between you and Roy Soltoff in Volume 6, Number 8 of NORTHERN BYTES, and I'm afraid that I have to side with Roy. At one time I nearly decided against continuing my subscription to Northern Bytes because it appeared to be becoming nothing but a compilation of patches to NEWDOS/80. I persisted, however, and have been rewarded with some interesting material. For example, I had the problem described in "FIX YOUR BLOOMING SCREEN!" but

assumed that I was an isolated case rather than realizing that I was experiencing another Tandy "save a nickle" design problem.

But I digress... I have had the impression in reading your comments in the past that your relationship to LDOS is like the woman with one child -- "Tried it once, but didn't care for it." LDOS is NOT an easy OS to master, but once you do, the rewards are rich. The flexibility of the system is incredible; it may take a while to figure out how to do it but more likely than not there IS a way.

At one time I, too, felt that NEWDOS/80 was the greatest thing since sliced bread. But then came double density and Apparat's decision to go with relative sector numbering. If they hadn't started that gobbledegook, I might have stayed with NEWDOS. Never mind that the directory track is no longer in the middle of the disk and relative track 17 is somewhere around physical track 9, as long as you use NEWDOS, you don't have to worry about it. And no automatic density recognition (without buying DSD from TAS) and different PDRIVE settings for system and data disks on a Model I? Bahhhhh!

I, like you, do not care for LDOS X'n'nnn' syntax, but the percentage of DUMP commands that I issue is not significant enough to keep me from using a superior DOS. Using LDOS for the Model III mode on my 4's and 4P's allows me to read and write directly from TRSDOS 6 disks, both SSDD and DSDD. And should I need to transfer files to a Model I user, I simply tell the machine to format in SD without going through a bunch of PDRIVE commands with the (a) option and then resetting the machine again after I've finished.

And where does one get support for NEWDOS problems? Surely not from Apparat who won't even answer their mail anymore! And you complain about having to patch programs to run them under LDOS? How about that wad of ZAPs to be applied to run everything from Electric Pencil to Visicalc V-316 under NEWDOS/80?

Do you have a problem with LDOS or TRSDOS 6? Then jump on over to CIS's Logical Systems SIG (PCS-49) and Joe Kyle DDPietropaulo (AKA jjkd) will get you an answer in a heartbeat. You say you want utility programs and filters for your LDOS? They are available in profusion from LSI, from MISOSYS and QSD, among others -- everything from full featured assemblers like EDAS to Compare/Modify packages that are extremely user friendly available in profusion. If you want a treat in a full-screen editor, get SAID (Mod 3) or PRO-SAID (Mod 4) from MISOSYS. It will load any kind of file (CMD,BAS,ASM,TXT, etc.) I'm using it for composing this diatribe and I use it for everything from C source code files to autolog files for Karl Denninger's DTERM II which requires imbedded characters with the high bit set.

But in the end, what we really need of Northern Bytes is support for all DOSes, not just NEWDOS or LDOS or DOSPLUS or any other. The only way we affectionatos of our Tandy Z80 machines will survive the MessDOS Menace is to bond together rather than fighting among ourselves. If you'll tolerate my LDOS prejudice, you may use NEWDOS all you want to in composing NB. (Do you really run Allwrite under NEWDOS?!)

Jim Gaffney

[Jim, I really appreciate your comments. To answer your last question first, yes, I do use Allwrite under NEWDOS/80 and it works just fine. Beyond that, I completely agree with the comments in your last paragraph. I make it a point not to turn away articles just because they refer to a DOS that I don't happen to care for (I even reprinted an article entitled "MODEL III TRSDOS 1.3 - THE BEST DOS !" and if that doesn't show a lack of prejudice, I don't know what does!) so I will again make the point that articles from users of all DOSes are welcome.

I will also agree that NEWDOS/80's relative sector numbering scheme for double density is one of the dumbest things I have ever seen (please note that I said "one of the dumbest", not "the dumbest"), even though I understand why they did it. It has caused no end of grief for NEWDOS/80 users. Folks have also complained, with some justification, about the gobbledegook that

is sometimes passed off as documentation in the NEWDOS/80 manual.

I happen to disagree that the PDRIVE scheme is bad, since it gives you almost total control over how you format a disk, though it may take you a few attempts to master the PDRIVE concept. I had a situation just the other day where I wanted to put a disk directory on track one (so that I could put a large, contiguous file on the rest of the disk) and NEWDOS let me do it with no complaints whatsoever. But, admittedly, that was a rather rare situation. Perhaps a better example of the flexibility of the PDRIVES is that they will allow you to copy files from or to a TRSDOS 1.3 disk, without the intervention of another utility program.

As for automatic density recognition, Alan Johnstone's NEWDOS/80 modification package (TAS Public Domain Library disk #ND-1) permits automatic density and side recognition of a number of different disk formats, including LDOS/TRSDOS 6. You might protest that having to buy the PD disk is an additional expense and that this code should have been included in the original NEWDOS/80 product. Fair enough, but then consider that NEWDOS/80 gives you, free of charge, utilities such as SUPERZAP, EDTASM, DIRCHECK, and LMOFFSET. Logical Systems charges extra for most of their utilities. In fact, to make a self-booting Model I LDOS system disk, you have to buy a utility called SOLE from Misosys, and go through what I consider to be an unreasonable number of gyrations to make it work (Roy Soltoff would no doubt dispute that since Misosys is his company, but on the other hand, we have reprinted an entire JCL file devoted solely to setting up a self-booting Model I system disk). In contrast, with NEWDOS/80, you simply set your PDRIVES properly and issue a simple COPY command (using the CBF option) and NEWDOS handles the format conversion, automatically and effortlessly.

Support for NEWDOS/80? The best support comes from other users of the DOS, and there are many of them out there. However, it is true that you might not get as much support as you'd like from Apparat, since (to my knowledge) there is only one person left at Apparat that understands anything at all about NEWDOS/80, and that is Brian L. Uitti. If you can reach Brian, he may be able to assist you, but if he can't help, you're on your own. It appears that the original author of NEWDOS/80 has turned to other pursuits and is no longer interested in supporting NEWDOS. But, again, there are MANY knowledgeable NEWDOS/80 users around (many of whom read NORTHERN BYTES!) that can help with various kinds of problems.

I suspect that many of the present deficiencies of NEWDOS/80 will be corrected in Bob Brumley's rewrite of NEWDOS/80 for the Model 4 (see last issue's page one mention of this Model 4 version of NEWDOS/80 that is now under development).

The point is that every TRS-80 Disk Operating System has its good and bad points, and neither LDOS/TRSDOS 6 nor NEWDOS/80 are any exception. I think that both have their place in the TRS-80 world, and that both (and all of the other TRS-80 DOSes) are far superior in many ways to MS-DOS. It would be a real shame if the "my DOS is better than your DOS" arguments drive a wedge between TRS-80 users. Nevertheless, I do feel that a discussion of the good and bad points of the various DOSes serves the interests of our readers by helping them to be better informed about the capabilities and disadvantages of each of the DOSes.]

Dear Jack:

... Regarding "Underused Basic Programming Statements": LSET and company should change a string literal within a BASIC program line, but they usually do not. I tried that trick in the past for self-modifying BASIC code. Unfortunately I found that the trick only works on MULTIDOS. The other Disk BASICs (perhaps for "safety") first move the target string to string space. For compatibility, I went back to the old POKE-it-in method. (See 80-Micro, 8/85, at page 112.)

Incidentally, my favorite "non-standard" use of LSET etc. is to quickly clear and write to the screen (as in a pointed LSET VID\$="Hello" rather than PRINT @ 64, "Hello"; CHR\$(30);). I have used CVI and company in programs but, in the end, they invariably related to information taken off of disk or perhaps in memory.

Andy Levinson

[You're right about not being able to use LSET or RSET to self-modify strings within a BASIC program under some DOSes. For example, NEWDOS/80 specifically checks to see if the target of an LSET or RSET string is within the program and if so, it moves it to high memory. The following segment of BASIC code illustrates the point:

```
10 IF PEEK(22809)=56 THEN POKE 22809,24 ELSE STOP
20 A$="This is a string literal"
30 LSET A$="A changed string literal"
40 LIST
```

Type the above lines in (under NEWDOS/80 BASIC only) and then type RUN 20. You will note that the string literal in line 20 is not changed. Now simply type "RUN" and note that the string literal in line 20 is changed to match that in line 30. The reason? The POKE in line 10 disables the test in NEWDOS/80 Disk BASIC that checks to see if the target string is within the program itself.

The two-byte word in reserved RAM that points to the start of the BASIC program is located at 40A4H (16548 decimal). This gives us a clue as to how BASIC can be "fooled" into permitting LSET and RSET to operate on strings within the program area. Try the following code (this should work under any Model I/III DOS, but I won't guarantee it):

```
10 A$="This is a string literal"
20 X1=PEEK (16548): X2=PEEK (16549): POKE 16548,254: POKE
16549,255: LSET A$="A changed string literal": POKE 16548,X1:
POKE 16549,X2
30 LIST
```

What we have done here is to save the current start-of-program pointer to variables X1 and X2, then we temporarily define the start of the program as FFFFH (don't use FFFFH as that indicates that there is no BASIC program in memory!) and then use the LSET command. Disk BASIC looks at the bogus start-of-program pointer, concludes that the target string is not within the boundaries of the program itself, and does the LSET. We then put the real start-of-program pointer back into place. WARNING: All of the commands shown in line 20 should be placed in the same program line if at all possible, and you definitely cannot do any GOSUBs or GOTOs (even an ON ERROR GOTO trap will bomb the program if executed) while the fake start-of-program pointer is in place. The reason is that BASIC searches for line numbers from the start of the program, and if it can't find the correct start of the program, it can't find any BASIC program lines referenced explicitly by line number.

Of course, for a program designed for your personal use only, it may be easier to just ZAP your version of Disk BASIC to eliminate the test (or use a POKE at runtime as shown in the first example above), if you know how to do it. Where there's a will, there's a way!

Dear Jack:

I have written ... to make some comments on Volume 6, #8.

I quite agree with your position regarding the Soltoff letter and LDOS. I mentioned the subject in our club newsletter.

Regarding the TRSDOS 1.3 patches at page 5: I believe that a goodly number of the patches by "original authors (unknown)" are from Northern Bytes. For example, patch 15 is the same as Tony Domigan's patch in Volume 4, Number 4; at page 15 (speaking of Tony, it was sad to read that he has "defected" to Big Blue). But as a contrasting example, patch 2 was issued by Radio Shack.

Patch 13 in the article, for changing the disk track count, did not have all necessary patches. Two more patches are needed:

```
PATCH *7 (ADD=4EB8,FIND=28,CHG=xx)
PATCH *14 (ADD=54EF,FIND=28,CHG=xx)
```

where "xx" is "28", "2A", or "50" for 40, 42, and 80 tracks respectively.

Regarding "More Patches to TRSDOS" at page 17: The patches allow use of a "." (period) for the date prompt up system boot but that accomplishes only half the job. There is also the DATE and TIME DOS library commands. Here are two patches to REPLACE the two patches that Mr. Birks gave for TRSDOS 1.3:

PATCH \*0 (ADD=4FB7,FIND=B9,CHG=AF)  
PATCH \*6 (ADD=53FF,FIND=B9,CHG=AF)

Both patches work the same. They allow you to use any character for the date and time entry separator. For example, "02.07.86" would work for the date and even "10<01>15" for the time. The two patch lines do not interface. The first is for the bootup entry request while the second is for the DATE and TIME commands. A person could apply only one if desired.

I did not bother to write a similar modification for TRSDOS 6.0 and 6.1 since both DOS versions have been superseded and should not be used if possible.

Andy Levinson

[Boy, Andy, you've kept your wordprocessor humming this month! Thanks for passing along the information. I'm really not too concerned about patches from NORTHERN BYTES being used without credit - as an editor, I know how difficult it would be to try and track down the original sources on all zaps.

In case any of our readers care, here's a quote from Mr. Levinson's article in "The Interface" (newsletter of the San Gabriel Valley TRS-80 Users Group), in regard to Roy Soltoff's letter to NORTHERN BYTES: "I felt that Soltoff's letter was one of the most arrogant pieces of writing that I have seen in quite some time. For example, he reported that a drive manufacturer introduced 'a drive with a microprocessor controlled speed that was a rock solid 300 rpm.' I would call that a major accomplishment, like a car that does not need constant carburetor adjustments or a record player that always plays records at the right speed. Soltoff described the drive as a 'fiasco'".

A record player that always plays records at the right speed?! Now, I could have really used one of those during my all-too-brief career as a Disk Jockey!!]

Dear Jack,

First, let me compliment you on the new look. The DMP-2100P sounds like a great printer (I've been impressed with our DMP-2100 at work).

Second, thanks for your comments on little-used BASIC statements. I had never really thought about the possibilities in LSET and RSET, but they were very useful in writing REPLACE/BAS and XFER/BAS.

I sure wish I'd had SAVESUB when I was writing my utilities. I had a few minor problems keeping the "versions" straight, and I lost one just the way they described in the article. Unfortunately Volume 6, Number 8 arrived just after I finished. If I'm ever foolish enough to write another program, you can be sure I'll use SAVESUB.

Could you explain the reasons for the patches to M-ZAL for TRSDOS 6.2? I have taken it off my DOSPLUS IV disk and used it under 6.2 to edit and assemble a few short programs without difficulty. What problem(s) have I been fortunate enough not to encounter that the patches fix?

Finally, a few comments on Mr. Soltoff's letter and Logical Systems. Although I agree with you that it is unreasonable to demand that the hardware conform to the software, I was inclined to view their decision as a value judgement plus rather narrow vision rather than arrogance. They decided that type-ahead was more important than a standard drive speed, but forgot that the purpose of standardization is compatibility. If some of us readjust our drive speed it becomes difficult to exchange disks. Also, think what it would do to someone who used SUPER/HYPER-CROSS to move data to CP/M or MS-DOS machines! Also, I've been reading about computers and using LDOS and TRSDOS for several years, and I had never heard about this, despite Mr. Soltoff's assertion that Logical Systems has "always told people of this phenomenon and recommended the drives be aligned to 301 or 302 rpm". I am very happy with both LDOS and TRSDOS 6. After trying several, I gave up on "alternate" DOS's, since too many programs won't run under one or another without modification. This was acceptable on the Model I where TRSDOS was so bad, but not on the Model III/4. Also PROFILE, which I use extensively, practically requires TRSDOS. Although I love type-ahead, I will give it up for reliable drive operation, if I have to. (Even with SMOOTH="NO", it is not entirely reliable during disk access.)

As I said, I was inclined to be charitable concerning "arrogance" until I heard another story recently. A friend bought LSI's LS-Utility Disk, and found that KSMPLUS/ELT would not work as advertised within PROFILE. When he called LSI for help, they confirmed that it should work with PROFILE, and insisted that the problem was with his hardware or his version of PROFILE, despite the fact that all other programs work on his computer. On two other Model 4's with another copy of PROFILE he got the same results, but they still insisted the problem was not with their software and offered no further help. Arrogance? Maybe you are right. They may have had reasons to believe they were right, but that's no way to treat a customer.

Don Singer

[Thanks for your comments, Don. Just in case any of our readers are wondering, the two BASIC programs mentioned in the above letter are ones that Don has contributed to the TAS Public Domain Library.

The article containing the M-ZAL patches was a reprint from another newsletter, so I can't really explain the reasoning behind them. Perhaps one of our readers can enlighten us on these.]

Dear Jack,

After reading the article on LDOS's "go to sleep" mode by Roy Soltoff [Volume 6, Number 8], I have to add my two cents. I never understood why no one has commented on adjusting the hardware. Are you supposed to void your warranty by removing the cover of your MODEL 3 to make the hardware adjustment? What do I adjust? If I find a screw on my drive do I turn it clockwise or counterclockwise? How much do I turn it? My LDOS manual says nothing about drive speed. Let's see, I need a utility program to time my drives (approximately \$29.95), and the Technical Manual, another \$29.95. Because I am no hardware expert, I should elicit the help of a qualified technician -- another 40 bucks! Does the MODEL 4 Technical Reference Manual tell us to adjust the speed to 301 or 302 rpm? Could I use 301.5?

I don't use LDOS, not because of the "go to sleep" mode, but because of the "holier than thou" attitude I have received from both Soltoff and LSI.

COMMENT ON FLOATING POINT ACCURACY:

The floating point errors are caused by the conversion of the decimal numbers to binary, operation performed in binary, then the conversion back to decimal.

COMMENT ON L?:

Volume 7, Number 1 letter from Bob Seaborn. BASIC does not store the '?' token. At ROM address 1BE5, the '?' can be found:

1BE4	CP	'?	
1BE6	LD	A,0B2H	:PRINT token
1BE8	JP	Z,1C5BH	

If BASIC encounters a '?' as the FIRST character in a statement or after another key word, then the '?' is changed to the PRINT token. In order to change L? to LPRINT, the following code would have to be inserted (no suggestions how) into the ROM:

1BE4	CP	'?	
1BE6	LD	A,0B2H	:PRINT token
1BE8	JP	Z,1C5BH	
	LD	A,B	:B holds
	CP	60H	
	JR	C,\$+8	
	CP	80H	
	JR	NC,\$+18	
	AND	5FH	
	CP	'L'	
	JR	NZ,\$+12	
	INC	HL	
	LD	A,(HL)	
	CP	'?'	
	LD	A,0AFH	:LPRINT token
	JP	Z,1C5BH	
	DEC	HL	
1BEB	LD	A,(HL)	

What I'd really like to do is contribute to Northern Bytes. I have plenty of knowledge on the TRS-80s but I don't know what the users are interested in.

Vern Hester

[Thank you for the comments and the information, Vern! As most of our readers already know, Vern is the author of MULTIDOS, and is not in the least overstating the facts when he says he has "plenty of knowledge on the TRS-80s". Vern is the only person I have ever known that can look at machine language code (hex bytes, like you see in a ZAP program display) and tell you what the disassembled code is. Also, Vern has a pretty good idea of how most of the other DOSes work (as evidenced by the fact that MULTIDOS is capable of reading disks created under just about ANY of the popular TRS-80 DOS formats). So, if you would like to see an article on just about any aspect of DOS or ROM code operation (or just about anything else having to do with the TRS-80 from a software standpoint), why not send your suggestion to Vern? Mark your envelope to the attention of Vern Hester and send it to NORTHERN BYTES at our Sault Ste. Marie address, and I will forward your letter on to Vern.]

Dear Jack,

Thought I'd drop a line and comment on the great new look of Northern Bytes. Not only is it more readable, it is also a lot cleaner looking, layout-wise.

I'd also like to comment on the NEWDOS/LDOS/TRSDOS debate. They are all good DOSes and have their good and bad points. I first used NEWDOS as a new computer user several years ago when I discovered several things TRSDOS would not allow me to do easily and turned to NEWDOS as the better DOS. Later when I was deeply involved in research and writing as a PhD. candidate I became disgruntled when I learned that NEWDOS could not handle Profile III+, my data base manager of choice. I needed a DOS which would easily handle Profile III+ and SuperScript, my word processor of choice. As fate would have it only LDOS solved my particular set of needs.

I used NEWDOS enough to be impressed with its superior file handling capabilities, but am not sadist enough to attempt the use of several DOSes on a continuing basis; after all, I want to be as efficient as possible with my computer and using several DOSes is not compatible with that end, at least not with me. So, I've become a habitual LDOS user and have grown quite fond of it. It supports the little hacking I do to make all my applications run together more smoothly and causes me very little trouble.

I have noticed a sort of frustration on your part concerning SuperScript. My wife and I both write for a living; she pursues a journalistic path while I follow one of scholarly research and editing. The only word processor for our two Model IIIs which satisfied our individual needs was SuperScript. If she had been introduced to other WPs such as Newsprint, etc., she would have panicked and quit computing even as she started. All of those little codes and command lines scared her. However, she took to SuperScript like a duck takes to water and marvelled at how similar to a regular typewriter it was. For my part its use of virtual files rather than memory allowed me to construct documents of well over a hundred pages (although I'd like to get around the 120 page limitation someday). No need to "chain" documents; I could just write and block move text all over the place. I also needed the feature of footers and the need to send printer codes, so SuperScript was our compromise.

Anyway, I wanted to let you know that lots of us really need and appreciate the "non-hacker" software that we use to earn our daily bread. The mainline software that Radio Shack has provided was really a bonus -- imagine where we would have been (in those first days before we knew of 80 Micro, and Northern Bytes, and all the other helpful magazines and newsletters) if we had bought our first computer and had to navigate the jungle of third party suppliers. I suspect we would still be using the trusty old electric typewriter and ekeing by on about 1/3 the output we now enjoy.

Now, considering all that digression I just went through, I only suspect that a lot of potential subscribers to Northern Bytes are a little turned off by what they sense to be software favoritism on your part. Personally, I don't mind; I get rather a kick out of polemics anyhow. I also think that a lot of the users of Radio Shack software would love to learn a few patches and zaps to allow them better use of what they have. You've done a great job in that respect (patches) in the year or so I've been

subscribing -- I would only suggest that non-hackers comprise a larger market these days than do hackers.

A friend of mine more creative in his hacking than myself did a breakdown on SuperScript's use of linked lists and file formats. If one is diligent enough he can even reconstruct a crashed file using his data. I'd be happy to send you a hard copy if you are interested in seeing it. I've also picked up a few handy SuperScript tips along the way, such as: the user key configurations are in the file SYSTEM/CTL and various combinations can be kept by copying to a file with another name before going on to a new configuration. I also keep all my names and phone numbers in a file named ADDRESS. Since I am almost always using SuperScript I can find an address or phone number in seconds by going to that file and using global search. Its been one of the handiest time-savers I've come across.

Incidentally, have you heard anything about the efficacy of the new 1 meg memory add on boards for the Model III. Any software out there for it? Would you consider such a board to be a good buy?

I hope I haven't sounded too negative in my comments. I only wanted to give you a grass-roots report from the non-hacking world. Keep up the good work.

Samuel J. Wells,

530 Buschman Street, Hattiesburg, Mississippi 39401

[Sam, I'm printing your address because I haven't heard all that much about the one meg memory boards yet (though the few reviews I've seen have tended to be mostly favorable), and perhaps one of our readers would care to drop you a line and tell you about their experience with the expanded memory boards.

Thanks for adding your comments to the fray. My only frustration with SuperScript has been that if someone sends me a file in SuperScript's native format, it is a real problem to convert to standard ASCII. That's only because I don't use SuperScript regularly, of course. I normally use Allwrite and while Allwrite does make use of "those little codes and command lines", I found it fairly easy to learn, and I can do just about anything I want to with it. But, as I've said before, what's right for me may not be right for you, or for your wife. There are many word processing programs available for the TRS-80, but most of us pick one and use it regularly, to the exclusion of the others.

That's why, when you say that you suspect "that a lot of potential subscribers to Northern Bytes are a little turned off by what they sense to be software favoritism" on my part, I have to ask "Why?". Show me a man that claims he doesn't have any software favoritism and I'll show you a man that should be running for president of the Liar's Club. Almost every computer user, at any given point in time, will use only one DOS, one word processing program, one data base manager, etc. You state that you, yourself, are "not sadist enough to attempt the use of several DOSes on a continuing basis" - does that mean that you are showing software favoritism toward one DOS? Sure you are. And so do I, and so does everyone else who writes in to defend their favorite DOS. And there's nothing wrong with that!

Of course, if I were to exclude articles from NORTHERN BYTES because they pertain to a DOS or a software product that I don't happen to use, then you'd have a valid complaint. But I don't do that. Unfortunately, though, what occasionally does happen is this: Let's suppose somebody sends me a really poorly-written article about, for example, some aspect of a particular Disk Operating System. Now, if it happens to be a DOS that I use, I may be able to understand and appreciate the article even though it is poorly written. Not only that, but because of my knowledge of the DOS, I'll be able to edit the article to some semblance of readability. But, if the article pertains to a DOS that I do NOT use on a regular basis, I probably will NOT be able to understand it well enough to rewrite it so that others can understand it, and in that case I may decide not to use it. But this kind of situation occurs only VERY rarely, since most of the articles I receive are fairly well written to begin with.

It's probably worth stating the obvious point that any editor of any publication will be more comfortable about including an article when he knows something about the subject matter of that article. But if the style of an article suggests that the author knows his subject, it goes a long way toward calming an editor's misgivings!

Now, when you say that non-hackers comprise a larger market these days than do hackers, that's probably true. But non-hackers also tend to be notoriously hard to please when it comes to reading material. If you aim your publication at the rank beginner, you tend to lose your audience because rank beginners don't stay rank beginners forever, and soon your publication is beneath them. On the other hand, if you aim at the more experienced user, you get constant complaints that your publication is too advanced. If you try for a mix - something for everyone - then maybe no one is happy because there are only one or two articles of interest to any given individual in an issue of the publication. In spite of all this, many computer publications aim for this general market. You may also have noticed that many of them have gone belly-up in the last year or so.

We aim NORTHERN BYTES at the more experienced TRS-80 user - the "hacker", if you will - and while this group may not be as large numerically, they tend to show a great degree of loyalty to their favorite publications. And, since the TRS-80 market is not exactly expanding by leaps and bounds these days, we need that kind of reader loyalty if we are to survive for any length of time. I suspect that if we had aimed NORTHERN BYTES at the rank beginner, we'd have been dead in the water after about two issues (if for no other reason than because we'd have been directly competing with 80-Micro, which has a lot more in the way of resources than we do!)

I do appreciate all of your nice comments and complements about NORTHERN BYTES, and hope that you didn't find my reply too negative-sounding - or too overly-long!

Dear Jack,

I have a short hint for users with LDOS or TRSDOS 6 system disks on other than the standard 40 track single-sided drives. Normally when making a new system disk the /SYS files will be copied to the lowest numbered tracks, giving a lot of head movement between the directory and the /SYS files. When making a new system disk, try formatting the disk, CREATE a large file or files using up almost all the space between track 0 and the directory on the disk. Then copy across the /SYS files, delete the large CREATED file, and copy across all the other required files from the original disk. The amount of head movement should be reduced considerably. You can use the FREE command between the intermediate steps to see how the disk allocation has been done. For 80 track double-sided disks the CREATE file could be about 1300 records of 256 bytes, for 40 track double-sided about half that size should be appropriate.

Concerning adding more than 2 external drives, there's no real problem with attaching them - the main problem is accessing them. Many of the old 8" disks had straps which could be set so they could either be addressed with a binary drive select (0 to F using 4 lines) or with a 1 of 4 system as used on TRS-80s. On the Model I some of the disk systems used drive 3 select as a side select, since side select was not on the bus connector. If all drives were single sided you could use side select to select from two groups of two external drives, but the DOS would have to be modified to allow this. By combining side select and the two external drive selects as a binary code you should be able to use up to 7 external single-sided drives, but this number will probably overload the bus. Using the drive selects alone as a binary code would only allow three external drives.

Arne Rohde

[Arne, thanks for passing along the hint for TRSDOS6 and LDOS users. I suspect that the same principle could be used with other DOSes as well, although it might be a bit more difficult to determine the size of the CREATE file under NEWDOS/80 due to the relative track scheme used by that DOS (which means that you have to use SUPERZAP to determine the actual physical track on which the directory resides).

I probably should have explained that the gentleman that wanted to patch NEWDOS/80 to recognize more than two external drives really didn't care that much about being able to use additional physical drives - what he really wanted to do was to have NEWDOS/80 accept a drivespec greater than 3. The reason he wanted to do this is that he is using a memory disk driver that makes the added 64K of memory in a Model 4 into a MEMDISK. When you install the MEMDISK, you have to specify which drive number it will be seen as by the DOS. Therefore, if you have four drives in the system and the highest drivespec permitted by

the DOS is 3, the only way to enable the MEMDISK is to knock one of the physical drives off line. But if NEWDOS would accept commands such as, say, DIR :4, then the MEMDISK could be installed as drive 4. Thus the drive select line problem wouldn't really be a problem.]

Dear Mr. Decker,

In Volume 6 Number 8 [page 8] you asked about the Easter algorithm. With the aid of the Information Please Almanac I have figured out some of the program.

Easter is the first Sunday following the first full moon on or after March 21. This means Easter is between March 22 and April 25. A lunar month is actually 29½ days long, but rounded off to 29 days for usage. Because of differences with the solar year, lunar years are grouped in periods of 19 years. Seven have 13 months while the other dozen have only 12.

Program explanation:

- 10-20 Year 1900 is a period boundary.
- 30-50 Determine year of period.
- 60 Find out if year has an extra month.
- 70-80 Number of days already past in present lunar month at start of solar year.
- 90 Number of leap years.
- 100-110 Number of elapsed weeks - 31 is a fudge to make it relative to Sunday.
- 120 Calculate relative to April 25th.
- 130 If negative, it must be March (31 days).

Edward F. Leach

[Thanks, Ed. This might be helpful to those writing calendar and scheduling programs]

Jack,

I recently received a letter from a gentleman in Australia. He had tried to apply my NEWDOS/80 modification for 4P self boot (see Volume 6 #6) to a double sided disk, and didn't have any success. I tried the same thing and found that I couldn't get the ROM loader to access the second side of the disk. I figured out that I could fake out the ROM loader by putting the MODELA/III ROM image sectors on the front side only. Of course, this wasted all the sectors on the back side of the tracks where the ROM image was. After sending him this information, I was reading the 4P service manual and found that the ROM loader does check for double sided drives. Finally after spending some more time DIGGING through the service manual and the Model 4 Technical Reference manual, I found that the ROM loader checks bit 5 of byte CD of the GAT sector to determine if the disk is single or double sided. As a result, here are the steps necessary to create a double sided 40 track self-booting disk. I would appreciate it if someone could try the same thing on an 80/80 track drive and let me know if it works.

The key to making any self booting disk that uses relative tracks that span physical tracks, is that the real or a fake GAT sector and FPDE for the ROM image file must be on the track pointed to by relative byte 3 of the boot sector.

1. Create a single sided double density version from your original NEWDOS/80 master.
2. Copy the MODELA/III file from a TRSDOS disk to your new NEWDOS single sided disk. Make sure you use the correct ?DRIVE setting.
3. Using your master, create a double sided system disk (PDRIVE specs TD=C, TC=40, SPT=36, DDSL=17).
4. Kill off any unnecessary files from your double sided disk except SUPERZAP and DIRCHECK.
5. Boot up on your double sided disk. Use SUPERZAP access relative sector 170 (GAT sector of directory).
6. Verify that bytes 3DH through 43H are FC's (if they aren't then you will have to KILL, COPY, or somehow move files around to release those granules).
7. Modify bytes 3DH through 42H and change the FC's to FF's and modify byte 43H and change it from FC to FD.
8. Move to sector 171. There should be a 00H in byte 60H. Change it to a 25H.
9. Move to sector 171. Beginning at byte 60H, this should be a 00. Modify the next 32 bytes as follows:  
5F20 0000 0053 5953 3232 2020 2053 5953  
5678 1234 4100 3D0C FFFF FFFF FFFF FFFF

10. Place your single sided disk with the MODELA/III file on it in drive one and make sure the PDRIVE specs on your double sided drive are set up to access it.

11. Using the DFS option of SUPERZAP, access the MODELA/III file and write down the beginning and ending relative sector numbers of the file and the number of sectors in the file (number of sectors in file = ending relative sector number - beginning relative sector number + 1, it should be 57 sectors).

12. Use the CDS option of SUPERZAP to copy the sectors of the MODELA/III file from drive one to drive 0 BEGINNING at relative sector 618.

13. Zero out sectors 612 through 617.

14. Access relative sector 612 on drive 0 and modify byte CDH and change it to 20H. This sector represents the fake GAT sector and byte CDH is what that ROM loader looks at to see if the disk is single sided or double sided (bit 5 of the byte is the flag: 1=double sided, 0=single sided).

15. Access relative sector 614 on drive 0 and modify the 32 bytes beginning at 00 to the following (this is the fake FPDE for the ROM loader):

```
1000 0000 004D 4F44 454C 4120 2049 4949
9642 9642 3900 1129 FFFF FFFF FFFF FFFF
```

15. Exit SUPERZAP and run DIRCHECK to make sure the directory is OK.

16. This double sided disk should now self boot on the 4P.  
Art Rasmussen, 612 West Hillcrest, Keene, Texas 76059

[If any of you try using this method with an 80 track double sided drive zero, please drop a line to Art and to us here at NORTHERN BYTES to let us know if it works!]

Dear Jack,

Whatever else you do please keep the TANDY HOBART USERS' GROUP (THUG) of the [address below] on "NORTHERN BYTES" mailing list!

I have been meaning to write for some time, but I never quite got round to it. The sad "passing" of Tony Domigan to 16 bit and the "PROSOFT ON PIRACY" article in "Northern Bytes" Volume 6, Number 8 provided the necessary catalyst.

Jack, space is at a premium in "Northern Bytes". However, I consider you should leave a "blank line" (or two) in honour of Tony Domigan's many contributions to "Northern Bytes". I was beginning to think that he had seen the light as he had started to transfer his files to TRSDOS 6.X (from NEWDOS) when I was told by a "shell shocked" friend that Tony had deserted the TRSDOS ranks. (I know your feelings about TRSDOS 6.X. If someone would fix the NEWDOS manual to make it "readable", I would feel more kindly towards it.)

Jack, what is the US Postal Service coming at with this 10 character zip code? (A bit of redesigning had taken place here to accommodate 5 digit zip codes on the THUG data base, and now some more hasty alterations are in order in the light of your new 10 digit zip code!)

Mention has been made in "Northern Bytes" of Dick Smith Electronics starting up in the U.S.A. Dick Smith Electronics no longer support the "System 80", a Model I clone, nor do they support the "Exidy Sorcerer" which they formerly marketed. Tandy Australia still supports the Model I. This, of course, is highly commendable.

Incidentally, Tandy Australia's software support is not all that it should be. Obviously their Australian management doesn't use Model 4 SuperScript otherwise they would realize the importance of the patches that were recently published in "Northern Bytes" and inform the local Tandy stores. (Having the computer lock up is no joke!) If we return the software registration cards in the manuals to Fort Worth we hear nothing. As far as I know, no data base of registered owners is kept in Australia. This is why "Northern Bytes" is so valuable to us who live outside the U.S.A., since it keeps us informed about the latest developments in the TRS-80 world.

Speaking of those SuperScript patches Jack, we are delighted that you now have a 2100P printer. We sure had a hard time typing in the patches from "Northern Bytes". Magnifying glasses were needed! (Sorry, Jack. I would willingly use a magnifying glass to read the whole of "Northern Bytes", if there was no other alternative.)

More seriously, I was interested to read Chuck Tesler's reasons for not selling his company's products to Australians. I wonder if there are more software pirates per head of population

in Australia than in the U.S.A.. (I think not.) I object to being discriminated against because I am an Australian! I too am not happy about Chuck Tesler's reference to "American English". We have similar machines (except for voltage requirements) and identical software to that available in the U.S.A.. ("American English" won't wash where Dot Writer is concerned.) I wish to obtain Model 4 Dot Writer and I appear to have the following choices:

(1) Get a US resident to buy it in the US and send it to me. (This may cause me some problems with support and upgrades.)

(2) Obtain a pirate copy. (This is not too difficult. I do not condone piracy. In any event support is essential.)

Dot Writer is a near essential for me. What would you do in my position, Jack? For that matter, what would Chuck Tesler do?

Before I knew about Prosoft's self-imposed ban on selling to Australians I wrote two letters to his company requesting information re Dot Writer. Rather annoyingly, there were no replies!

A query. Has anyone ever had Model 4 Le Script lock up when Electric Webster's Grammar and Style checker's option is being used to check a file? I wrote to Cornucopia some 3 months ago but haven't yet received a reply.

Bert Smith

("Bert's Ramblings", Tandy Hobart Users' Group)  
G.P.O. Box 1271N, Hobart, Tasmania 7001, Australia

[We also miss Tony's insights in these pages. However, Tony is apparently alive and well and struggling with MS-DOS. He occasionally sends me a note via MCI Mail or Delphi.

Because you're in Australia, you are probably not aware that the nine digit (plus hyphen) zip code is considered optional. At this point in time only about 10% of the mail uses it (this is just my estimate, not a post office statistic). The first five digits (which are used by almost everyone here) directs the letter to the proper city and/or post office. The last four digits further directs the mail to a specific location - perhaps one side of the street for one block, or a specific apartment or commercial building, or (as in my case) a small group of mobile homes within a mobile home park. So, although you may wish to use the nine-digit zip codes to speed up delivery, they are not required.

I think you may be just a bit confused about the situation with Prosoft. As far as I know, there was never a ban on selling Dotwriter outside the U.S. Dotwriter is a dealer product (in other words, another software supplier can obtain it from Prosoft for resale purposes) so you should have no difficulty obtaining it from Prosoft or from another dealer (maybe I should insert a plug here for The Alternate Source...). However, Allwrite is a Prosoft product that is sold only directly to the consumer by mail order, and is the product that was unobtainable in Australia. But, as explained in last issue's "The Exterminator" column, Allwrite is now sold to Australia and other countries, provided the prospective purchaser first signs a release (obtained from Prosoft) acknowledging certain limitations on the amount of support he may expect to receive. I cannot explain why you never received a reply on your letters to Prosoft, but perhaps you should send copies addressed directly to the attention of Chuck Tesler, and see if that helps.]

Dear Jack,

..... I'm interested in hearing from anyone who would like to see the Model 4 version of VisiCalc modified to perform some advanced spreadsheet functions (ala Lotus). If I get enough feedback, I'll seriously consider writing a modification package for it. This is something I'd really like to see.

One more thing, I recently returned to the States for a week and a Radio Shack "tech" told me some interesting things about the "new" 4D.

He said that he'd received a service bulletin from Tandy that says the 4D won't recognize external peripherals. He went on to say that "Orchestra-90", voice synthesizers, etc. weren't functioning, period. Basically the only thing that was working was a Hard Disk. I hope this isn't another case of "shooting themselves in the foot".

I'm not a hardware man, I don't have a 4D Tech Manual, and I don't have a 4D to work with, but it SOUNDS like the external I/O select line.

You may wish to pass this along to your readers.  
Michael R. Johnston, E. Co. 4th Spt. Bn., APO New York 09185

[Anyone interested in the VisiCalc mods, write to Mike!  
As for the 4D "problem", I think this is really more of a case of a misinformed Tandy employee (at least I HOPE that's all it is!). I passed your question along to Bryan Eggers of Software Affair, which produces Orchestra-90 and also runs the Tandy Special Interest Group on Delphi. Bryan replied, "The Orchestra-90 has been tested on a Model 4D and there were no problems. I've also had this reported by several other people. That's the first report I've heard claiming that no peripherals work on the 4D. I was told that Tandy didn't get FCC approval for the ORCH-90 as used on a 4D. For that reason, a Tandy bulletin said that 'ORCH-90 was not intended to be used on the 4D' or words to that effect, but not that it didn't work. Could be that Tandy has a new board revision that introduced new incompatibilities in later 4D's. Tandy customer service ought to have fun with this one."

Another Delphi user made the comment, "Weird thing (to me) is that, though the Hard Disk works fine on the 4, Orchestra-90 is said not to but it uses the Hard Disk port."

A little further searching on the Tandy Forum on Delphi came up with the full story on the problem. This was also a message from Bryan Eggers, but it was left on the system way back on October 11, 1985 so he had no doubt forgotten about it. It read:

"As a computer peripheral, ORCH-90 had to be checked for radio frequency interference (RFI) levels and certified by the FCC. When ORCH-90 was originally certified by the FCC it was for use on a Model III. However, Tandy recently determined that this certification did not automatically apply to its use on the Model 4/4D.

"Therefore, Tandy had to hastily instruct its salespeople NOT to specifically recommend the product for use on the 4D because they cannot legally sell the product "for the 4D" without getting into trouble with the FCC due to lack of certification on that particular computer. Tandy is attempting to resolve this problem, but as you are probably aware, FCC certification can take time and at this late point in the ORCH-90 and Model 4-series product life cycles, I doubt that it will receive much priority.

"As you have probably already guessed, the ORCH-90 works fine on the 4D as well as the III/4/4P. There are no incompatibility problems whatsoever. Hope this clears up the confusion."

I would assume that this same FCC registration problem might apply with regard to voice synthesizers and other peripherals. If anyone knows of an actual case where a Model 4 peripheral refused to operate properly when connected to a Model 4D, we'd like to hear about it!

Dear Jack,

Do you or any of your readers have any ideas on how to install a TRS-80 Model I CPU and keyboard, plus the expansion interface and RS-232 boards in a Model III case? If this is possible, I think the CRT and disk drives could be installed using Model III hardware. Just think how nice it would be to move a Model I in less than two or three hours.

Jack, I hope Northern Bytes continues. It is a great newsletter and I commend you for the hard work and effort put into each issue.

Dale L. Kerr  
Agri Data Control, Route 2 Box 64, Claude, Texas 79019

[Thanks, Dale. I've never heard of anyone putting a Model I into a Model III case, and I suspect that it would be a rather difficult conversion since the Model I boards were not designed to fit into a Model III case. If I were you, I think I'd consider selling my Model I system and purchasing a (used) Model III system in its place. The price difference between the two models probably isn't that great anymore. Of course, if you have a lot of Model I software that is not Model III compatible, you might need to figure the costs of replacing your software into the calculation.

If any of our readers have managed to perform this type of conversion, perhaps they will drop you a line and let you know how they did it!]

Dear Sirs:

I am interested in finding out what happened to Micro Projects Engineering Inc. (3951 Higuera Street, Suite B, Culver City, California 90230). They made the microMERLIN that later was developed into the XCALIBER (see 80-Micro review, September 1984, page 36). Do you know who I could contact to find out more about them and if they still have a way of supplying their product?

Joel B. McDonald,  
13048 S.E. Ridgecrest Road, Portland, Oregon 97236

[Sorry, Joel, no idea at all. Readers?]

Dear Jack, ... I am looking for old TRS-80 text adventure games and simulations, especially ASYLUM II. I want original copies with original documentation ... I am also interested in the old text adventures from Softside and other obscure sources.

Michael A. Mahaffey, Box 6396, APO New York 09057

[Readers, got any of these you want to get rid of? Contact Mike!]

Dear Jack,

..... a comment on the Stoner proposal for a Six-meter Packet Radio Service. I am a Ham, and I think its great, but it will meet heavy opposition from the common carriers unless it is restricted to non-commercial uses the way amateur radio is. There is some justification for this. The commercial services are expensive, but offer a degree of reliability, speed, service and privacy needed by many users, especially for business, for which they deserve a profit. Amateurs can't guarantee this level of service (e.g. coverage of a specific geographic area, rapid delivery or 24-hour response to equipment failures) which is no problem for personal use. Perhaps this service should be part of the Amateur Radio Service. I suspect the F.C.C. will be reluctant to approve it without operator licensing, and I think that the computer public might accept testing if it were done properly. For example, a written examination (no code) comparable to the present Novice Class test, covering the Regulations but substituting VHF/digital equipment and operating practices for HF/CW modes might qualify the applicant to OPERATE commercial equipment and use the service. To design, build, repair or adjust equipment would require passing a more difficult test, probably the current Technician Class amateur exam, perhaps modified for more emphasis on digital communications. Holders of some of the current amateur licenses could also be allowed full privileges, and holders of the "computer-only" license could upgrade to other classes of amateur licenses if desired. This would benefit both the computer public (access to radio communications) and amateur radio (more members, some of whom would become interested in other aspects of the hobby).

Jack, I was sorry, but not surprised, to read your impression of amateur radio. I'm afraid our official actions do give the impression of "self-righteous, elitism", but most hams don't feel that way, and try hard to help others enter our hobby. The ARRL's position reflects member's letters on a recent "no-code" license proposal, but I think the responses were skewed, since most opponents of the proposal felt very strongly and expressed their opinions, while those not opposed (myself included) were probably less motivated and failed to write.

I intend to send my comments on the Stoner proposal to the F.C.C., to Don Stoner, and, after I read their comments, to the ARRL.

..... I'm looking forward to reading the letters in Volume 7, Number 2.

Don Singer

[But were you looking forward to reading THIS MANY letters?? My mailman is demanding an assistant....

To respond to your comments on the Stoner proposal: First, I agree that the service should be non-commercial in principle, but I think that the Ham radio regulations go far overboard. As I understand it, you can't even conduct a private sale over Ham radio (with the interesting exception of Ham radio equipment). In other words, a Ham who wants to sell his used car or his stereo equipment to another ham can't legally discuss the details over the air. This, to me, seems an unnecessary restriction, particularly on a digital radio service. It is my feeling that

private transactions should be unrestricted, so long as they are incidental to the use of the service. Because of the characteristics of the service you mentioned (the possible lack of reliability, speed, service, and privacy), the service will probably not be suitable for general commercial use. Therefore, I don't feel that any further regulations (which, if enacted, would probably be ignored anyway) are necessary.

It can be argued here (and I will do so) that the service is NOT intended to be another amateur radio service! So what if people want to conduct casual business over the airwaves. Who's being harmed? No one, except for those "purists" who want to dictate to those outside of their ranks how the airwaves should be used (another example of "self-righteous, elitism"?). I guess that this is the main reason that I would really prefer to see such a service as completely separate and distinct from the amateur radio service - users of the proposed service will have certain needs and expectations for the service which will no doubt conflict with the goals of many of the amateur radio organizations. I doubt that the American Radio Relay League would ever be able to fairly represent users of a digital packet radio service.

If the equipment is constructed properly, there will be no need for operator examinations. We don't require them for the use of cordless or mobile telephones, or two-way (business) radio stations. Generally speaking, the equipment can easily be designed to prevent misuse, and the operator will only need to know a couple of rules: 1) It's illegal to tamper with the innards of the equipment, and 2) It's illegal to transmit profanity or to make slanderous or libelous statements on the service. That little bit of knowledge hardly requires testing - and, anyway, from all indications it appears that the Federal Communications Commission wants to eliminate testing and licensing requirements as much as possible.

I'm sure that there are MANY fine amateur radio operators out there. I just wish they'd realize how bad an image of your hobby a few of your "purists" (maybe I should say "traditionalists") are leaving in the minds of others. Japan (which has eight times more amateurs per capita than the United States) and several other countries offer a "no-code" license, and it is high time that the amateur radio hierarchy here in the U.S.A. realizes that there are thousands of potential amateur radio operators out there that do not wish to learn or use the Morse code!! At present these people are being denied access to the airwaves (or forced to use an inferior service such as the Citizens Band) by this ... I hesitate to say it ... "self-righteous elitism". I suspect that the REAL reason that some amateurs want to continue the code requirement is either a) because they figure that THEY had to suffer through it, and want to inflict the same pain on others entering the hobby (this type probably feeds their kids Cod Liver Oil, too), or b) because they think that they're just a little better than everyone else, and don't want to see too much "new blood" (people who, in their minds, don't have their same high standards) entering the hobby.

I think I'd better shut up before I start sounding too much like Wayne Green (who has purchased "73", an amateur radio magazine, back from CW Communications and is starting to publish his fiery editorials again)...

#### CONVERTING TOSHIBA P1351 PRINTER FONTS FOR USE WITH THE TANDY DMP-2100P PRINTER

If you happen to own or use both a Toshiba P1351 (or the newer P351) printer and the Tandy DMP-2100P printer, you may have wondered if the Toshiba downloadable fonts (sometimes given away by Toshiba dealers with purchase of a printer) can be used with the Tandy printer. If properly converted, the answer is "yes". The program below does the conversion and, as a bonus, creates a /TAB width table file for use with Allwrite (which you can just delete later if you don't need it). Once converted, these fonts are used in exactly the same manner as the downloadable fonts sold by Prosoft. If you don't have the Prosoft downloadable fonts or Allwrite, then you'll also need a program to read the bytes from the converted file and send them to the printer. That program could be written in either BASIC (a-l-o-w!!!) or machine language, but I'll leave that as an exercise for the reader.

This is a "quick and dirty" program and could probably be rewritten for better appearance or faster execution, but since it is basically a "one application" program, I didn't think it was worth the extra effort. Once the fonts have been converted, you don't need this program any more!

To use, simply give the program the name of the file you want to convert (which must NOT end in /FNT or /TAB - rename it first if it does) and the drive number for the destination files. Then go away for a few minutes and let it work. It will create the download font file, containing the actual bytes that need to be sent to the printer to download the font (with a /FNT extension) and the Allwrite width table file (with a /TAB extension).

One warning - there may be copyright restrictions in effect for the fonts distributed by Toshiba, so it might be wise to check with your Toshiba dealer before you do any conversions (if you can buy a Toshiba font disk from a Toshiba dealer, then you would probably be legally entitled to use the fonts with any one printer of your choice, whether Toshiba or something else. But I'm not a lawyer, so don't take my word on that subject as being necessarily correct). Of course, if Toshiba printer users have created any fonts and placed them in the public domain, there would be no restrictions in converting those fonts for use on the DMP-2100P.

One other note of interest - Arne Rohde (Box 82-211, Highland Park, Auckland, New Zealand) is putting the finishing touches on a font editor program that will (hopefully) work with both Toshiba P1351 and Tandy DMP-2100P fonts. This will be a user-supported program (you send a contribution to Arne to register the program and receive updates). Hopefully, I'll have more on this later but if you've been wanting to create your own fonts, you might want to get in touch with Arne.

```

10 CLEAR1000:LINEINPUT"FILE TO CONVERT: ";A$
20 IF INSTR(A$,"/FNT")>0OR INSTR(A$,"/TAB")>0 THEN PRINT"DUPLICATE FILE
  NAMES":END
30 OPEN"R",1,A$,1:T=INSTR(A$,"/"):IFT>0 THEN A$=LEFT$(A$,T-1)
40 INPUT"OUTPUT TO DRIVE #";T:IFT<0 OR T>9 THEN A0ELSE B$="":MID$(ST
  R$(T),2,1)
50 OPEN"R",2,A$+"/FNT"+B$,1:OPEN"R",3,A$+"/TAB"+B$
60 FIELD1,1ASA$:FIELD2,1ASB$:FIELD3,1ASX$,255ASZ$
70 LSETX$="2":Y$=STRING$(255,0):FORX=1TO255:MID$(Y$,X,1)=CHR$(X):NEX
  T:LSETZ$=Y$:PUT3:Y$=STRING$(255,0)
80 FORX=1TO256:GET1:NEXT
90 FORX=1TO2:GET1:LSETB$=A$:PUT2:NEXT
100 GET1:LSETB$=A$:PUT2:D$=A$
110 GET1:LSETB$="0":PUT2:P$=A$
120 GET1:LSETB$=A$:PUT2:P=VAL(A$)
130 FORX=1TO3:GET1:LSETB$=A$:PUT2:NEXT
140 GET1:LSETB$=A$:PUT2:T=ASC(A$):IFT=0 THEN 190 ELSE IFT<14)OR(T>15)
  THEN 200
150 GET1:LSETB$=A$:PUT2:IFT=15 THEN T=ASC(A$) ELSE T=ASC(A$)+128
160 T$="":FORX=1TO2:GET1:LSETB$=A$:PUT2:T$=T$+A$:NEXT:MID$(Y$,T-31
  ,1)=CHR$(VAL(T$)+64)
170 T$="":FORX=1TO2:GET1:LSETB$=A$:PUT2:T$=T$+A$:NEXT
180 IF VAL(T$)=0 THEN 140 ELSE FORX=1TO4*VAL(T$):GET1:LSETB$=A$:PUT2:N
  EXT:GOTO140
190 GET1:LSETB$=A$:PUT2:IF A$<>CHR$(0) THEN 260
200 IF(PAND1) THEN LSETB$=CHR$(27):PUT2:LSETB$=CHR$(61):PUT2:LSETB$=
  D$:PUT2
210 IF(PAND2) THEN LSETB$=CHR$(27):PUT2:LSETB$=CHR$(62):PUT2:LSETB$=
  D$:PUT2
220 IF(PAND4) THEN LSETB$=CHR$(27):PUT2:LSETB$=CHR$(63):PUT2:LSETB$=
  D$:PUT2
230 LSETB$=CHR$(27):PUT2:IF P$="N" THEN LSETB$=CHR$(18) ELSE IFF P$="E" TH
  EN LSETB$=CHR$(20) ELSE LSETB$=CHR$(17)
240 PUT2:IF MID$(Y$,96,1)=CHR$(0) THEN MID$(Y$,96,1)=LEFT$(Y$,1)
250 MID$(Y$,193,31)=STRING$(31,82):LSETZ$=Y$:PUT3:CLOSE:END
260 PRINT"INVALID CHARACTER DETECTED"

```

#### TRS-80 MODEL I EXPANSION INTERFACE WANTED

Preferably a later model that does NOT use the special buffered cable, but that DOES have a double density adapter and/or RS-232 board already installed. Does not necessarily have to be a Tandy-manufactured unit. Contact Ronald Schnurer, 770 West 4th Street Lot 38, Sault Ste. Marie, Michigan 49783 or telephone (906) 635-0858.

#### POWER SUPPLY AND/OR SCHEMATIC DIAGRAM WANTED

For a Zenith Data Systems Model ZTX-11 Digital Computer (this is a small keyboard unit about the same size as a TRS-80 Model I keyboard). Any technical information on this unit would be appreciated. Contact Ronald Schnurer (see above item).

NEWDOS/80 PDRIVE AND SYSTEM PARAMETERS  
by Kevin J. O'Hare

[This article originally appeared in a less complete form in the SYDTRUG NEWS (P.O. Box 297, Padstow, New South Wales 2211, Australia), and prior to that in the TRS-80 SYSTEM 80 Computer Group newsletter (16 Laver Street, MacGregor, Queensland 4109, Australia).]

Before we start I would like to point out that the memory map of where the parameters are stored and referenced for both PDRIVE and SYSTEM are different addresses in the Model III to the Model I. This is caused primarily by the fact that the DOS sector buffer has been relocated in the Model III to one sector higher in memory (4300H in the III to 4200H in the I). One of the points about this is that, while both the DOSes have similar addresses for the most part from 4400H up, the Model III has its communication area in a continuous block from 4000-42FFH but the Model I has two areas 4000-41FFH and 4300-43FFH with the DOS sector buffer in between. I guess Apparat wanted to tidy it up. However, this makes for a whole new set of parameter addresses for the Model III when compared with the Model I. At least the third sector (Relative sector 2) on each disk (PDRIVE + SYSTEM storage) has not been changed from the Model I to the Model III. The PDRIVE parameters are stored in this sector in the 00-9F section, (i.e. 10 x 16 byte slots), while the SYSTEM table uses the A0-FF area.

SYSTEM Parameters

The following is a summary of all the "SYSTEM" parameters of NEWDOS/80 Version 2. The parameters marked with one star are Model I only while the parameter with two stars is Model III only.

The parameters given in the table are as per disks supplied by Apparat. A full explanation of the parameters and their uses and functions are given in pp 2-47 and 2-49 of the NEWDOS manual and any may be changed to suit the individual requirements of the user with the "SYSTEM 0 ..." command.

En Passwords...AA=N	RUN-Only Mode...AB=N	* En Debounce...AC=Y
Enable "JKL"...AD=Y	Enable "123"...AE=Y	Enable "DFG"...AF=Y
En BREAK key...AG=N	* Lower case Mod...AI=N	Kbd Intercept...AJ=Y
* Drives...AL=4/4H	Reads/Error...AM=19/4H	DIR Drive No...AN=9/0H
Lowest Drive...AO=0/0H	HIMEN...AP=0/0H	En CLEAR key...AQ=Y
Copy No Pswrd...AR=N	* ForceBASIC U/C...AS=Y	Chain Input?...AT=Y
Kybrd Repeat...AU=Y	25ms Delays...AV=30/1EH	Writes/Error...AW=2/2H
Penrtr Hi Char...AX=99/5AH	Date Time Cold...AY=Y	DateTimeReset...AZ=N
ROUTE DO:NL...BA=N	** Clock 50/60Hz...BB=N	Chain Pause?...BC=Y
Override AUTO...BD=Y	R=Cad Repeat...BE=Y	* Force LCDVR.N...BF=N
Force L/C.Y...BG=N	Blink Cursor...BH=Y	Cursor Value...BI=0/0H
Time Loop Mult.BJ=1/1H	Enable WRDIRP...BK=Y	Verify withFMT.BM=Y
	* DIR read III/I...BN=N	

You will note that there are 35 active parameters for the Model I and 31 for the Model III. The places these parameters are stored/used from are quite varied according to the function. The parameter information is stored in words, bytes or bits depending on the requirement. The list shows where the parameters are stored in the lower part of Sector 2 of each disk.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A0	0402	0000	0000	0000	0A1E	5A01	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
B0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
C0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
D0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
E0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
F0	00F8	8000	0000	0000	7789	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

The Word storage is from D0 to EF

Byte	Param.	Function	Model 3 Storage	Model 1 Storage
D0-D1	AP	HIMEN value	42C7H and 4411H	43A9 and 4049H

The Byte storage is from A0 to CF

Byte	Param.	Function	Model 3 Storage	Model 1 Storage
A0	AL	Number of drives in system	42BFH and 4727H	439FH and 477AH
	AU	Number Write/Verify before error	445FH	44BAH
	AN	Default drive for DIR	42C0H	43A0H
A3	AO	Lowest drive new file creation	42C1H	43A1H
A4		Not used (maybe was AK)		
A5	BI	ASCII value cursor character	4023H	4501H

A6	AM	Number of reads before error	4405H	445AH
A7	AV	Number 25 ms delays before repeat	4534H and 4500H	4570H and 452FH
A8	AX	Highest Printer character	4290H	4370H
A9	BJ	Timing loop multiplier	42C2H and 4C94H	43A2H and 4CEFH

The Bit mapped storage is from F0 to FF.

The list below gives the bit involved, the function, and the memory storage. The first column holds the address (in Sector 2 of the disk) of the information. The next column contains the bit involved. The next column is the code letters and the function of the parameter, while the other columns give the actual memory address of the bytes involved.

To explain for the Model III/I, the byte at F0 goes to 428C/436CH, F1-428D/436DH, F2-428E/436EH, F3-428F/436FH, while the bits in F8 and F9 go to the code involved in various locations.

Byte	Bit	Param.	Function	Model 3 Storage	Model 1 Storage
F0	7	AA	Enable Password checking	428CH	436CH
F0	6	AB	Force RUN-ONLY condition AB=Y disables BC=Y	428CH	436CH
F0	5	AG	Enable Break key	428CH	436CH
F0	4	AI	* Lower Case Mod installed (Model I)	N/A	4593H and 4505H
F0	3,2	..	Not used		
F0	1	AR	Allow Copy without Password	428CH	436CH
F0	0	AS	* Force Basic text input to Uppercase	428CH	436CH
F1	7	AT	Allow Input requests from Chain	4280H	4360H
F1	6	BC	Enable Chaining pause or cancel	4280H	4360H
F1	5	BE	Enable "R" for repeat DOS Command	4280H	4360H
F1	4	BK	Enable WRDIRP & Dircheck W/C functions	4280H	4360H
F1	3	BM	Follow Format with Verify (Eliminated by Zap No. 44)		
F1	2-0	..	Not used		
F2	7	..	Used but for ????	428EH	436EH
F2	6-0	..	Not used	428EH	436EH
F3	7-0	..	Currently not used	420FH	436FH
F8	7	AC	* Enable keybounce - Model I only	N/A	4570H
F8	6	AD	Enable "JKL" - Screen print (change RET to RET,NZ)	4500H	45EEH
F8	5	AE	Enable "123" - DEBUG (in Interrupts set JR to JR,NZ)	4586H	4508H
F8	4	AF	Enable "DFG" - Mini-DOS (in Interrupts set JR to JR,NZ)	45AAH	45CCH
F8	3	..	Not used		
F8	2	AQ	Enable Clear key (in Keyboard driver set to 0)	4544H	4590H
F8	1	AJ	Enable DOS Keyboard Driver (4016H is loaded with Driver address)	4016H	4016H
F8	0	AU	Disabling can also be accomplished by pressing the Up arrow on Boot Enable Keyboard repeat (changed from JR to JR,NZ)	4513H	4538H
F9	7	AY	Require Date/Time cold - Tested at 4EC6H not stored		
F9	6	AZ	Require Date/Time reset - Tested at 4ECDH not stored		
F9	5	BA	ROUTE DO:NL - Reset Video DCB at 4E90H		
F9	4	BB	* Clock 50cps or 60cps - Set at 4EB0H		
F9	3	BD	Permit AUTO negate - Tested at 4F11H not stored		
F9	2	BF	* Force Lowercase Driver at reset - Model I only (changes C9 to C0)		45BAH
F9	1	BG	LCy at reset - Reset at 4E49H not stored		
F9	0	BH	Enable blink (Set by 4E4FH/4E6AH)	401CH	401CH
FA	7	BN	* DIR read by Model III NEWDOS/Model I TRSDOS - Used by SYS6		
FA	6-0	..	Not used		

[Editor's note: For those using NEWDOS/80 patched with Alan Johnstone's modifications, the additional SYSTEM parameters are stored as follows:

Parameter BZ uses the entire Byte BF.  
Other parameters (BR through CD) are stored in single bits of Bytes FE and FF:

Bit ---->	0	1	2	3	4	5	6	7
Byte FEH:					BR	BS	BT	BU
Byte FFH:	BY	BX	BW	CD	CC	BV	CB	CA

In any single bit parameter 0=N (disabled), 1=Y (enabled). This information was extracted from an article by Gary K. Bryce.]

**PDRIVE Parameters**

The disk used in this explanation is a Model III just as it came from Apparatus except for the PDRIVE specification lines (slots) 1 & 9 (Line 1 was the same as 0 and Line 9 was the usual configuration for an 8 inch drive). The original PDRIVE 0 screen below has 1 & 9 changed to accommodate an 80 track double sided drive (an exact copy of NEWDOS' own recommendation. See example 7 on Page 2-39 of the manual). A very full explanation of the parameters and their use is contained in the NEWDOS manual in section 2.37 on pages 2-33 to 2-39. Any of the parameters can be changed to suit your individual requirements using the PDRIVE 0 d .... command. A notation is added to each line below to simplify recognition.

Note :- Remember, NEWDOS retained compatibility with Single Density by having 5 Sectors per Granule (SPG) standard except if TI = M, when it changes to 3 to be able to read Double Density TRSDOS.

- 0# TI=A, TD=E, TC=40, SPT=18, TSR=3, GPL=2, DOSL=17, DPGA=2 40T, DO, SS
- 1# TI=A, TD=G, TC=80, SPT=36, TSR=2, GPL=8, DOSL=35, DPGA=2 80T, DO, DS
- 2# TI=A, TD=E, TC=40, SPT=18, TSR=3, GPL=2, DOSL=17, DPGA=2 40T, DO, SS
- 3# TI=A, TD=G, TC=40, SPT=36, TSR=3, GPL=4, DOSL=17, DPGA=3 40T, DO, DS
- 4 TI=AM, TD=E, TC=40, SPT=18, TSR=3, GPL=6, DOSL=17, DPGA=2 40T, DO, SS
- 5 TI=A, TD=A, TC=40, SPT=18, TSR=3, GPL=2, DOSL=17, DPGA=2 40T, DO, SS
- 6 TI=AK, TD=E, TC=35, SPT=18, TSR=3, GPL=2, DOSL=17, DPGA=2 35T, DO, SS
- 7 TI=A, TD=C, TC=80, SPT=20, TSR=2, GPL=2, DOSL=17, DPGA=2 80T, DO, DS
- 8 TI=D, TD=C, TC=80, SPT=20, TSR=3, GPL=2, DOSL=17, DPGA=2 80T, DO, DS
- 9 TI=A, TD=G, TC=80, SPT=36, TSR=2, GPL=8, DOSL=35, DPGA=2 80T, DO, DS

The above gives examples of both single and double density and should be enough to explain both Models I & III. The single density slots show a "+" next to them. For comparison, part of sector 2 of the same drive shows the following :-

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Tks	Den	Sid
00	1148	0728	1202	0001	1102	0000	0301	0004									40	Dbl	Sng
10	2348	0650	2408	0041	2302	0000	0201	0006									80	Dbl	Dbl
20	1148	0728	1202	0001	1102	0000	0301	0004									40	Dbl	Sng
30	1148	0728	2404	0041	1103	0000	0301	0006									40	Dbl	Dbl
40	1128	2728	1206	0011	1102	0000	0301	1004									40	Dbl	Sng
50	1128	0728	0402	0000	1102	0000	0301	0000	+								40	Sng	Sng
60	113F	4723	1202	0003	1102	0000	0301	0404									35	Dbl	Sng
70	11A0	0650	1402	0040	1102	0000	0201	0002	+								80	Sng	Dbl
80	11A0	0650	1402	0040	1102	0000	0300	0002	+								80	Sng	Dbl
90	2348	0650	2408	0041	2302	0000	0201	0006									80	Dbl	Dbl

These are the functions of the bytes in each 16 byte line :-

- 00 DOSL - Directory starting lump (Sector = DOSL \* SPG \* GPL)
- 01 Total number of lumps on the disk (Bytes used in GAT) = Total Sectors/Sectors per lump = (TC \* SPT) / (SPG \* GPL) e.g. DDens SSided 40Track has 720 Sectors and 72 or 40H Lumps
- 02 Bit mapped for Type Interface (bits 7-2) & TSR (bits 1-0)
  - Bit 7 - TI = H : Head settle delay
  - Bit 6 - TI = K : Track 0 formatted opposite density
  - Bit 5 - TI = M : TRSDOS 1.3 or TRSDOS 2.38 or higher on read
  - Bit 4 - TI = C or E : Percom or LNW
  - Bit 3 - TI = B or D : Omikron or Apparatus
  - Bit 2 - TI = A : Standard interface
  - Bit 1 - ) Track stepping
  - Bit 0 - ) rate (0, 1, 2 or 3)
- 03 TC - Number of tracks on the disk
- 04 SPT - Sectors per track
- 05 GPL - Granules per lump
- 06 Port F2 command only with Apparatus controller
- 07 Bit mapped TI & TD special conditions
  - Bit 7 - TI = H : 8 inch head settle
  - Bit 6 - TD = C, D, G, or H : Double sided
  - Bit 5 - not used [TI=N: Directory is full Track]
  - Bit 4 - TI = I : Sector 1 lowest each Track
  - Bit 3 - TI = J : Track 1 lowest Track
  - Bit 2 - TI = L : Double step
  - Bit 1 - TI = J or K : Track 0 not same
  - Bit 0 - TD = E, F, G, H : Double Density

- 08 DOSL - same as 00
- 09 DPGA - Directory granule allocation
- 0A Not used (See note # 1 below)
- 0B Not used (may be used for Sectors per Gran parameter)
- 0C TSR - Track stepping rate
- 0D Bit mapped TI - Type of Interface only
- 0E Bit mapped TI - Type of Interface Special conditions
- 0F TD - Type of drive (0-7 = ASC(A-H) minus 41H)

N.B.-The square brackets are Warwick Sands' new use.

The actual memory storage in the Model I is in the 43H page as follows :-

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4300	14FF	FFFF	FFFF	FFFF	FFFF	0001	1128	0728	0A02							
4310	0000	C930	4501	0007	00FF	FFFF	FFFF	FFFF	FFFF							
4320	FFFF															
4330	FFFF															
4340	0000	0000	0000	0000	0000	0000	0000	0000	0000							
4350	0000	0000	0000	0000	0000	0000	0000	0000	0000							
4360	0000	0000	0000	0000	0000	A540	0000	10F8	8000							
4370	5F11	2807	280A	0200	0011	0226	2807	2805								
4380	0200	0026	22FF	0100	0100	0000	00FF	00FF								
4390	0100	0100	0000	00FF	0071	4300	0000	0000								
43A0	0000	0100	001C	0053	55FF	FFA5	2814	0755								
43B0	0401	0000	00FF	FF00	0000	0000	FFFF	0000								
43C0	0000	00FF	FF00	0000	0000	FFFF	0000	0828								
43D0	0000	4200	00FF	0000	0500	4000	1400	FFFF								
43E0	0088	2000	501F	0063	1F00	0000	0000	1D20								
43F0	FFFF															

The memory storage in the Model III is in the 42H page

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4200	0500	00C3	3040	C3FA	35C3	FA35	4132	0332								
4210	2803	3C04	0000	0A36	03FF	00FF	0000	0000								
4220	0000	0000	FF52	0D00	FF00	FF00	FF00	FF00								
4230	FF00															
4240	FF00															
4250	FF00															
4260	FF00															
4270	FF00	FF00	7FA5	0B33	FF00	FF00	FF00	0081								
4280	1148	0728	1202	FE01	0740	2000	10F8	8000								
4290	5A11	4807	2812	02FE	0111	0223	4806	5024								
42A0	08FE	4123	0211	4807	2812	02FE	0111	0211								
42B0	4807	2824	04FE	4111	0391	4200	0000	0004								
42C0	0000	0100	0000	0052	4FFF	FFA5	FF00	FF00								
42D0	FF00	0000	0000	0000	00FF	FFA1	4C00	0000								
42E0	FFFF	0000	0000	00FF	FF00	0000	0000	FFFF								
42F0	0000	0000	00FF	FF00	0000	0000	FFFF	0000								

An explanation of the contents of these pages might be in order. NEQ means no equivalent. The square brackets indicate Warwick Sands' modifications.

Model III	Model I	Function
41FD-4202		Keyboard work area (4030H)
4203-05	4030	Jump for SYS1
4206-08	NEQ	Jump for RS232 Output
4209-0B	NEQ	Jump for RS232 Input
420C-0D	ROM	Address-Cassette activate routine
420E-0F	ROM	Address-Cassette routine to blink STAR at 3C3F
4210	4030	Control byte for Port EC
4211	NEQ	Cassette Baud rate flag
4212		Cassette counter byte
4213	NEQ	Control byte for Port E0
4214	NEQ	Video scroll protect # lines
4215		Not used
4216	4040	Count Interrupts -- Time/Date
4217-1C	4041-46	Time and Date storage
421D-23	NEQ	Route Device Control Block
4224	NEQ	Status byte CONTROL (shift down arrow)
4225-74	4318-67	DOS Command Buffer
4274	4367	Used for storage by DOS Modules
4275	4368	Separation byte (ASH) end DOS buffer
4276-7D	4300-07	Stores last track accessed for each drive. The DOS on accessing the Disk stores the previous Track/Lump accessed in this byte for its own use. Thus Drive 0 = 4276/4300, Drive 1 = 4277/4301, etc. This storage is for head travel so that if double stepping the number will be double.

427E 430B Current drive accessed (#-3)  
 427F 430F Port F4 command -  
 Bits #-3 - Drive in 1,2,4,8 format  
 Bits 4-7 - Sides, Write Compensation, Wait States, Density  
 Current PDRIVE specs being used

4280-87 430A-11 Break vector  
 7B-7A 4312-14 DEBUG active under TRSDOS  
 4315 unknown  
 4316  
 4288 4317 RST2B DIR value current SYS file + flags  
 Bit 7 - > Any of top 3 bits signify  
 Bit 6 - > current RST2B val below  
 Bit 5 - > invalidated.  
 Bit 4 - > Bits 4-# (value 00H-1FH) is  
 Bit 3 - > the RST2B value of SYS file  
 Bit 2 - > occupying the DOS overlay area  
 Bit 1 - > 4000H - 51FFH. This value is in  
 Bit 0 - > the 'A' register when loaded.  
 4289 4369 Bit mapped condition flags  
 Bit 7 - DEBUG activated  
 Enter DEBUG before executing CMD file  
 Bit 6 - Overlay area 4000-51FF used  
 disables DFG,JKL,123.  
 Bit 5 - Chaining in progress  
 i.e. get next input character from DO file.  
 Bit 4 - Break disabled by SYSTEM option.  
 Bit 3 - FORMAT or COPY resident  
 Input not from DO file  
 Bit 2 - RUN only  
 Bit 1 - [Alternate Keyboard active]  
 Bit 0 - [BASIC active]

428A-8B 436A Temporary storage TI specs from 0D & 0E drive #  
 428A Storage of conditions & overlays bits  
 Bit 7 - MiniDOS currently being accessed.  
 Bit 6 - DOS-call current  
 Bit 5 - DOS awaiting Keyboard Input  
 Bit 4 - Do not exit DOS-Call current level  
 Bit 3 -  
 Bit 2 - DOS fatal error  
 Bit 1 - DOS-Call level  
 Bit 0 - DOS-Call level  
 SYS1 zeros this byte.  
 Bit 7 - > Temp SYS module  
 Bit 6 - > bit storage.  
 Bit 5 - > Used for  
 Bit 4 - > FCB status.  
 Bit 3 - [Disable automatic DOS format recognition]  
 Bit 2 - DIR write  
 Bit 1 - Sector write  
 Bit 0 - used by SYS2- set when first extend  
 file space attempt fails (12.6.4)

428C 436C SYSTEM parameters from byte F0 in sector 2 of the disk  
 428D 436D SYSTEM parameters from byte F1 in sector 2 of the disk  
 428E 436E SYSTEM parameters from byte F2 in sector 2 of the disk  
 428F 436F SYSTEM parameters from byte F3 in sector 2 of the disk  
 4290 4370 Highest printable character for the printer  
 4291-9A 4371-7A 10 bytes of PDRIVE specs for drive 0  
 429B-AA 437B-8A 10 bytes of PDRIVE specs for drive 1  
 42A5-AE 4385-8E 10 bytes of PDRIVE specs for drive 2  
 42AF-98 438F-98 10 bytes of PDRIVE specs for drive 3  
 42B9-BA 4399-9A Address of current accessed PDRIVE specs  
 4291/429B/42A5/42AF or 4371/437B/4385/438F

428B-8C 439B-9C Used by Mini-DOS to store Stack Pointer address  
 428D-8E 439D-9E Used by SYS1 to store SP address under DOS-CALL  
 When exiting DOS-CALL (flagged by bit 6-428AH/436AH)  
 the Stack is reset to this stored value.

428F 439F Number of active drives in system (AL SYSTEM parameter)  
 42C0 43A0 Default drive for DIR (AN SYSTEM parameter)  
 42C1 43A1 Lowest drive for new file creation (AO SYSTEM parameter)  
 42C2 43A2 Timing loop multiplier (BJ SYSTEM parameter)  
 42C3-CA 43A3-AA Reserved for future SYSTEM parameters  
 42C5 43A5 Used by DO to store next input character  
 42C6 43A6 Unknown  
 42C7-CB 43A7-AB Contains the first 2 characters of last Command ("R")  
 42C9-CA 43A9-AA HIMEM value as computed by SYS0  
 42CB 43AB Warn boot check byte (AS)  
 42C-03 43AC-B1 DOS clock storage (6 bytes)  
 04-05 Unknown  
 42D4-FF 4382-CD ROUTE storage (Model III=6 bins, Model I=4 bins)  
 4480-9F 4480-9F DCB for loading files/sectors (PDRIVE, etc.)  
 44A0-BF 43E0-F1 DCB for chain files, etc.  
 44C0-D1 43CE-DF DCB for SYS modules and other files  
 43F2-FF Model I last two bins ROUTE storage

N.B.- NEWDOS is capable of handling six separate ROUTE commands simultaneously in both the Models I & III. The 42 bytes used for the storage are cut into 6 x 7 byte storage bins and hold the following information :-

Byte	Model III	Model I	Function
0,1	42D6-07	4382-83	Source device DCB address
2	42D8	4384	Type value of DCB being ROUTED
3,4	42D9-DA	4385-86	Pointer to next Route bin
5,6	42DB-DC	4387-88	Destination device DCB address

N.B.- 3800H in the Keyboard buffer set-up in Model 4 :-  
 Bit 0 - left Shift Bit 1 - right Shift  
 Bit 2 - CTRL (Control) Bit 3 - CAPS (up/low Case)  
 Bit 4 - F1 function key Bit 5 - F2 function key  
 Bit 6 - F3 function key Bit 7 - not used.

SOME NOTES (subheading :- the ramblings of a disturbed mind) :-

1. In the last sector of the PDRIVE module (SYS16) there is an unused set of two parameters for the use of the OA and OB bytes of each PDRIVE slot. They are LLD= and NLD=. I have not come up with any good explanation of what Apparatt had in mind for these two. Please let me know if any ideas that occur to anybody are likely to solve this puzzle.

2. The information above has real relevance to and probably infringes the Copyright of NEWDOS' owners. But I have yet to hear a squeak from them to say that the NEWDOS followers are being considered now that there is no more dough in it. If the problem arises I will face it.

3. Remember, NEWDOS retained compatibility with Single Density by having 5 Sectors per Granule (SPG) standard except if TI = M, when it changes to 3 to be able to read Double Density TRSDOS. I have a habit of theorising about what is best or ideal for a given situation. The single density has the advantage of everybody using a 5 sectors per gran and 2 gran per track (lump), thus any DOS can be read by any other.

But in double density, the 18 sectors per track has a problem of incompatibility when NEWDOS retained the 5 sectors per gran and the others used 3 or 6. However, in a double density disk of another DOS, if, in the PDRIVE slot, the number of the lump the DDSL points to is divisible by 18 then maybe, only just maybe, you can read the foreign Directory.

We have the TRSDOS directory on sector 306, which seems odd but is only 18\*17. We have LDOS using the centre track for the Directory so that sector 360 (lump (=track) 20 at 3 GPL and 6 SPG) is the start of the directory. To read this the NEWDOS would have to have the DDSL pointing to lump 36, which is reasonable when you calculate the following:-

$$\begin{aligned}
 40 \text{ Tracks} \times 18 \text{ sectors per track} &= 720 \text{ sectors per disk} \\
 \text{divided by } 5 \text{ sectors per gran} &= 144 \text{ gran per disk} \\
 \text{divided by } 2 \text{ granules per lump} &= 72 \text{ lumps per disk} \\
 \text{divided by } 2 \text{ half way thru the disk} &= 36 = \text{lump number for directory.}
 \end{aligned}$$

A simple matter to make you compatible with the other Directories. Yes, sure you can read a directory, but not access a file from the information contained in the directory, because of the SPG & GPL differences. Well, then we had a bright idea, just have a SPG parameter in the PDRIVE command and you have it made. Yes, you can read the other DOSes if you can only get a 6 or a 3 instead of a 5 for the SPG.

In SYS0 in the Model III, you have at 4C2DH and 4C58H code giving the value of 5 to the A register. This is the default sectors per gran parameter. If you can adjust this every time a PDRIVE slot is installed as the current one you have it. As you know the granules per lump parameter already is changeable in NEWDOS so you can have any value you like.

The routine at 4754H-475EH LDIR's the PDRIVE specs to be used from the storage area. Instead of three bytes (say the LD DE,4280H = 11 80 42) you put a call to the spare space at the end of SYS0 (4CAEH-end - Model III only) and install your routine there as well as the bytes you pinched to get there. But, I hear you say, where are the values stored for each drive? Another problem.

To overcome this one you do a clever trick. You use the high nybble of the GPL byte as it is never > 8. The table above shows the byte 05 or the 6th byte storing the GPL parameter. To install the new SPG value in byte 0B of the PDRIVE specs in sector 2 of the disk, we have the idea of using a base of 5 to

retain normalcy in the PDRIVE specs. That is if the parameter is 6, a 1 is stored there, and if a 5, a 0 is stored or in the case of a 3 an FE is stored in the byte OBH of each PDRIVE slot. This, after adding the necessary 5, is transferred in the Initialisation process (4D00-51FFH) on boot to the high nybble of the GPL byte (05) in each PDRIVE slot in memory.

The code to modify the transfer of each PDRIVE slot in the initialisation can be put at 50A8H which is pretty obvious at the bottom (B4) of the 2nd last sector of SYS0/SYS on the disk. Again some NEWDOS code, this time in the initialisation, can be overwritten and replaced in the new area to give access to the new code at the end of SYS0. Install code to transfer the low nybble of the SPG byte (OBH) to the GPL byte (05H). Then at the end of the 4C00H sector install some code to reverse the process; some code that can be accessed by the code that does the LDIR from the PDRIVE table (4291H-4\*10bytes-42B8H) to the active PDRIVE slot (4280H). This code would then load the two addresses (4C2DH, 4C58H) holding the SPG value.

You reckon I made it yet? No sir, I have not got it at all. Only the DOS routines which use the 4C00H mathematics, and this does not include SYS6 (Copy), are able to work properly. But a DIR on any foreign disk should be eminently capable. So good luck to you, I hope my ideas started you thinking.

\* \* \* \* \*

Kevin O'Hare is the author of the TRS-80 Hacker's Handbook.

### What? Me, Cheat? by John Pearce

[Reprinted from SYDTRUG NEWS, P.O. Box 297, Padstow, New South Wales 2211, AUSTRALIA.]

So this bloke looks me straight in the eye and says: "You mean to tell me that you BOUGHT a copy of the new Multidos, AND a new copy of the new Lazywriter? You bought them?"

I tell him that I did. I also wanted to tell him that, as a bloke who's written four books and countless articles for the world's magazines, I have sympathy for authors. But he wasn't listening.

"If you don't pirate stuff," he was continuing, "how come you bought that third drive -- AND the photocopy machine?"

I commence telling him that I use the third drive to call up sub-headings from different fonts when I write; and that the photocopy machine has other uses than copying computer documentation. But his eyes are all glazed over. He orders another drink, and shudders into it.

The word Pirate is no longer a piece of filth in the vocabulary. We are living in an age when governments are doing their best to emulate Robin Hood.

But my mind went back to that chap I met at lunch a few months ago. He told me that he was, himself, in the computer software business. "But," he added, "I have an IBM PC on my desk at the office, and another one at home; and there's no way that I would have bought duplicate software!" The way he read it, as long as he didn't cheat by selling software copies, or accepting software he hadn't bought, all would be okay.

I'm glad I write books and not computer programs!

For what hope do they have? "Professional Computing" magazine recently did an article about hackers and the law, and cited a couple of ridiculous examples.

In Hong Kong a home computer enthusiast "broke into" the Cable & Wireless network's "Dialcom Automated Office System", something similar to the British Prestel, from which our new Viatel is adapted.

But, when they came to prosecuting him, they discovered that they hadn't written any laws to cover it yet. All he was charged with was "stealing the electricity that illuminated the red indicator light on the C & W computer when he linked with the system!"

Meanwhile, back in Germany, a couple of hobbyists went on live television to give a demonstration. In a matter of moments, they "broke into" a local savings bank computer and transferred 37,000 pounds sterling to a dummy bank account.

So, where is the security? In the United States you can now buy a program called "Password" which has only one purpose: breaking passwords!

Goodbye, security. From now on, mine is the one house in the street with the blinds up!

### A LAZY MAN'S FIX FOR DOUBLEDUTY (or How I Got Around DDuty's Dislike For JCL Files) by Dave Bower [Compuserve 70635,330]

Double Duty is what you'd call a background program. It's not one that you would actually use by itself but rather one that you would use while using another program.

This means that everytime you load Double Duty you will have to load another program. And this presents two problems. First off, I'm lazy. I bought the computer to do things for me, not so I could do things for the computer. I like a one button turn-on procedure.

Unfortunately Double Duty does not provide a means to load another file, nor does it work with JCL files. It aborts every time.

The following patch allows Double Duty to pass a command string to TRSDOS for execution and passes control to this command instead of going back to TRSDOS Ready. Double Duty will still not run in a JCL file, but you can pass control from DDuty to a JCL file.

Patch to DDuty/Cmd  
by Dave Bower, (Compuserve 70635,330)  
March 1, 1986

For DoubleDuty version 02.05.00, by Software Concepts and sold by Radio Shack for the TRS-80 Model 4/4p

Allows DDuty to pass a command to TRSDOS for execution using the @CMNDI supervisor call.

Place our hook in DDuty

```
D04,CE=C3 DA A4  
F04,CE=3E 16 EF
```

Add code for @CMNDI Trsdos service call to end of DDuty

```
X'A4DA'=-3E 18 21 E0 A4 EF
```

This is the command string that will execute after DDuty has installed itself.

```
X'A4E0'="FILENAME/CMD" 0D
```

End of patch

## 64K \$59.95 PPD

INSTALLED IN KEYBOARD

### TRS-80\* Model I-LII

Send us your Keyboard and we will convert it to full 64K memory (48K RAM). Improved performance with or without Interface. 90 day warranty. Satisfaction guaranteed. Quick return. Free return freight within U.S.A.

**ICE**

International Carbide & Engineering, Inc.  
100 Mill St. • P.O. Box 216  
Drakes Branch, VA 23937  
(800) 424-3311  
(804) 568-3311 TWX: 910-997-8341  
\*TM TANDY CORP.

This month: Proposed AT&T rate changes, how CompuServe and MCI Mail users can exchange electronic mail, and other notes on MCI Mail services.

AT&T MAY INCREASE RATES FOR NIGHT-OWL COMMUNICATORS

AT&T has proposed a change to its long-distance rate structure that would change the discount for calls made during the night-weekend rate period from 60% to 55%. Or, to put it another way, the present cost of calls made during the night-weekend rate period would increase by 12.5%. This same increase would also be applied to WATS line rates, and in addition, the volume discount for more than 80 hours of WATS line usage per month would be eliminated. The night and weekend rate period is seen as the most unprofitable for all long distance companies, and some analysts believe that other major long distance companies (such as MCI and Sprint) will raise their rates by the same percentage.

My feeling on this is that as long as AT&T's competitors rush to get into the major metropolitan areas and ignore the rural areas, they will be forced to follow AT&T's lead on pricing. The reason is simple: They use AT&T's long distance service to complete calls to areas they don't serve, and they can't price calls below their costs (not for very long, anyway!). The interesting thing about this is that while the increases would apply to regular long distance service (which the general public uses) and to WATS lines (which the competing long distance companies use to complete their calls in rural areas), they would not apply to AT&T's Reach Out America plan, a discount plan that operates in the night-weekend rate period! So, suddenly, Reach Out America becomes more competitive with those other long distance companies, without lowering the price of that service. Whether the F.C.C. will approve this remains to be seen.

There are also reports that AT&T is planning to file a tariff on a service called Mininet, which would provide 800 number service for small businesses and residential customers. The difference between Mininet and the present 800 INWATS service offered by AT&T is that Mininet would use the customer's regular phone line for BOTH regular and 800 service calls. Thus there would be no need to have a second phone line and associated telephone equipment to have 800 number service. There would be no new service installation, and the per-minute rates would be lower than regular long distance rates. AT&T's competitors are expected to object to this offering on the grounds that the rates are not based on the costs of offering the service. They said that about Reach Out America when it was first offered, but that plan has been proven to be an income-producer for AT&T. So, it remains to be seen what will happen with all these new tariffs. One thing's for sure, AT&T has certainly become a lot more innovative in pricing strategies since the breakup of the Bell System!

COMPUERVE AND MCI MAIL ARE INTERCONNECTED

One of the problems with electronic mail systems is that if you use one system and the person you want to correspond with uses a different system, you're almost always out of luck. I say "almost always" because if both the system you use and the system that your correspondent uses offers Telex Dispatch capability, you can possibly send the message to the competing system as a Telex message. This is not by any means an ideal solution, for several reasons. One is that Telex messages are generally charged at a much higher rate than normal electronic mail messages, which makes this method prohibitively expensive for anything longer than a short note. Then, too, there are technical considerations that limit the usefulness of this method. For example, you are limited in the selection of ASCII characters that you may use in your message. You cannot use lowercase characters (actually, you can use them when creating your message on some systems, but they will be automatically converted to uppercase when sent via the Telex network). And, some electronic mail systems require a multiple-step access procedure that a competing electronic mail system might not be equipped to generate.

The ideal solution to this problem would be for a user of one electronic mail system to be able to send a message to a user of a competing system by specifying the proper address information. Well, it appears that two of the giants of electronic mail, CompuServe and MCI Mail, have done just that. It

is now possible for CompuServe users to exchange messages with MCI Mail users. The procedure for exchanging messages is fairly simple. Here's how it works:

MCI MAIL USERS can send messages to anyone who uses CompuServe's EasyPlex service or to companies who subscribe to CompuServe's InfoPlex service (if you're in doubt, most personal computer users are on the CompuServe EasyPlex service). The cost to send a message is the same as to send an MCI Mail message of comparable length. All you need to know to send a message is your recipient's CompuServe address. Your recipient can read the message for just the cost of the CompuServe connect time.

The first time that you correspond with a CompuServe user, you should give them your MCI Mail address, and either provide them with the instructions for sending a reply (which you can download from MCI Mail by typing HELP COMPUSERVE RECEIVE from the main menu), or instruct your recipient to type HELP MCI at the "Enter choice:" prompt in EasyPlex for instructions on how to send a message back to you.

To send a message to a CompuServe subscriber:

1. At Command, type CREATE as if you were creating an MCI Mail message.

2. At TO, type the correspondent's name and EMS in parentheses. For example:

TO: The Alternate Source (EMS)

3. At EMS, type CompuServe and press ENTER. For example:

Enter name of mail system.

EMS: CompuServe

EMS 288-7782 CompuServe CompuServe Info, Columbus, OH

4. At MBX, type the recipient's address as it is known on the CompuServe system. For example:

EasyPlex:

Enter recipient's mailbox information.

MBX: 72167,171

InfoPlex:

Enter recipient's mailbox information.

MBX: TAS:JJD [Just an example - we're not on InfoPlex!]

Verify the recipient address. Be sure that you typed it accurately so that your message is delivered properly.

5. At the next MBX, press ENTER to end this recipient's address.

If additional mailbox lines are not needed press RETURN.

MBX: [RETURN]

6. The address is displayed for your approval. Type YES if the address is accurate; or type NO if there is an error. For example:

TO: The Alternate Source

EMS: CompuServe / MCI ID: 288-7782

MBX: 72167,161

Is this address correct (Yes or No)? YES

7. When the next TO or CC appears, type the next recipient's name, if you're sending the same message to more than one recipient. If you're sending the same message to more than one CompuServe recipient, then each must be entered individually at TO or CC.

If you ever forget these instructions, you can simply type HELP COMPUSERVE from the main MENU to get the latest information.

COMPUERVE USERS sending messages to MCI Mail users should type HELP MCI at the "Enter choice:" prompt in EasyPlex, since there may be a couple of new wrinkles not covered here. Basically, though, all that needs to be done to send an EasyPlex electronic mail message to an MCI Mail user is to type >MCI MAIL: followed by the user's name or seven-digit MCI Mail User ID

SAMPLE SESSIONS: EMAIL BETWEEN COMPUSERVE AND MCI MAIL

Here's an actual session in sending mail from CompuServe to MCI Mail. The parts we typed in are underlined>. To get to this point, you type GO MAIL from any "I" prompt, which brings us to the .....

EasyPlex Main Menu

\*\*\* No mail waiting \*\*\*

- 2 COMPOSE a new message
- 3 UPLOAD a message
- 4 USE a file from PER area
- 5 ADDRESS Book
- 6 SET options

Enter choice !2

EasyPlex Compose

Enter message. (/EXIT when done)

- 1: This is a short test file to test the operation of the
- 2: CompuServe to MCI Mail
- 3: communications link for a NORTHERN BYTES article.
- 4: /EXIT

EasyPlex Send Menu

For current message

- 1 SEND
- 2 EDIT
- 3 TYPE
- 4 FILE DRAFT copy
- 5 SEND with /RECEIPT (\$)

Enter choice !1

EasyPlex

Send to (Name or User ID): >MCI MAIL 109-7407

Subject: TEST MESSAGE  
 Your name: Jack Decker  
 To: >MCI MAIL 109-7407  
 From: Jack Decker  
 Subj: TEST MESSAGE  
 Easyplex to MCI \$ .45  
 Is this correct (Y or N)? Y  
 Message sent to >MCI MAIL 109-7407

Press (CR) ! QEE

Thank you for using CompuServe!

Off at 07:04 EST 27-Feb-86  
Connect time = 0:04

Note that the cost of the message is displayed prior to the sender giving the go ahead - a nice touch. Here's how the received message looked on MCI Mail. Notice the approximate delay of 52 minutes before the message arrived in the MCI Mailbox:

Date: Thu Feb 27, 1986 7:56 am EST  
 From: Jack Decker  
 ENS: COMPUSERVE / MCI ID: 281-6320  
 MBX: [72167,161]  
 TO: \* Alternate Source / MCI ID: 109-7407  
 Subject: TEST MESSAGE  
 This is a short test file to test the operation of the CompuServe to MCI Mail communications link for a NORTHERN BYTES article.

Press (RETURN) to continue

Now we'll go in the opposite direction, from MCI Mail to CompuServe. We pick up the sequence here just after the display of the main menu, at the bottom of which we see .....

Command (or MENU or EXIT): CREATE  
 TO: The Alternate Source (ENS)  
 Enter name of mail system.

ENS: COMPUSERVE  
 ENS 281-6320 COMPUSERVE Columbus, OH

Enter recipient's mailbox information.

MBX: 72167,161  
 If additional mailbox lines are not needed press RETURN.

MBX: [ENTER only was pressed here]  
 TO: The Alternate Source  
 ENS: COMPUSERVE / MCI ID: 281-6320  
 MBX: 72167,161

Is this address correct (Yes or No)? Y  
 TO: [ENTER only was pressed here]  
 CC: [ENTER only was pressed here]  
 Subject: TEST MESSAGE  
 Text: (Enter text or transmit file. Type / on a line by itself to end.)

This is a short test file to test the operation of the MCI Mail to CompuServe communications link for a NORTHERN BYTES article.

You may enter:

- READ to review your letter
- READ PAPER to review your letter for paper
- EDIT to correct your letter
- SEND Postal delivery for paper; instant electronic delivery
- SEND ONITE OVERNIGHT courier for paper; PRIORITY electronic delivery
- SEND AHOUR FOUR-HOUR courier for paper; PRIORITY electronic delivery
- HELP for assistance

Command (or MENU or EXIT): SEND  
 One moment please; your message is being posted.

Your message was posted: Thu Feb 27, 1986 7:09 am EST.  
There is a copy in your OUTBOX.

Press (RETURN) to continue

On CompuServe, the received message appeared as shown below. Note the double display of message headers, one from CompuServe and one from MCI Mail. I suspect that the second header (the one sent by MCI Mail) is included in the total character count of the message for billing purposes. Thus, if you want to send a 45¢ "note" to a CompuServe user, you'd be wise to limit the actual text of your message to around 300-350 characters to avoid being inadvertently charged \$1.00 (for a full-fledged "instant letter" of over 500 characters). MCI Mail Telex Dispatch users may be familiar with this problem; it's nearly impossible to send any kind of a meaningful Telex message that is charged at the rate for only one "Telex minute", due to the copious amount of header information that MCI Mail adds to the beginning of all Telex Dispatch messages.

Once again, note that there is a delay in the message transfer, but in this direction it was only about 15 minutes:

Date: 27-Feb-86 07:24 EST  
 From: Alternate Source >MCI MAIL 109-7407  
 Subj: TEST MESSAGE

Date: Thu Feb 27, 1986 7:09 am EST  
 From: Alternate Source >MCI MAIL 109-7407  
 To: The Alternate Source >[72167,161]

This is a short test file to test the operation of the MCI Mail to CompuServe communications link for a NORTHERN BYTES article.

PLEASE NOTE that the times for message delivery shown here may or may not be typical. We weren't about to send dozens of electronic messages back and forth to test for "typical" delivery times!

number at the "Send to:" EasyPlex prompt. For example, >MCI MAIL:TAS or >MCI MAIL:109-7407 (it appears that a space character can be substituted for the colon) typed at the EasyPlex "Send to:" prompt will send the message to MCI Mail for delivery. CAUTION: MCI Mail allows duplicate User Names - for example, The Alternate Source and Texas Apparel Shop might both have the User Name TAS, but only The Alternate Source has the User ID number 109-7407. If you were using MCI Mail directly and typed a User Name assigned to more than one user, you'd receive a menu of users with that User Name and would be asked to pick the proper recipient from the list. But, obviously, this capability is not present when you're on CompuServe! The moral is to always use the MCI Mail User ID number (which is unique to each user) if you possibly can.

The ">" character is used to signify to EasyPlex that this is a remote electronic mail service delivery. Your EasyPlex address book can also be configured to accommodate mailings to MCI Mail. The cost to send an EasyPlex message to an MCI Mail user, in addition to CompuServe connect time rates, is 45 cents for 500 characters or less, \$1 for 501 to 7,500 characters and \$1 for each additional 7,500 characters. These are the same rates that MCI Mail users pay to send an "instant" electronic letter (but, of course, they do not have to pay for the CompuServe connect time).

CompuServe InfoPlex customers use a similar address format for MCI Mail messages (/Send >MCI MAIL:109-7407). Whether you use EasyPlex or InfoPlex, if your message cannot be delivered, you will be notified electronically. Check the address to be sure you used the correct format and verify the recipient's MCI Mail ID, then send your message again.

At present this link permits only electronic messages to be exchanged - a CompuServe user cannot access MCI's Telex Dispatch service, or send hardcopy (paper) mail via MCI Mail. For actual examples of how this service is used, see the box on the previous page.

#### SENDING TELEX MESSAGES TO MCI MAIL USERS

Some of our overseas correspondents have had some difficulty in sending Telex messages to an MCI Mailbox. Every MCI Mail user can receive Telex messages; the address is "650" plus the MCI Mail User ID number (minus the hyphen). Thus, MCI ID number 109-7407 becomes Telex number 6501097407. The answerback is the Telex number followed by a space and "MCI", thus the answerback in this example would be 6501097407 MCI.

No problem so far. But, what many Telex users fail to understand is that this number is on the MCI/WUI network. Thus, they have to use the MCI/WUI USA Access Code plus the MCI Mail "650" number. Access Codes vary by country but in many countries the proper code is "23", "023", or some other code containing the digits "23". This is not universal, though, as some countries have unique access codes. Canada is one such exception, their MUI/WUI access code is 06096.

If you are an MCI Mail subscriber, you can type HELP TELEX INBOUND at the main menu to obtain the proper codes to give to your foreign correspondents.

#### MCI MAIL ANNOUNCES ONLINE BULLETIN BOARDS

Ever wanted to run a bulletin board without having to tie up your computer 24 hours a day? Well, if all you want to do is disseminate information, MCI Mail now offers you the capability to set up your own bulletin board. This is not going to be inexpensive, mind you, but for certain business applications it may prove worthwhile.

Here's how it works. If you are an MCI Mail Advanced Service user (Advanced Service costs you \$10 per month, although if you're a regular MCI Mail user, there are discount plans that include advanced service), you can pay an additional \$25 per month, which gives you the capability of setting up as many as ten bulletin boards. You also pay 25¢ per 1000 characters that you upload to your bulletin boards. And, any message left on the system more than five days incurs a 1¢ per 1000 characters per day storage charge.

Starting to sound expensive? Well, now that you've got your pearls of wisdom online, somebody's got to pay to read them - at the rate of 30¢ per minute (this, of course, is in addition to any TYMNET or WATS line charges that may be applicable)!! I say "somebody" because you can elect to have a "viewer pay" system, where each person accessing your bulletin board gets nicked for the 30¢ per minute, or you can opt to pay for the viewing time

yourself. You might opt for the latter if, for example, you were distributing bulletins for your sales force in the field to read.

According to MCI Mail, you have total control over your bulletin boards. That includes who may read a bulletin board (you can restrict this to a small list of people, or make the board totally open so that any MCI Mail user may access it), the content of the items posted, when the item is posted and deleted, and the order of items listed. This also means that persons viewing your bulletin board cannot directly add anything to it - they have to send it to you as a regular MCI electronic message (for which they pay MCI) and then you can transfer the message to your bulletin board (for which YOU pay MCI).

If your bulletin board is open for any MCI Mail user to read, you can request a listing in the Bulletin Board Directory. The name of your bulletin board and other pertinent information will be posted. System users can see a directory of Bulletin Boards that are open to all users by typing HELP BULLETIN BOARD DIRECTORY (for more help on viewing a Bulletin Board you can type HELP VIEW).

Given the high cost of such a venture, why would anyone want to run such a system? MCI suggests that it can be used for publishing price changes, inventory updates, schedules, newsletters, job postings, and classified ads with one message posted in a central location. Should a change in the posted information become necessary, you can issue an update immediately. And don't forget that MCI Mail is now accessible in many foreign countries and via TYMNET in this country, thus making your gems of information immediately available all over the globe.

Yup, I noticed that MCI suggested that this service can be used for newsletters, but personally, I can't think of too many people who'd be willing to pay 30¢ per minute (plus communications charges in most areas) to read this rag - and who'd pay the upload, storage, and monthly charges? I think I'll pass.

By the way, even though I'm carping about the prices, some of the packet networks offer similar services, and their prices are just as high or higher - and you have to commit to a certain minimum volume per month. MCI Mail does have the advantage that anyone who wants to view your bulletin board can get onto the system by paying the standard \$18 annual mailbox fee. And, if a viewer fails to pay for his viewing time, you're not on the hook (as is the case in most other systems). For more information on this service, contact MCI Mail Customer Support at (800) 424-6677 in the U.S., or (202) 833-8484 elsewhere.

#### SUPREME SCROLL PROTECTOR PLUS by Mike Zarowitz

Recently, 80-MICRO has printed BASIC routines that access the scroll protection subroutine of TRSDOS 6.2.0. However, all of these routines are limited to 7 lines of protection and the protection is volatile, i.e. - all text on the screen is still erased with the CLS command. The following BASIC subroutine is what I call Supreme Scroll Protector Plus. It will support up to 22 non-volatile (not erased with CLS) lines (NVLINES%) starting at the top of the screen, and up to 7 volatile lines (VLINES%) of scroll protection appended to the non-volatile lines. The maximum number of protected lines is 22 and non-volatile lines have precedence - the command, NVLINES%=19:VLINES%=5:GOSUB 10000, will result in a change to 19 and 3 lines, respectively.

```
Supreme Scroll Protector Plus
10000 NVLINES%=NVLINES%-(NVLINES%<23)+22*(NVLINES%>22)
10010 VLINES%=(VLINES%-(VLINES%<8)+7*(VLINES%>7))*(NVLINES%+VLINES%<23)+(22-NVLINES%*(NVLINES%+VLINES%>22))
10020 CRTBGN%=&HF800+NVLINES%*80:LSB%=PEEK(VARPTR(CRTBGN%)):MSB%=-PEEK(VARPTR(CRTBGN%)+1)
10030 POKE &HC06,LSB%:POKE &HC07,MSB%:POKE &HCD4,LSB%:POKE &HCD5,MSB%
10040 CRTSIZE%=&H780-NVLINES%*80:POKE &HCD7,PEEK(VARPTR(CRTSIZE%)):POKE &HCD8,PEEK(VARPTR(CRTSIZE%)+1)
10050 POKE &HB94,(PEEK(&HB94) AND &HF8 OR VLINES%):RETURN
10060 'Scroll Protection Program. Version 1.0BAS
10070 'NVLINES% = CLS-independent scroll protected lines (0-22).
10080 'VLINES% = Regular scroll protected lines (0-7).
10090 'if nvlines%+vlines% > 22 then vlines% = 22-nvlines%.
```

## ALLWRITE HINTS

by Jack Decker

Here's a couple more hints for Allwrite users:

First of all, the Radio Shack DMP-2100P printer has an "IBM" character set that is not directly supported by Allwrite ("IBM" is a registered trademark of International Business Machines Corporation). However, it can be accessed by sending the proper control codes to the printer using the ";SY@" control word. For example, to switch to the "IBM" character set (which, by the way, differs from the "Tandy" character set only for ASCII codes 128 decimal and above), you can use the following Allwrite command:

```
;SY@ 27,58
```

You'd substitute the codes 27,59 to switch back to the "Tandy" mode. This works fine as long as you are using a monospace pitch, by if you are using a proportional font, you'll run into trouble because the character widths are not the same and the line will not right justify properly. There is a way to make Allwrite use the correct character widths, however, and the method might be useful for those with printers that can use downloadable fonts as well.

First, you must create a new character width table for Allwrite to use. To do this, COPY an existing TAB file (one of the ones that came on your master Allwrite disk, not one of the ones created when you ran the ALINSTAL/CMD program) to another file with a unique filename and a "/TAB" extension - for example:

```
COPY ORATORPR/TAB TO XYZ/TAB
```

Next, use a disk zapping program such as SUPERZAP and call up file relative sector zero of the file that you've just created. Ignore byte 00. Starting at byte 01 is a translation table for ASCII characters 1 through 255 (the pattern will be obvious when you actually look at the table). Look it over to see that all characters that your printer can handle have the proper byte in place this table, and if there are any characters that your printer does NOT want to see (such as ASCII characters 127 through 159 in the DMP-2100) you may change them to some other character such as a space. Actually, if you don't understand what I've just said, you can probably safely ignore the contents of sector 1.

Now we'll get down to business. Advance to File Relative Sector 1 (do not pass GO, do not collect \$200) and once again, ignore byte 00. Starting at byte 01, you will be putting in the widths of the characters. Byte 01 is the start of a table of widths for ASCII characters 32 through 255 (20H-FFH). In other words, byte 01 contains the width of the ASCII 32 (SPACE) character, byte 02 contains the width of the ASCII 33 ("") character, and so on. The widths themselves are the actual width PLUS 40H. Thus, if your printer manual (or whatever reference you are using) says that the width of a space character is 9 ("dot spaces" or "incremental units" or "points", or whatever terminology your printer manufacturer uses) then you'd zap a value of 49H (9 plus the 40H offset) into byte 01 of file relative sector 2. You'd place the width of the exclamation point character ("!") (plus 40H, of course) into byte 03, and so on. One note of caution - your printer manual will probably give the character widths in decimal, but your disk zap program will probably expect you to enter the values in hexadecimal. So, you'll need to convert the widths to hexadecimal, add 40H, and use that as the zap value.

If any characters are changed in the translation table back in sector zero, you should use the correct width for the translated character (the character that will actually be printed). In most cases, though, these characters can probably be ignored with some degree of safety since you most likely won't be trying to print one of the "translated" characters, anyway.

The DMP-2100P actually has two "IBM" character sets - one in Courier font and the other in Elite font. When you select the "IBM" character set, you get the one that corresponds to the "Tandy" character set that you were last using (in other words, if you were using the "Tandy" Elite proportional character set and you switched to the "IBM" character set, you'd automatically get the "IBM" Elite proportional character set). When you create your new /TAB file, be sure to use the correct widths for the character set size you want to use. If you want to use both the "IBM" Courier and Elite character sets, you will need to create two new /TAB files, one for each character set. The remainder of this article assumes that you are using the Elite character set (as activated by the ";PI @" control word).

Once you've created the new width table, here's how you use it. First of all, at the top of your Allwrite document (after any ";XT" control words that you may be using), use the ";WT" control word to inform Allwrite that you are going to use an external width table (see the Allwrite manual for an explanation of the ";WT" control word). Use a command similar to this:

```
;WI X,XYZ/TAB
```

(both "X" and "XYZ/TAB" may be different in your application. Again I encourage you to read the explanation of the ";WT" control word in the manual).

Then, when you wish to change to the "IBM" character set, you must use two Allwrite control words, as follows:

```
;PI X ;SY@ 27,58
```

The ";PI X" control word tells Allwrite to use the width table specified by the ";WT X,XYZ/TAB" command, while the ";SY" sequence actually tells the printer to switch to the "IBM" character set as specified above. When you want to return to the normal "Tandy" character set, you'd simply use the control words:

```
;PI @ ;SY@ 27,59
```

"PI @" tells Allwrite to use the regular "Tandy" proportional character set (you may wish to switch to some other pitch, in which case you should use the appropriate variation of the ";PI" control word) and the ";SY" sequence actually commands the printer to resume using the "Tandy" character set.

It should be noted that neither the ";PI" control words nor the ";SY@" control words cause a text break, thus the "Tandy" and "IBM" character sets can be mixed within the same line. If you want to force a text break, you'll have to add a ";BR" control word.

A final note of caution - character widths given in printer manuals should not be totally relied upon. They sometimes lie! Do some test printings and watch for justification errors, and make adjustments as necessary.

The credit for this technique goes to Chuck Tesler at Prosoft, who described it to me during a recent telephone conversation (I just hope that I got it all down properly!). Very few printers have alternate character sets as the DMP-2100P does, but this method could probably be used in any similar situation.

One thing that I have not figured out how to do is to use the "IBM" characters of the DMP-2100P below ASCII 32 (20H), nor those between 128 and 159 (80H-9FH). I don't think it is possible without entering the "IBM Emulation Mode", which would not be practical while printing a document (besides that, entering the "IBM Emulation Mode" seems to completely reset a part of the printer, even the "ON LINE" LED goes off momentarily while the printer is switching modes). You could possibly switch modes to print a full line using the "IBM" character set, but it would be very difficult to switch modes in the middle of a line. In case you are getting confused, it is possible to access the "IBM character set" (those characters greater than or equal to 32 decimal, anyway) without actually going into the "IBM Emulation Mode" (using the techniques described above).

Now, if you've muddled through the above, here's an Allwrite tip that everyone can use. Beware of the "WHOOOPS" command, especially if you have been typing in new text (as opposed to editing existing text). The use of "WHOOOPS" will undo all changes that have been made on the screen but not yet put into Allwrite's text buffer. Thus, if you've been entering new text, it's possible that you might delete the last screenful of text that you've entered by using "WHOOOPS". One way to avoid that is to first use the arrow keys to scroll all of the known "good" text off the screen before using "WHOOOPS". As each line scrolls off the screen, it is saved into Allwrite's text buffer. Be careful, though, that you don't scroll the portion of the text that you want to undo by using "WHOOOPS" off the screen, because once it's off the screen, it's saved and cannot be undone by "WHOOOPS". I lost a couple of screenfuls of text that I had typed before I figured this out!

### ATTENTION: COMPUTER CLUBS AND USER GROUPS

Now you can bulk distribute NORTHERN BYTES!! Your price per issue may be lower than the cost to photostat your club's exchange copy (and your support helps bring you future issues). And, you save \$\$\$ when your group buys ten or more copies of each issue. For further information, please contact The Alternate Source at (517) 482-8270.

**NEWMAP**  
by Kenneth B. Gash

I use NEWDOS/80 version 2.0 for most of my work and I find that it has most of the utilities that I need except for a Map program which gives the exact track and sector locations of all files on a diskette. When I want to repair a zapped sector or two, I find that I need some way to determine if the bad sectors are in important files or whether they can simply be written over or reformatted to repair them. After using the Superzap DFS (display file sectors) function for each file on a full double density diskette, I knew there had to be a better way to determine the location of all the file sectors and all of the free sectors on the diskette. What I needed was a directory mapper, similar to the one provided on DOSPLUS 3.4.

I had seen a program called DISKMAP written by Larry Sylvia (The Alternate Source Programmer's Journal, Volume 1, #2, pp 22-24) for use with NEWDOS+ and single density diskettes. The program was excellent as far as it went but it did not give sufficient information in a directly useable form for the more complex disk organization of double density (in the NEWDOS/80 2.0 configuration) so I set out to modify the program to make it more suitable to my needs. By the time I was done, I found that my program retained the original concept of Larry's program but the modifications were so extensive that I felt that it might be useful to others using this operating system. The program will read, analyze and display the results from the directory of any 5 or 8 inch, single or double-sided, single or double-density NEWDOS/80 2.0 diskette on the Model I, the Model III and the LNW computers. It will also probably work with single density diskettes in any TRSDOS compatible system although it may be necessary to apply a ZAP to one byte in the directory of those disks to make it work correctly.

**THE PROGRAM**

The program begins by reserving some string space, defining integer and string variables, designating some string variables to be used later, and disabling interrupts. When the program is run, you must have at least three file buffers specified (the default when BASIC is entered) since line 110 POKes a machine language routine directly into the file buffer area for file 3. This area of RAM is protected by NEWDOS/80 2.0 and the program will not interfere with any high-memory routines which may be present. The 72 byte USR routine is a bit search program which was adapted from a program called BITSRCH published in BASIC Faster and Better & Other Mysteries by Lewis Rosenfelder; it will be used to scan the bit-mapped sector information in the directory Granule Allocation Table (GAT).

In line 150, you are asked to type in the number (from 0 to 3) of the disk drive containing the diskette you want to map. The response is gotten through the INKEY\$ routine (line 770) and immediately checked for validity (0, 1, 2, or 3); if any other character (except the 'CLEAR' key) is depressed, the program will end. Since map listings can be somewhat long, the program assumes that you will want to have the output directed to the lineprinter. If you respond with a "Y" to the query in line 170, make sure that your printer is on-line and ready to go. If you respond with any other character (except the 'CLEAR' key), the ROUTE function of NEWDOS/80 2.0 will be invoked in line 200, sending all LPRINT output to the video display.

Line 210 causes the DIR/SYS file to be opened as a random access file. This will work fine on NEWDOS/80 2.0 but I have found that NEWDOS/80 version 1, NEWDOS+ and TRSDOS single density disks may have a false hash code for the DIR/SYS file in the HIT (hash-index-table) sector of the directory. You can check this by using SUPERZAP to look at DRS (disk-relative-sector) 171, byte 01. If you see a C4, this is the correct hash code. A 2C is false, preventing the opening of the DIR/SYS file. Simply ZAP the 2C to a C4 and everything will be fine from then on. The directory file is also read protected, so line 220 anticipates that an error will be generated when an attempt is made to read a directory sector. Line 730 checks to see if it was indeed the "correct" error, and if so, causes the program to resume as if no error occurred. The disk directory sectors are read even though an ERROR 58 is generated. If, in fact, no error was generated, the program assumes that this was caused by the false hash code mentioned above, prints a message to this effect and ends.

At line 230, the field specifications are set to the first 208 bytes of the physical record which of course is the entire GAT. Since NEWDOS/80 2.0 diskettes can be formatted in many ways up to and including 8 inch double-sided, double-density with a 6 granule directory, the entire GAT can be used for sector allocation information (this means that the lock-out table of TRSDOS may not be available). After the record is read (GET 1,1), the BOOT/SYS file of the diskette being mapped is opened, fielded and sector 2 (which is physical record number 3 of the file) is read into memory. This sector contains the PDRIVE information on NEWDOS/80 2.0 formatted diskettes. If a non-NEWDOS/80 2.0, TRSDOS compatible single-density diskette is in the drive, the program jumps to line 790 and requests certain information before proceeding. Lines 280 to 320 then print out the information about the diskette size, density, and number of tracks and sides. The number of "granules per lump" (GL) and the number of GAT entries (TG) are POKed into the appropriate places in the bit search program in line 280 while the location of the GAT data (D\$) is determined and this location POKed into the USR routine in line 330.

Lines 350 to 370 form a loop which repeatedly calls the USR bit search routine. Each time the routine is called, a jump instruction in the routine is toggled from JR C to JR NC by POKing the appropriate values (56 and 48 decimal) into the routine. This will cause the USR routine to first look for a bit which is set then look for the next bit which is not set and so forth. A set bit in the bit map indicates a sector which is allocated while a bit value of zero indicates a free sector. At the return from each USR call, the information is simply accumulated and printed. When a negative value is returned, the loop is exited and some summary information is printed.

During the printing of the initial information (just after responding to the question about printed output), it is possible to have some control over which files and which sectors are mapped. If the SPACE bar is held down while this information is being printed, you will be prompted for the number of the directory sector that you want the map to start on, and all files on that and subsequent sectors will be mapped. If the SPACE bar and the ENTER key are held down, you will be prompted for the directory sector number you wish to start on but no /SYS files will be mapped. If you hold down only the ENTER key, the mapping will start at the first directory sector containing FPDE's and map only the /SYS files.

Each directory sector after the HIT sector (2 to n, where n is determined by the number of granules assigned to the directory) contains up to eight 32 byte directory entries, either File Primary Directory Entries (FPDE) or File Extension Directory Entries (FXDE). Lines 430 to 450 of the program format the FIELD specifications for this information. The main portion of the program is contained in lines 480 to 700 where each of the directory entry sectors is read and the information in them is translated and displayed. Line 500 scans the keyboard to see if the 'CLEAR' key is depressed, and if it is, it ABORTS the remainder of the mapping, closes the file, clears the ROUTE function and returns to the start of the program. Line 510 looks at the first byte of each of the entries to see if the file is "dead" or active. Line 530 checks to see if the entry is an FPDE or an FXDE, and if it is an FXDE, it reprints the name of the file, labeling it as an FXDE.

Line 540 prints the name of the file and lines 550 to 560 calculate and print the number of sectors the file occupies on the diskette. Lines 570 to 670 scan each of the directory extent entries in turn to determine if it is an FPDE, an FXDE pointer or an end-of-extends (FFFF). In certain cases, such as when a file is opened but not written to, there are no sectors occupied by the file and no extents. If the entry is an FXDE pointer, and the FXDE is on another disk directory sector, that information is provided at this time. The nifty subroutine at 740 was in Larry's original program; it converts the ASCII value of the FPDE to ASCII HEX characters and prints the result.

To use the routine, it is simply necessary to hit a key from 0 to 3 in response to the DRIVE NUMBER prompt. The program assumes that there are 4 disk drives in the system. If you have less than 4 drives, change the '51' (ASCII 3) in line 160, as appropriate.

There are several places in the program where the user could determine additional or more precise information. For instance, in line 240, D\$ could be fielded to provide all of the information in relative sector 0 of the directory so the date and/or any AUTO commands can be extracted and printed. Another.

piece of information that may be useful would be the FRS (file relative sector) which can easily be calculated and printed. For further information on the structure of diskette directories, I strongly recommend H. C. Pennington's excellent book, TRS-80 DISK & OTHER MYSTERIES.

### PROGRAM LISTING

```
100 CLEAR2000:DEFINT A-Y:DEFSTRS:DIMF$(8,13),Q$(30,8):FORI=5TO8:S(I)=S
TRING$(I,32):NEXT:CMD"Y"
110 P=20949:DEFUSR=P+1:FORI=0TO71:READX:POKEI+P,X:NEXT
120 DATA0,56,69,105,79,205,127,10,17,0,229,235,35,94,35,86,213,221,225
,225,17,0,12,13,40,38,221,126,0,8,2,229,183,237,82,225,40
130 DATA8,19,31,16,245,221,35,24,233,31,48,18,35,16,250,221,35,13,40,7,2
21,126,0,8,8,24,238,33,255,255,195,154,10
140 H$="0123456789ABCDEF":CLS
150 PRINT"DRIVE NUMBER ----> ";
160 GOSUB770:ES=A$:IFASC(ES)<48ORASC(ES)>51THENCMD"R":CLEAR50:END
170 PRINT$:"PRINT"PRINTED OUTPUT (Y) or (N) ? ";
180 GOSUB770
190 IFA$="Y"PRINT"YES":CMD"ROUTE,PR,PR,DO":GOTO210
200 PRINT"NO":CMD"ROUTE,PR,DO"
210 K=0:OPEN"R",1,"DIR/SYS:"+E$
220 ONERRORGOTO730
230 FIELD1,200 AS D$,8 AS D1$:GET 1,1:IFK=0THEN780
240 ONERRORGOTO0:IFA$<>"Y"THENCLS
250 OPEN"R",2,"BOOT/SYS:"+E$
260 FIELD2,1 AS D0$,2 AS DU$,1 AS SC,1 AS SS,1 AS SL,2 AS DU$,1 AS
D0$,1 AS SG,3 AS DU$,2 AS SL,1 AS SD
270 GET2,3:CLOSE2:TC=ASC(SC):TS=ASC(SS):GL=ASC(SL):DG=ASC(SG)*5:TD=A
SC(SD):IF((ASC(D0$)<>0)AND(ASC(D0$)<41))AND(D0$=D0$)THEN280ELSE790
280 POKEP+32,GL:POKEP+63,GL:TG=(TC*TS)/(5*GL)+1.9:POKEP,TG
290 LPRINT" ":LPRINT"MAP OF DIRECTORY ON DRIVE "E$:LPRINT" "SS="5
.25"+CHR$(34):IF(TDAND1),SS="8"+CHR$(34)
300 LPRINTD1$ "--TC"TRACKS "TS"SECTORS/TRACK "LPRINTSS" ":I
FTD>3,LPRINT"DOUBLE":ELSELPRINT"SINGLE";
310 DE=2:LPRINT" DENSITY, ":IF(TDOR5)=7,LPRINT"DOUBLE":ELSELPRINT"S
INGLE":DE=1
320 LPRINT"-SIDED":LPRINT" ":LPRINT"THE FOLLOWING DISK SECTORS AR
E UNASSIGNED : "
330 PR=VARPTR(D$):Z1=PR/256:POKEP+10,INT(Z1):POKEP+9,256*(Z1-INT(Z1
))
340 Y=0:K=0:DZ=0:F=0
350 N=USR(Y):IFN<0THEN380ELSEIF K=0,LPRINT5*N"TO":K=1:POKEP+49,56
:DV=N:GOTO370
360 K=0:LPRINT5*N-1,POKEP+49,48:DZ=DZ+N-DY:F=F+1:IFA$="Y"ANDF=4,F
=0:LPRINT" "
370 Y=N+1:GOTO350
380 DS=3:LPRINT" ":IFDZ=0THENLPRINT"***** DISK IS FULL *****":LPI
NT" "ELSEIF(F>0)THENLPRINT" "
390 LPRINTDZ"FREE GRANULES ("INT(DZ/1000*256*50+.5)/10"Kbytes )
"GL"GRANULES PER LUMP "
400 LPRINT"DIRECTORY ("RIGHT$(STR$(DG),2)" SECTORS) BEGINS ON DISK
"RELATIVE SECTOR"5*GL*ASC(D0$):LPRINT" "
410 X=0:IFPEEK(14400)=128THENX=0ELSEIFPEEK(14400)=129THENX=1ELSE
IFPEEK(14400)=1THENX=2:GOTO430ELSEGOTO430
420 DS=2:IFPEEK(14400)=128THEN420ELSEPRINT:PRINT"WHICH DISK SECTO
R ( 2 TO"DG-1")TO START FROM "":INPUTDS:DS=DS+1
430 FORL=1TO8
440 FIELD1,(L-1)*32 AS D$,1 AS F$(L,1),2 AS D$,1 AS F$(L,2),1 AS F$(L
,3),0 AS F$(L,4),3 AS F$(L,5),2 AS F$(L,6),2 AS F$(L,7),2 AS F$(L,8),2
AS F$(L,9),2 AS F$(L,10),2 AS F$(L,11),2 AS F$(L,12),2 AS F$(L,13)
450 NEXT
460 ONERRORGOTO730
470 LPRINT"DSEC"TAB(10)"FILENAME"TAB(23)"SEC"TAB(32)"EXTENT"TAB
(40)"TRK"TAB(50)"DSK REL SECS"
480 FORM=0:STODG:GET1,M
490 FORF=1TO8:G=ASC(F$(F,1)):IF(G AND 64)<>0 AND (X-1) THEN700
500 IFINKEY$=CHR$(31)THEN710
510 IF(GAND16)=0THEN700ELSEIFX=2 AND (GAND64)=0 THEN700
520 LPRINTUSING"###";M-1:IF LEN(Q$(M-1,F-1))=0 THEN Q$(M-1,F-1)=STR
ING$(17,32)
530 IFG=144THENLPRINTQ$(M-1,F-1) (FXDE)"S(6)::B-V:GOTO570
540 LPRINTS(5)F$(F,4);"/";F$(F,5);
550 B1=ASC(F$(F,8)):B2=ASC(RIGHT$(F$(F,8),1)):B=256*B2+B1
560 LPRINTUSING"#####";B:LPRINTS(6);
570 C1=1:A=B-INT(B/5)*5:U(0)=A+((A-0)*5)-1:U(1)=4:FORGG=9TO13:IFGG<
13THENIFASC(F$(F,GG+1))=255,C1=0
580 CC$=F$(F,GG):C=ASC(LEFT$(CC$,1))
590 IFC=255ANDGG=0THENLPRINT"NO EXTENTS":GOTO700ELSEIFC=255THEN
700
```

```
600 IFC<254THEN630
610 V=B:C=ASC(RIGHT$(CC$,1)):J=(CAND15)+2+(CAND16):K=(CAND224)/32:IF
ASC(F$(F,4))>31THENQ$(J,K)=S(5)+F$(F,4)+"/"+F$(F,5)ELSEQ$(J,K)=STRING$
(17,32)
620 LPRINTTAB(33)**** FXDE ON DIR SECTOR":GOTO700
630 IFGG>9,LPRINTTAB(33)";
640 GOSUB750:LPRINT" ";
650 A1=C*GL*5:A2=(ASC(RIGHT$(CC$,1)))
660 FORR=(GL-1)TO1STEP-1:R1=32*R-1:IFA2>R1,A1=A1+5*R:A2=A2-R1-1:R
-1
670 NEXTR:LPRINTUSING"###";INT(A1/TS):LPRINT":":U=INT(((A1/TS)-INT(A
1/TS))*TS+.5):LPRINTRIGHT$(STR$(U),LEN(STR$(U))-1):LPRINTTAB(48)";
680 LPRINTUSING"###";A1:LPRINT" to ":LPRINTUSING"###";A1+A2*
5+U(C1)
690 NEXTTGG
700 NEXTF:LPRINTSTRING$(63,"-"):NEXTM
710 LPRINT:ONERRORGOTO0:CMD"ROUTE,CLEAR":CLOSE
720 GOTO150
730 K=1:IFERR/2+1=58THENRESUMENEXTTELSEONERRORGOTO0
740 LPRINTMID$(H$,(ASC(C$)AND240)/16)+1,1):MID$(H$,(ASC(C$)AND15)+1,
1):RETURN
750 C$=LEFT$(CC$,1):GOSUB740:C$=RIGHT$(CC$,1):GOSUB740
760 RETURN
770 A$=INKEY$:IF(A$="")THEN770ELSEIFASC(A$)-31THEN770ELSERETURN
780 LPRINT" ":LPRINT"**** CHECK 'HIT' ENTRY FOR DIR/SYS -- SHOULD
BE C4 (NOT 2C) *****":LPRINT" ":CLOSE:KILL"DIR/SYS:"+E$:GOTO710
790 TC=35:PRINT:PRINT"**** NOT A NEWDOS/80-2.0 DISK *****":PRINT:LI
NEINPUT"NUMBER OF TRACKS (DEFAULT=35) ":TS:IFTS<>"":TC=VAL(TS)
800 PRINT:TS=10:LINEINPUT"SECTORS PER TRACK (DEFAULT=10) ":TS:IFTS
<>"":TS=VAL(TS)
810 GL=2:DG=10:TD=0:PRINT:LINEINPUT"SINGLE <S> OR DOUBLE <D> SIDE
D DRIVE (DEFAULT=<S>) ":SX:IFSX="D",TD=2
820 D0$=CHR$(17):GOTO200
```

### TRSDOS 6.2 AND LDOS PATCHES

[Reprinted from the Adelaide Micro-User News, 36 Sturt Street, Adelaide, South AUSTRALIA 5000.]

#### TRSDOS 6.2 PATCHES - (we suggest using FED to do these):

1. Byte CE-CF in the first record of DIR/SYS should be E042 to delete the master password, and therefore the annoying messages that present themselves from time to time if one is there. Also true for LDOS 5.1.4.
2. To disable tab expansion: BASIC/CMD record 11 byte 41 change from 20 to 18.
3. To prevent clearing the screen between pages of directory: SYS6/SYS record 0A bytes from 5F change from 3E 69 EF 20 F8 to all zeros.
4. To disable checking for a password: SYS2/SYS.LSIDOS record 2 byte 33 change from 28 to 18.
5. To enable the KILL command (as well as REMOVE): SYS1/SYS record 2 byte 81 change 00 to 4B.

#### LDOS 5.1.4 PATCHES:

1. See 1 above.
2. To change the default name on format: FORMAT/CMD record 9, bytes 34-5 for Model 1 or bytes 2C-D for Model 3 change to 00 00; record 10, bytes from 0D for Model 1 or from 02 for Model 3 change the 8 characters as you wish.  
Although these patches have appeared before, it was some time ago, & some members have forgotten them. So here they are again.  
To remove password checking: SYS2/SYS Record 2, Model I byte 19 or Model III byte 37 change to 18.  
To scroll screen on subsequent pages of a directory: SYS6/SYS Record 8 byte E1 change to 00 00 00.  
To turn on the A parameter as default for a DIR: SYS6/SYS Record 5, byte D9 change to FF FF, to turn it off change to 00 00.  
To display characters >7F in FED: Record 12, byte 6B and Record 13, byte 96, change 3E 88 to 00 00. In FED2 it is Record 11, byte EA and Record 12, byte CE.

## CLUBS AND USER GROUPS

This is a list of the TRS-80 oriented and general interest computer clubs and user groups with which we exchange newsletters, plus four or five others that haven't been publishing a newsletter but have taken the trouble to let us know that they are still around (we've also included our address in this list, for reasons that should be obvious but are too numerous to explain here). If a club or user group does not appear on this list, it's most likely because we have heard nothing from them in the past three or four months. We have printed warnings that we were about to publish this list in the opening remarks of the last two issues of NORTHERN BYTES, so a club that has been receiving this newsletter and is NOT listed here doesn't have much excuse. The list is current as of March 1, 1986.

Again, we must point out that all exchange newsletters MUST be sent to NORTHERN BYTES, c/o Jack Decker, 1804 West 18th Street, Lot # 155, Sault Ste. Marie, Michigan 49783-1268. Sending them to any other address (such as TAS in Lansing) DOES NOT COUNT!!!!

This list is sorted within countries by zip or postal code order.

### U.S.A.:

Western Massachusetts Computer Club, 59 East Canton Circle, Springfield, Massachusetts 01104  
Connecticut Computer Society, Inc., P.O. Box 1841, Avon, Connecticut 06001  
Fairfield County TRS-80 Users Group, c/o Alan Abrahamson, 10 Richlee Road, Norwalk, Connecticut 06851  
BUG-80 Users Group, c/o 22 Alexander Avenue, Freehold, New Jersey 07728  
Central Jersey Computer Club, 41 Ferry Street, Lambertville, New Jersey 08530  
Amateur Computer Group of New Jersey, P.O. Box 319, South Bend Brook, New Jersey 08880-0319  
Metro TRS-80 Users Group, c/o G.F. Mueden, 310 West 106th Street (15D), New York, New York 10025  
Long Island Computer Association, P.O. Box 71, Hicksville, New York 11802  
Western New York TRS-80 User's Group, c/o Dr. A. Neil Yerkey, 67 Countryside Lane, Williamsville, New York 14221  
Rochester Area Microcomputer Society, P.O. Box 90808, Rochester, New York 14609  
Capitol Area TRS-80 Users Group, c/o Tim Sukay, 407 East Crestwood Drive, Camp Hill, Pennsylvania 17011  
Philadelphia Area Computer Society, P.O. Box 312, LaSalle University, Philadelphia, Pennsylvania 19141  
NCTCUG, Inc., P.O. Box 2826, Fairfax, Virginia 22031  
Richmond TRS-80 Users Group, c/o Merle D. Kerby, 13419 Oak Lane, Midlothian, Virginia 23113  
Greensboro Eighty Microcomputer Society (GEMS), P.O. Box 10014, Greensboro, North Carolina 27404  
TACS, Jason Torchinsky - Newsletter Editor, 4101 Redwine Drive, Greensboro, North Carolina 27410  
Midlands-80 Computer Club, P.O. Box 7594, Columbia, South Carolina 29202  
CSRA Computer Society, Post Office Box 284, Augusta, Georgia 30903  
Tallahassee Amateur Computer Society, Post Office Box 6716, Tallahassee, Florida 32314-6716  
Central Florida Computer Society, P.O. Box 8019, Maitland, Florida 32751  
Central Alabama Microcomputer Society, P.O. Box 17021, Montgomery, Alabama 36117  
Greater Gulf Coast Computer Club, c/o Gerald T. Regan, Spring Hill College, Mobile, Alabama 36608  
Toledo Area Tandy Special Interest Group, c/o A.F. Morel, Editor, 1632 South Wheeling Street, Oregon, Ohio 43616  
LDOS-Hard Drive SIG of CINTUG, c/o John W. Southcombe, 1184 Fay-Blan Road, Blanchester, Ohio 45107  
Cincinnati TRS-80 Users Group (CINTUG), P.O. Box 399145, Cincinnati, Ohio 45239  
Northeast Computer Club, P.O. Box 50252, Indianapolis, Indiana 46250-0405  
Dearborn TRS-80 Users Group, P.O. Box 1942, Dearborn, Michigan 48121  
Kalamazoo Area Computer Association, 7044 Leawood, Portage, Michigan 49081

Christian Computer Users Association, 1145 Alexander S.E., Grand Rapids, Michigan 49507  
Christian Computer Users Association TRS-80 SIG, c/o Bob Grommes - B & G Microsystems, 1733 Eastern S.E., Grand Rapids, Michigan 49507  
Northern Bytes, c/o Jack Decker, 1804 West 18th Street - Lot # 155, Sault Ste. Marie, Michigan 49783-1268  
Iowa City TRS-80 Users Group, P.O. Box 1494, Iowa City, Iowa 52244  
Milwaukee Area TRS-80 Users Group, 3822 North 75th Street, Milwaukee, Wisconsin 53216  
TRS-80 Users Group of Madison, 354 West Main Street, Madison, Wisconsin 53703  
Twin Cities TRS-80 User's Group, c/o Duane Stabler, 11226 Radisson Drive, Burnsville, Minnesota 55337  
Tandy Business Users Group, c/o Linda Hapner, 3329 B. Beacon #50, North Chicago, Illinois 60064  
Northern Illinois Computer Owners League, c/o Charles C. Hornaday, 26 W 484 Butterfield Road, Wheaton, Illinois 60187-7938  
Chicago Area Computer Hobbyist Exchange, Box C-176, 323 South Franklin, # 804, Chicago, Illinois 60606-7092  
SIG-80 Users Group of C.A.C.H.E., c/o George Matyaszek, President, 1718 North Long Avenue, Chicago, Illinois 60639  
Quad City Computer Club, c/o Cy Galley, 3318 26th Avenue, Rock Island, Illinois 61201  
North County 80-Group, P.O. Box 92, Hazelwood, Missouri 63042  
K.C. South Computer Club, c/o Richard Coleman, 2801 West 76th Street, Prairie Village, Kansas 66208  
Omaha Area TRS-80 Users Group, c/o Dan Shelby, Chairman, 7318 South 33rd Street, Omaha, Nebraska 68147  
LNU Computer News, Jay J. Hokanson, Editor, 4345 Manchester Road, Grand Island, Nebraska 68803  
The Green Country Computer Association, Inc., P.O. Box 2431, Tulsa, Oklahoma 74101  
Tulsa Computer Society, Post Office Box 3211, Tulsa, Oklahoma 74101  
Tandy User Group Newsletter, c/o Ed Juge, Director-Market Planning, 1700 One Tandy Center, Fort Worth, Texas 76102  
Corpus Christi and Bay Area Computer Users' Society, P.O. Box 8534, Corpus Christi, Texas 78412-0534  
Denver Amateur Computer Society, Box 477, Wheatridge, Colorado 80034-0477  
Waterton Computer Club (TRS-80 SIG), c/o Lowell Simons, 1150 Ridge Road, Littleton, Colorado 80120  
Club 80 Plus-Salt Lake, c/o Steven Baxter, 553 Spacrama Drive, Murray, Utah 84123-5749  
Valley Computer Club, P.O. Box 6545, Burbank, California 91510-6545  
San Gabriel Valley TRS-80 Users Group, c/o John T. Phillipp, 815 South Walnut, San Dimas, California 91773  
Orange County TRS-80 Users Group, P.O. Box 8074, Orange, California 92664  
North Orange County Computer Club, P.O. Box 3616, Orange, California 92665-0616  
Cabrillo Computer Society, P.O. Box 3032, Camarillo, California 93011  
South Bay TRS-80 Users Group, P.O. Box 60116, Sunnyvale, California 94088  
Bay Area Tandy Society, P.O. Box 23438, Pleasant Hill, California 94523  
Tandy TRS-80 User Group, c/o Del DelRosario, 430 Ilimano, Kailua, Hawaii 96734  
Northwest Computer Society, P.O. Box 75061, Seattle, Washington 98125-0061

### CANADA:

Micro-80 Computer Club of Ottawa, 150 Metcalfe Street Suite 409, Ottawa, Ontario, K2P 1P1  
NEWDOS/80 International Users Group, c/o Greg Small, Box 607, Stouffville, Ontario, L0H 1L0  
TRS-80 Users Group, c/o Norman Freidin, 2129 Larabee Court, Burlington, Ontario, L7P 3S3  
Mr. John Easton, Christian Computer/based Communications, 44 Delma Drive, Toronto, Ontario, M8W 4N6  
Winnipeg Micro-80 Users Group, c/o Keith R. Bradley, 208 1035 Beaverhill Boulevard, Winnipeg, Manitoba, R2J 3R2  
Regina Color Computer Club, 1112 College Avenue, Regina, Saskatchewan, S4P 1A8

**OUTSIDE THE U.S.A. & CANADA:**

- Sydney TRS-80 Users Group (SYDTRUG), P.O. Box 297, Padstow, New South Wales 2211, AUSTRALIA
- Canberra Micro-80 Users Group, c/o H.J. Cooper, 113 Owen Dixon Drive, Evatt, ACT 2617, AUSTRALIA
- Microcomputer Club of Melbourne, Inc., P.O. Box 60, Canterbury, Victoria 3126, AUSTRALIA
- TRS-80 System-80 Computer Group, c/o W.J. Allen, 16 Laver Street, Macgregor, Queensland 4109, AUSTRALIA
- Adelaide Micro-User News, 36 Sturt Street, Adelaide, South Australia 5000, AUSTRALIA
- Tandy Hobart Users Group, G.P.O. Box 1271 N, Hobart, Tasmania 7001, AUSTRALIA
- Compu-80, c/o Bert C. Guffens, Kasteelstraat 28, B-1800 Vilvoorde, BELGIUM
- TRS-80 Gebruikers Vereniging, Postbus 551, 2070 AN Santpoort-Nord, HOLLAND
- New Zealand TRS-80 Users Group Inc., P.O. Box 841, Auckland 1, NEW ZEALAND
- Christchurch 80 Users Group, Brendon Thompson, P.O. Box 4118, Christchurch 8000, NEW ZEALAND

**WARNING TO PURCHASERS OF TAS PUBLIC DOMAIN LIBRARY DISK #ND-1 (ALAN JOHNSTONE'S NEWDOS/80 MODS)**

Several early copies of TAS Public Domain Library Disk #ND-1 (Alan Johnstone's NEWDOS/80 mods) were sent out with a bad SYS8/ZAP file on the Model I side (this was our fault, not Alan's). If you use the DOS LIST command to LIST SYS8/ZAP and see a lot of "6m" or "6." characters at the start of the file, you have a bad file. If that is the case, you may use any word processing program that is capable of saving a file in pure ASCII format to create the correct SYS8/ZAP file, by simply (and carefully!) typing in the lines shown below, then saving them to the disk as SYS8/ZAP. Note that each CHANGE and TO line is a single line, but some of the lines won't fit as a single line on this page, so DON'T type a carriage return until you've reached the very end of each CHANGE or TO. If you don't think you can manage to do this, please send your disk back for a FREE upgrade (this is the ONLY file that will be changed). By the way, DON'T try to use the SYS8/ZAP file from the Model III side of the disk on a Model I, because the two files are NOT the same. We apologize for any inconvenience!

```

; SYS8/ZAP - Model 1
; This zap is part of Extended Disk Addressing.
; NOTE: This zap disables the 'DIR $' command.
ZAP SYS8/SYS 0 1F
CHANGE CD DB 4F
TO CD 18 51
ZAP SYS8/SYS 0 65
CHANGE CB F0
TO 00 00
ZAP SYS8/SYS 4 2C
CHANGE 4D 4F 55 4E 54 20 03 20 44 49 53 4B 45 54 54 45 20
4F 4E 20 44 52 49 56 45 20 30 20 20 28 45 4E
TO 3A A7 47 FE 08 CA DB 4F 3A 9F 43 3D B9 30 04 3E 20
B7 C9 C5 3E 90 81 26 43 6F 4E CD DB 4F C1 C9
;
; This ZAP is part of ADFD, and needs the above zap.
ZAP SYS8/SYS 0 E1
CHANGE CB 70 28 15 79 B7 20 05 3E
TO 18 17 C1 FE 1F D8 3E 1E C9
ZAP SYS8/SYS 3 3F
CHANGE 3A 1F 42 C6 08
TO E5 CD 38 51 E1
ZAP SYS8/SYS 4 4C
CHANGE 54 45 52 29 0D 53 59 53 54 45 4D 03 54 41 52 47 45
54 03
TO C5 DD 2A 99 43 DD 6E 09 3E 05 CD 92 4C 7D 3D 3D C3
DF 4D
END

```

Have you ever wanted to read and/or copy files off of a Model I single density diskette, but you were using a Model III or 4 and for whatever reason, you didn't want to WRDIRP the disk, but NEWDOS steadfastly refused to read it otherwise? Next time that happens, just run the machine language program shown below. This is one of our famous "the documentation is in the comment lines" programs. Just read the comments in the program listing for more information on how the program operates.

```

00100 ;THE FOLLOWING PATCH IS TO NEWDOS/80 (MODEL III)
00110 ;TO LET IT READ A MODEL I DISK WITHOUT DOING A
00120 ;WRDIRP ON THAT DISK (YOU MIGHT NOT WANT TO WRDIRP
00130 ;A MASTER PROGRAM DISK, FOR EXAMPLE)
00140
00150 ;THE PATCH SIMPLY IGNORES THE FACT THAT THE DIRECTORY
00160 ;IS PROTECTED WITH THE WRONG DATA ADDRESS MARKS.
00170 ;HOWEVER, THAT ALSO MEANS THE DOS MAY NOT BE ABLE TO TELL
00180 ;WHEN IT ISN'T REALLY ACCESSING A DIRECTORY SECTOR (ON
00190 ;ANY OF THE DISKS IN THE SYSTEM, NOT JUST THE ONE YOU ARE
00200 ;TRYING TO READ). THEREFORE, THIS PATCH SHOULD BE USED
00210 ;ONLY WITH EXTREME CAUTION (A HEALTHY DOSE OF FEAR AND
00220 ;TREMBLING WOULD BE APPROPRIATE, I THINK).
00230
00240 ;THE FOLLOWING WARNINGS SHOULD BE OBSERVED:
00250
00260 ;1. MAKE ABSOLUTELY CERTAIN THAT ALL DRIVES ARE
00270 ; CORRECTLY SET BEFORE INVOKING THIS PATCH
00280 ; (OTHERWISE YOU RISK OVERWRITING DATA WHEREVER
00290 ; THE DOS THINKS THE DIRECTORY OUGHT TO BE).
00300
00310 ;2. LEAVE THE PATCH IN PLACE ONLY LONG ENOUGH TO
00320 ; COPY THE PROGRAMS OR DATA FROM THE MODEL I DISK
00330 ; TO A MODEL III FORMAT DISK. THEN, PRESS RESET
00340 ; TO REBOOT THE SYSTEM.
00350
00360 ;THE PATCH RELOCATES ITSELF TO HIGH MEMORY AND PROTECTS
00370 ;ITSELF WHEN FIRST INVOKED. ALSO, IT IS INTENDED TO BE
00380 ;USED ONLY FOR READING MODEL I DISKS; NOT FOR WRITING TO
00390 ;THEM.
00400

```

```

00410 ORG 6000H ;JUST HERE TIL RELOCATED
00420 LD A,(54H) ;GET BYTE FROM ROM
00430 DEC A ;CHECK IF ON MODEL I
00440 RET Z ;BAIL OUT IF MODEL I
00450 LD HL,END ;END OF UNRELOCATED PGRM
00460 LD DE,(4411H) ;END OF UNPROTECTED MEM
00470 LD BC,END-PATCH+1 ;LENGTH OF MAIN PROGRAM
00480 LDDR ;MOVE THE PROGRAM
00490 LD (4411H),DE ;SAVE NEW HINEM POINTER
00500 INC DE ;DE=START RELOCATED PGRM
00510 LD (48ADH),DE ;PUT JUMP TO PATCH AT END
00520 LD A,0C3H ; OF DOS SUBROUTINE (C3H
00530 LD (48ACH),A ; + ADDR = JP PATCH)
00540 RET ;EXIT RELOCATOR ROUTINE
00550
00560 ;THIS IS THE ACTUAL PATCH CODE
00570
00580 PATCH CP 06H ;"PROTECTED" DIR SECTOR?
00590 RET Z ;BACK TO CALLER IF SO
00600 CP 31H ;WRONG RECORD TYPE ERROR?
00610 RET NZ ;BACK IF OTHER TYPE ERROR
00620 LD A,06H ;CHANGE ERROR CODE
00630 RET ;BACK (WITH Z FLAG SET)
00640
00650 END EQU $-1 ;USED BY RELOCATOR
00660
00670 END INTLZE
00000 TOTAL ERRORS
END 6020 INTLZE 6000 PATCH 6020

```

SOME SECRETS OF SUPERSCRIPSIT  
by Samuel Wells

Some of the following information I obtained by examining and disassembling SuperScrpsit with Tasmon; much of it was generated by Dr. John Van Wingen, presently teaching computer science somewhere in Florida. First a breakdown of the various SuperScrpsit program files and their loading points and functions:

FILENAME	LOAD	TO	COMMENTS
SCRIPSIT/CMD	5200	BAD2	MASTER PROGRAM
SCR16/CTL	8B2B	965E	TABS, HEADER/FOOTER MENU
SCR17/CTL	8B2B	9729	MAIN MENU, DIRECTORY
SCR18/CTL	8B2B	F1D7	UNKNOWN
SCR19/CTL	8B2B	8F27	UNKNOWN
SCR32/CTL	972B	A1FF	BLOCK ACTION MENU
SCR33/CTL	972B	A201	SPELL CHECK, SEARCHES
SCR34/CTL	972B	9E28	ASCII/COMPRESS UTILITIES
SCR35/CTL	972B	9BDO	SYSTEM SETUP MENU
SCR36/CTL	972B	9FDF	SYSTEM SETUP, CONT.
SCR50/CTL	EB00	EDDF	DICTIONARY, ERRORS
S/CTL	BAD3	BD80	UNKNOWN
DWP410/CTL	BAD3	BF24	PRINTER DRIVER (0000)
OTHER FILES:			
SYSTEM/CTL			SYSTEM SETUP, USER KEYS
ERRORS/CTL			
HELP/CTL			HELP SCREENS
WORDS/CTL			USER VOCABULARY?
MOVE/CTL			BLOCK MOVE FILE

SCRIPSIT/CMD loads SCR17/CTL and possibly sets up a stack at 8B0A. Necessary modules are overlaid when required.

The first six records of any SuperScrpsit generated text file are used for housekeeping; the text and imbedded printer codes, etc. begin at record seven. The first record contains the document variables such as margins, footers, etc. in the following scheme:

Record position	Byte	Remarks
00		SuperScrpsit file ID (E0 hex)
01-18		Filename; blanks = 84 hex
19		Line/page, each half line counted as one
1A		Printer pitch
1B		Line spacing, half line counted as one
1C-23		Printer type
24		Header start page--least significant byte
25		Header start page--most significant byte
26		Footer start page--least significant byte
27		Footer start page--most significant byte
28		Odd footer, number of half lines
29		Odd header, number of half lines
2A		Even footer, number of half lines
2B		Even header, number of half lines
2C		Cursor--start position of tab line
2D		Cursor--Screen line number
2E		Cursor--Tab line position
2F		Cursor--Document line#, least significant byte
30		Cursor--Document line#, most significant byte
31-45		Tab line 00
46-5A		Tab line 01
5B-71		Blocks In Use line--each bit signals a block present if 1, free if 0
72		Odd footer block#--default FF
73		Odd header block#--default FF
74		Even footer block#--default FF
75		Even header block#--default FF
76		Tab line block#--default FF
77		Total number of tab lines
78-97		Author name--blanks = 84 hex

98-B7 Operator name-- "  
B8-D7 Comments-- "  
D8-FF Filler

The second record consists of a linked list type directory which points to block numbers in sequence. I am not sure if a block = 1024 bytes or 768 bytes (3 or 4 records), or if block size is somehow varied. Experimentation should resolve this question. Record # 2 is thus configured:

Byte	Remarks
00	Total number of blocks of text
01	First Block number
02	Number of positions - 7 in last Logical record.
03	Number of logical records in block of text
04	Number of lines in block of text
05	Markers: Start text=80; Start block=20; End block=40
06	2nd block number (FF if no more blocks)
07-0A	Same as 02-05 if more blocks
0B-0F	Same as 06-0A...until last block

This scheme apparently continues over to next record (#3) if necessary

As one adds, inserts, or deletes text, the block numbers and pertinent info is changed. The text normally remains in its original record, but is detoured around by dropping it from the list if deleted, is pointed to if an insert. For this reason a heavily edited document with many block moves will be very difficult to read in sequence. The Compress utility of SuperScrpsit will correct this to a certain degree, but the best text straightener is to convert text to ASCII (which puts all ducks in a row) and then re-convert to SuperScrpsit (winding up with the shortest, most linear text sequence).

Since the destruction of any byte of the block descriptors will cause loss of sense in the program, some seemingly irrecoverable crashes can occur. Most loss of pointer problems give the error messages, "Attempt to read past end of file," "End of file reached," or result in a crash and reboot. On short, simple documents a diligent hacker can use Tasmon or Super3 or any other monitor (including TRSDOS Debug), can inspect the files and restore or reconstruct the block pointers. If all else fails use the BASIC program, SALVAGE/BAS, included. It kills the first six housekeeping records of the lost file, replaces any control code from 00 hex byte to 31 hex with a space (00 hex is read as end of file, and the other stuff garbages up the text), and writes an ASCII file named NEWFILE which is salvagable. Check to make sure that the last byte of the last record of the Newfile is 00 hex; if not make it so (SuperScrpsit demands to see an end of file marker). Convert the NEWFILE from an ASCII file to SuperScrpsit. With any luck you can edit and otherwise clean up your salvaged file. Remember, the more block moves and other editing changes made to the original the more jumbled will be your salvaged file.

Although this is only a scratch on the surface of SuperScrpsit, it might generate other's interests in how this rather complex program works. Several of us have been working on patches with only marginal success) to allow the reading of granules left on a disk along with the directory read. A major problem is that SuperScrpsit, loading at 5200, is overlaid by the regular directory call. And that reminds me of another hint: SuperScrpsit is rather unrelenting in its refusal to do anything once all disc space is used up, and is even more peevish if the TRSDOS extent limit is reached. The latter problem can crop up quite often if your file disc has fragmented extents of files scattered all it. Best bet here is to frequently clean up the disc with a copy by file type of backup. That way your file extents will stay down around 4 or 5 where they belong. To the best of my knowledge there is no recovery from an "Out of extents" error except to repair the pointers or strip the file back to ASCII.

Even though SuperScrpsit has its gremlins and shortcomings it is the word processor of choice among non-hacking professionals with Models III & 4 with a need to write long, professional looking documents and who desire their word processor to act as much like the friendly ol' typewriter as it

can. The sheer number of the program actively being used should be reason enough to master the intricacies of SuperScripsit. Once tamed its quite a workhorse.

P.S. I make no claims of expertise in my hacking -- I'm self-taught and highly pragmatic, and also quite prone to error. Any corrections to information appearing here would be more than welcome. Maybe several of us can eventually get it right.

[Readers wishing to contact author Samuel Wells can write to him at 530 Buschman Street, Hattiesburg, Mississippi 39401.]

```

20 CLS: CLEAR1000
40 PRINT "Enter name of file to be salvaged": INPUT F$
60 PRINT "Enter number of records in file": INPUT NR
80 OPEN "R", 2, "NEWFILE": FIELD 2, 255 AS B$
120 OPEN "R", 1, F$: FIELD 1, 255 AS A$
130 FOR X = 7 TO NR
140 GET 1,X
160 FOR Z=1 TO 255
180 IF MID$(A$,Z,1) < CHR$(32) THEN LSET
A$ = LEFT$(A$,Z-1) + CHR$(32) + RIGHT$(A$,255-Z)
200 IF MID$(A$,Z,1) > CHR$(122) THEN LSET
A$ = LEFT$(A$,Z-1) + CHR$(32) + RIGHT$(A$,255-Z)
220 NEXT Z
240 LSET B$ = A$: PUT 2,X-6
260 PRINT@320,"#X-6" of "NR-6" files corrected & transferred"
300 IFX=NRGOTO340
320 NEXT
340 LSET B$ = LEFT$(B$,254) + CHR$(00): PUT 2,NR-6
360 CLS: PRINT "All files transferred, please load NEWFILE into
SuperScripsit as an ASCII file and proceed to edit."
380 PRINT "On some operating systems it may be necessary to add a
00 hex ".PRINT" byte at the end of the last file in NEWFILE before
converting to SuperScripsit.
400 PRINT "Some editing of control codes will be necessary in the
new ".PRINT" SuperScripsit document."
420 CLOSE: END

```

#### USING TRAKCESS WITH DOUBLE SIDED DRIVES by Dave Owen

This article is reprinted from MICRO INFO EXCHANGE, newsletter of the Cabrillo Computer Society, P.O. Box 3032, Camarillo, California 93011.

In the beginning TRAKCESS, by Roxton Baker, was written for the Model I and was the epitome of innovative programming. For the first time a computer user could access tracks, sectors, and, yes, individual bytes on his diskette to modify, repair, or duplicate bits, bytes, sectors, tracks, or the entire diskette. The program has been updated by the author to include support for the Model III. But now there is the Model 4D. Following are patches to this program to support double sided diskette operation!

But first a word of caution. After examining the code, I am convinced that the author of TRAKCESS originally intended to include double sided capability. I believe he, in the original Model III release, did not support double sided operations because he could not properly keep track of all variables with the memory available in the Model III. Therefore, if you modify TRAKCESS for double sided examination, you need to follow one axiom: Complete all actions on one side before switching and performing actions on the other side. Do NOT switch from side to side to side to side...

I am modifying Version 3.0

```

Line 80 Add at the end: ".,SH!"
Line 110 Add after "DP=128": ".,SH=0"
Line 220 Change the PRINT@ to read:
PRINT@920,"X";:PRINT@933,"TO TOGGLE HEADS":
Line 270 Add a "+SH" after each "DS" (Two locations)
Line 580 Add a "+SH" after "DS+DP"
Line 650 Add a "+SH" after "DS" and "DS+DP"
Line 760 Add a "+SH" after "DQ+DP"
Line 850 Change to read "*HEAD TOGGLE*"
Line 860 SH=(SHAND15)OR(16-(SHAND16)):RETURN
Line 880 Add a "+SH" after "DS+DP"
Line 950 Add a "+SH" after "DS" and "DS+DP"
Line 980 Add a "+SH" after "DS" and "DS+DP"
Line 1010 Add a "+SH" after "DS" and "DS+DP"
Line 1180 Change "POKEDR,DS" to "OUTDR,DS+SH"
Line 1250 Add a "+SH" after "DS" and "DS+DP"
Line 1300 Add a "+SH" after "DS" (twice)

```

```

Line 1340 Add a "+SH" after "DS" (at the end)
Line 1370 Add a "+SH" after "DS" and "DS+DP"
Line 1440 Add a "+SH" after "DS" and "DS+DP"
Line 1690 Add a "+SH" after "DS" and "DS+DP"
Line 1740 Add a "+SH" after "DS" and "DS+DP"
Line 1750 Add a "+SH" after "DS" (at the end)
Line 3310 Add a "+SH" after "DS" and "DS+DP"
Line 3340 Add a "+SH" after "DS" and "DS+DP"
Line 3360 Add a "+SH" after "DS"
Line 3510 Change [part of this line] to read:
GOTO3250ELSEPRINT@76,"* DRIVE";DN;"HEAD":
PRINT@91,MID$("01", (SHAND16)/16+1,1):
PRINT " IS AT TRACK";ST(DN):PRINT@106,
Line 3600 Add a "+SH" after "DS" and "DS+DP"

```

[NORTHERN BYTES editor's note: In re-typing the above for publication, I discovered several apparent typographical errors. Therefore, don't assume that the above list is 100% correct (I may have missed a couple of errors). I particularly question the change of a POKE to an OUT in the change for line 1180 - unless this fixes a bug in the original program (which is doubtful but not impossible), the statement (POKE or OUT) probably should not be changed (although the "+SH" probably should be added in that line). Use common sense when making these changes and you should be okay.]

#### MODEL 4 SUPERSCRIPSIT ZAPS by Jim Whittaker

[Reprinted from SYDTRUG NEWS, P.O. Box 297, Padstow, New South Wales 2211, AUSTRALIA.]

I have just zapped my SuperScripsit (Model 4 version) to stop those infernal triangles appearing all over the screen when I press the space bar twice. The two zaps are for version 1.0 and for both the keyboard handler and the ASCII conversion. (SYDTRUG NEWS Editor's note: I have presented these patches in the original format that Jim supplied [the format used by LSFEDII] as well as the normal patch format, so that those who have not got the LSFEDII program or similar utility may still perform the patches.)

```

SCRIPSIT/CTL RECORD X'0014'
LOAD ADDRESS X'4302' CHANGE FE 20 20 TO FE 20 18
PATCH SCRIPSIT/CTL(D14,28=FE 20 18:F14,28=FE 20 20)

SCR35/CTL RECORD X'0001'
LOAD ADDRESS X'99C7' CHANGE FE 20 20 TO FE 20 18
PATCH SCR35/CTL(D01,30=FE 20 18:F01,30=FE 20 20)

```

#### MODEL 4 TIDBITS by James Abbassian

[Excerpted from Micro Notes, the publication of the Dearborn (Michigan) TRS-80 Users Group.]

The first tidbit was given to me by Wayne Chervekowaky. Anytime you use MEMDISK on a Model 4 you can change the initial delay factor to .5 seconds [instead of] 1.0 seconds with the command:

```
SYSTEM (DRIVE=n, DELAY=NO)
```

This will make an already fast MEMDISK even faster. Thanks, Wayne!!!

The second tidbit is for Model 4 owners with a Model 4 High Res Board. It seems that there are a few undocumented ports which seem to have been ignored, that is until now. Remember when you bought your high res board, there in the fine print read something like 32K of memory included. Well, it is used to hold all the pixels that are displayed on your screen. However, 640x240 resolution only consumes 18.5K that was all wasted [sic] until a few smart fellows who frequent CompuServe found out that you can use all 32K and achieve 1024x255 operation.

[Editor's note: I'm not sure how many NORTHERN BYTES readers use the High Res Board, so I won't explore this subject any further unless there is a demand for more information. In the meantime, Mr. Abbassian states that several files will be donated to their club's library explaining this in greater detail, with additional examples. If you wish to contact Mr. Abbassian, you can write him c/o Dearborn TRS-80 Users Group, P.O. Box 1942, Dearborn, Michigan 48121.]

# NEW!

# THE ALTERNATE SOURCE

704 North Pennsylvania Avenue  
 Lansing, Michigan, 48906-5319, USA  
 Phone: (517) 482-8270  
 MCI Mail ID: 109-7407  
 TELEX: 6501097407  
 Answerback: 6501097407 MCI  
 Compuserve EasyPlex: 72167,161  
 Delphi Mail: TASIO



Toll Free Ordering:  
 Michigan: 800-632-7818 Ex 700  
 Other US: 800-253 3200 Ex 700

### TURBO CHARGE

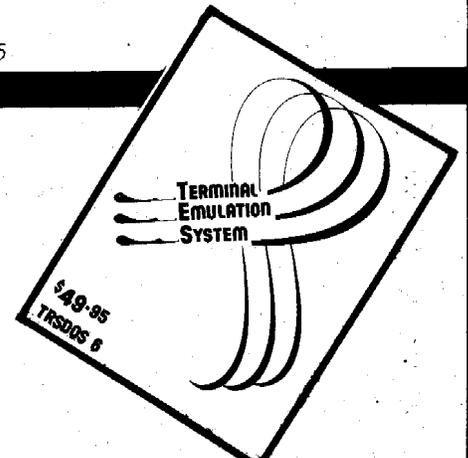
"Designed by an automotive engineer and guaranteed NOT to be RECALLED!" Bruce Hansen, author of our popular TASMON and ADVENTURE SYSTEM programs, is now an automotive engineer. Amidst an MSDOS world he has still found time to develop a superb package for the TRS-80 I, III(4)(P)(D). Turbo Charge displays the curving, winding roads of the racetrack. Glance in your rear-view mirror and you can see your opponents as they challenge your position or eat your dust. YOU control the shifting, the acceleration, the steering and the braking.



D8TURBO: Bruce Hansen's Turbo Charge ..... \$29.95

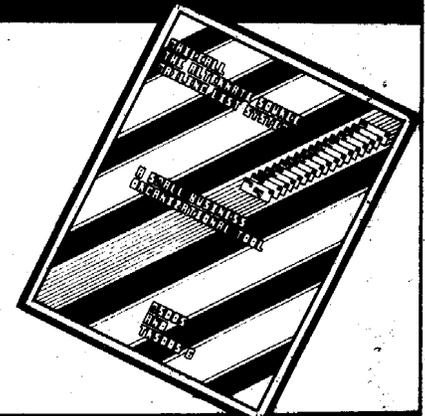
### TERMINAL EMULATION SYSTEM (TES)

Why invest in a dumb terminal when you can use a comparable priced TRS-80 Model 4 or 4P and have the power of a full computer available to your operators? Whether you are talking with a mainframe or a dumb terminal, TES will train your TRS-80 to properly respond to control codes from over one hundred different terminals! As the world of telecommunications expands, your TRS-80 can grow with it! TES includes a "terminal emulation editor for defining your own protocol and adding new ones (up to 300) as you meet them.



D4TES: Terminal Emulation System for TRSDOS 6 ..... \$79.95

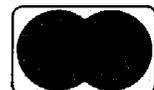
MAILCALL can manage one or more of your mailing lists with minimum intervention and maximum performance. MAILCALL supports names and addresses from around the world. You will no longer need to give your foreign customers second class service. Complete setup instructions and a friendly operator interface are included to get you organized fast. Once organized, you stay that way, because your master MAILCALL list is never sorted, yet you can instantly access your list in both name and zip code order. A conversion utility is included to assist you with transferring your current files to MAILCALL. Unprotected and hard disk compatible. MAILCALL also integrates with the optional TAS ORDER ENTRY SYSTEM.



DMMC: MAILCALL for MSDOS ..... \$79.95  
 D4MC: MAILCALL for TRSDOS 6 ..... \$79.95

USE THIS TO ORDER

Name: \_\_\_\_\_ Total \$: \_\_\_\_\_  
 Address: \_\_\_\_\_ Tax: \_\_\_\_\_  
 Address: \_\_\_\_\_ Shipping: \$3.00  
 Address: \_\_\_\_\_ COD: \_\_\_\_\_  
 Charge Card: \_\_\_\_\_ Expiry: \_\_\_\_\_ Total: \_\_\_\_\_  
 Special Instructions: \_\_\_\_\_  
 Please add \$3 shipping/handling per order; COD \$2 additional



# NORTHERN BYTES



## Subscription Information

Northern Bytes is edited by Jack Decker and published on an irregular basis by The Alternate Source Information Outlet. Back issues are available starting with Volume 5, Number 1. Issues prior to that are not available. Some of the most valuable articles from earlier issues may be reprinted in future issues of Northern Bytes. Currently there are eight back issues available for Volume 5, as well as all issues from Volume 6. All back issues are \$2 each.

It is very easy to be placed on the Northern Bytes REGULAR list. Simply place your address, Visa or MasterCard number and expiration date on file with us. We will start with the issue you request. We do not bill you for ANY ISSUE until that issue has been mailed. This way, we can continue to offer you top quality information with absolutely no risk to you. There's no question of what to

do about unfulfilled issues if we decide to quit publishing. Unless otherwise requested, we presume your subscription will extend through the month of your expiration date.

Don't have a charge card, huh? We understand the myriad of reasons for not having them and we feel that a "To-Be-Invoiced" policy could help increase the demand for Northern Bytes. If you'll do it for us, we'll do it for you. Would you like to be placed on a regular list TO BE BILLED for each issue? You could then send a check for the issues as they are mailed. If you didn't send a check, we would presume that your interest has died and discontinue your subscription. The only requirement for getting onto the list is to pay for the first issue up front; the next will be mailed automatically, if you request. You are insured that you will receive top of the line TRS-80 information as it is released. Ask to be placed on the NO RISK "To-Be-Invoiced Northern Bytes List".

Call or write, but SIGN UP TODAY!

The Alternate Source Information Outlet  
704 North Pennsylvania Avenue  
Lansing, Michigan 48906-5319

Telephone (517) 482-8270, or use our toll-free lines for ORDERS ONLY:  
(800) 632-7818 ext. 700 in Michigan, (800) 253-3200 ext. 700 elsewhere in U.S.A.  
EMAIL: CompuServe: 72167,161 / Delphi: TASIO / MCI Mail: 109-7407 / Telex: 6501097407 MCI

## NORTHERN BYTES

c/o Jack Decker  
1804 West 18th Street \* 155  
Sault Ste. Marie, Michigan 49783-1268  
U.S.A.

**POSTMASTER:** Address Correction Requested.  
Send address changes, USPS Form 3579, and undeliverable copies to: The Alternate Source Information Outlet,  
704 North Pennsylvania Avenue, Lansing, Michigan 48906-5319

Bulk Mail  
U.S. Postage Paid  
Permit 815  
Lansing, MI

To:

**ELECTRONIC MAIL ADDRESSES:**  
Compuserve EasyPlex: 72167,161  
Delphi Mail: TASIO  
MCI Mail: 109-7407  
Telex: 6501097407  
Answerback: 6501097407 MCI