TANDY LAPTOP COMPUTING MARCH 1989

TERRY KEPNER'S

portable 100

$3.95/CAN $4.95

A CLOSER LOOK AT DOS

First aid for *FLOPPY.*

TEXTSTAR

Wordstar cursor control.

A 1400LT Hard Drive

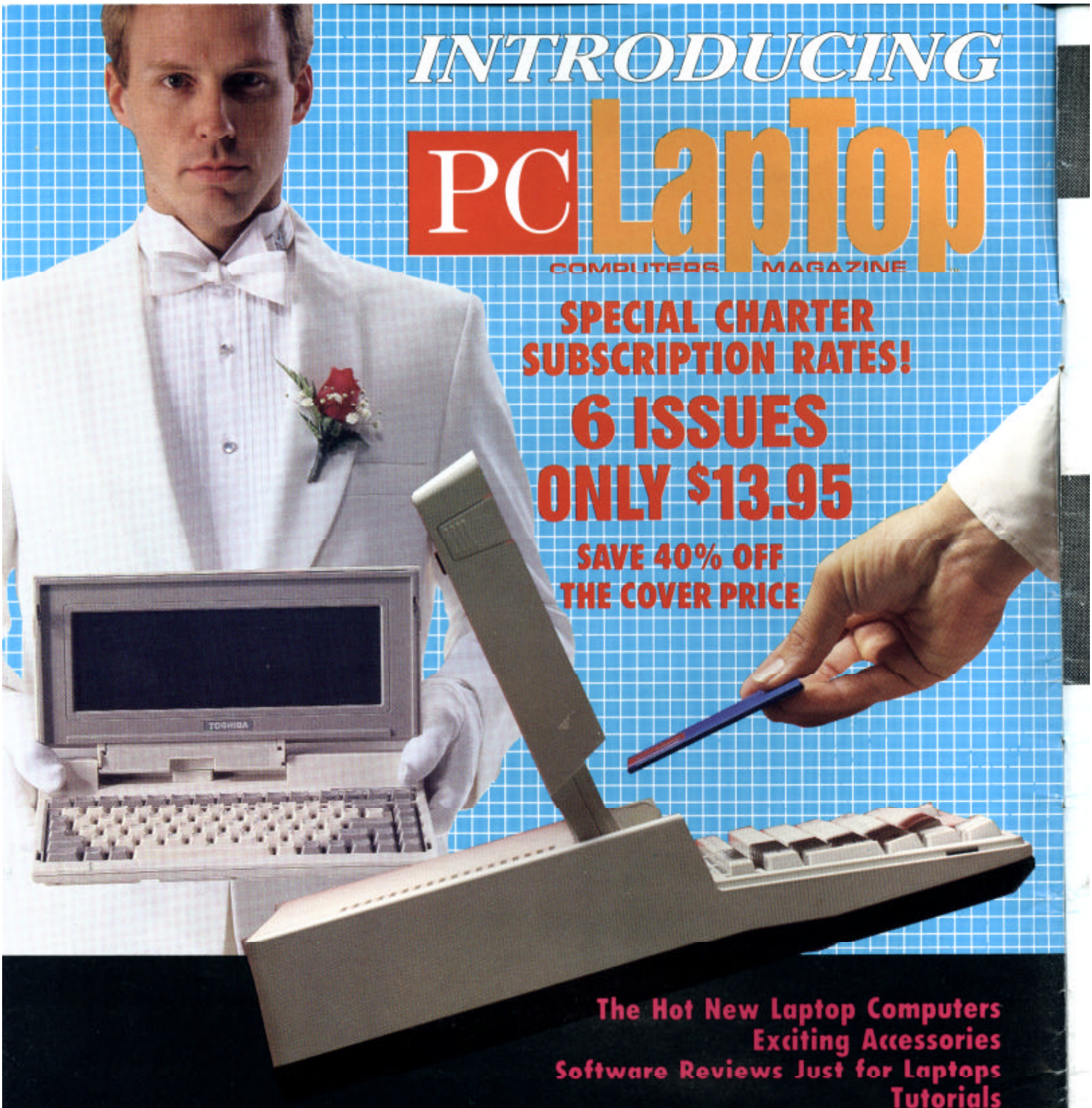
Five months later.

WEETAX

Software Review

Be prepared for next year.

Free programs inside!



The Hot New Laptop Computers
Exciting Accessories
Software Reviews Just for Laptops
Tutorials
News and Information
All MS-DOS Hardware and Software

PC LapTop COMPUTERS MAGAZINE

P.O. Box 16927,
N. Hollywood, CA 01615

CJQXV

☐ SIGN ME UP FOR 6 ISSUES AT THE SPECIAL CHARTER SUBSCRIPTION OF ONLY $13.95!

☐ PAYMENT ENCLOSED ☐ BILL ME CHARGE MY ☐ VISA ☐ MC

No._____ EXP_____ SIGNATURE_____

NAME_____

ADDRESS_____

CITY_____ STATE_____ ZIP_____

Money back on unused subscriptions if not completely satisfied. Foreign add $7. Make checks payable to LFP, INC., Your first issue will arrive in 6 to 8 weeks. Watch for it.

Circle 90 on reader service card.

ARTICLE INDEX!

With over 47 issues, and hundreds of articles, *Portable 100* is THE source for information, programs, and applications for your Tandy Portable Computer. But how do you find what you need among all those issues? The answer is the *Portable 100 Article Index*. Covering every issue from September 1983 to the combined July/August1988 issue, the index is designed to make it simple for you to find what you need. And at a cost of only $7.00 (postage and handling included), you should buy one today!

And what do you do when you find an article you want? Easy. First check the back issue chart below for back issue availability. If the issue is available, just order it using the handy order form in the index. If it isn't, don't despair. Photocopies of any article cost just $1.75 each. Simply list the article, or articles, by title, author, and issue, enclose payment, and in just a short time you'll have the articles in hand.

BACK ISSUES!

New low collector edition prices!

Month	83	84	85	86	87	88	89
January			NOT PUBLISHED				
February							
March			OUT		NOT PUBLISHED		
April		OUT					Not Available Yet
May		OUT					
June		OUT					
July		OUT		OUT		combined July/ August Issue	
August							
September	Premier Issue						
October		OUT		NOT PUBLISHED			
November							
December						OUT	

BACK ISSUES!

We've decided to inaugurate the new year by reducing the price of our collector edition, pre-1987 back issues to only $9.95 each, postage and handling included. The price of issues from August 1987 to present are $5.00 each, postage and handling included.

To help you decide which issues you want, we have a special, comprehensive 18-page article index covering every issue published from September, 1983 (premier issue) to the July/August 1988 issue. This index is available for only $7.00 (includes shipping & handling).

- ❑ Send me the Portable 100 Article Index for **$7.00** (P&H included).
- ❑ Please send me the back issues I've indicated above (pre-1987— $9.95 each; 8/87-on—$5.00each; shipping and handling charges included).

PAYMENT MUST ACCOMPANY ORDER

❑ CHECK ENCLOSED ❑ MC ❑ VISA

Total enclosed: _____ CARD# _____ EXP. DATE ____

Be sure to include entire card number

NAME _____

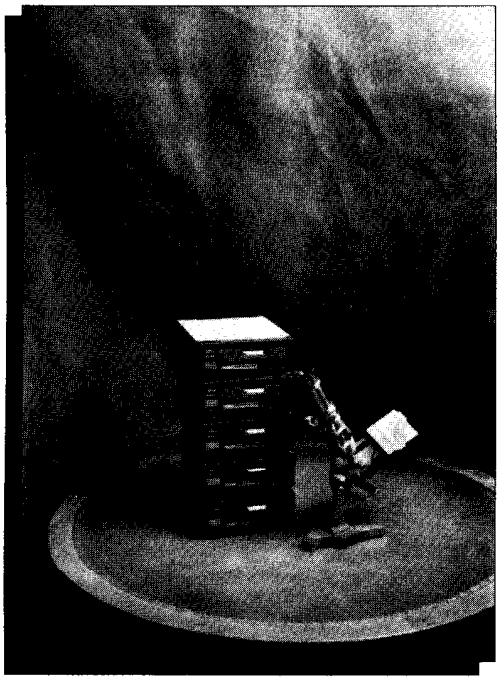
STREET ADDRESS _____

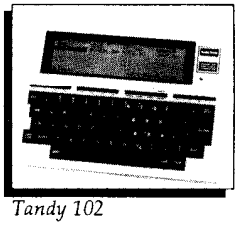
CITY _____ STATE _____ ZIP _____

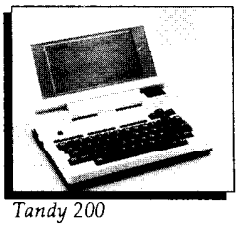
Please allow six to eight weeks for delivery

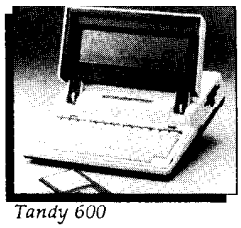
VOL. 6, NO. 3
March 1989

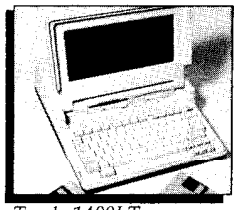
ON THE COVER:
This month's disk patches should be the final word on how how to make *FLOPPY* friendly to other programs. Photo by Richard Brayshaw, concept by Mike Nugent.




Tandy 102


Tandy 200


Tandy 600


Tandy 1400LT

DEPARTMENTS

ROM WITH A VIEW

Poetic License

CONTEST ANNOUNCEMENT! Portable 100, in coordination with Tandy Corporation, is having a contest open to EVERYONE, the grand prize is a Tandy Portable Disk Drive 2.

The contest itself is simple: Poetic License. That is, license plates that play on the portable computer industry (80C85, TPDD, M102, M100, M200, OLLYM10, NEC8201, NEC8300, and so forth). The rules are simple: All entries must be seven characters or less; valid characters include the uppercase English alphabet, the numerals zero (0) through nine (9), and the special characters #, $, +, -, and & (sorry, but none of the other characters are allowed or available on license plates in most states).

The winning plate will be judged by the staff here at Portable 100, based on originality and cleverness (relevance to Tandy computing is a plus). In case of a tie, the entry with the earliest mail postmark date will be declared the winner.

The Tandy Portable Disk Drive 2 (catalog #26-3814) works through the serial port with all the Tandy 100, 102, 200 computers, and the NEC 8201A/8300 and KC-85 clone computers.

Because this is the March issue, the deadline for entries is May 31st, 1988. Send all entries to Portable 100 Poetic License Contest, P.O. Box 428, Peterborough, NH 03458-0428. Good luck!

As you may have noticed, February Portable 100 was quite late in coming to you last month. Unfortunately, so is this issue, and so will the next few issues. When dealing with a printer, you usually have a scheduled "window" of time to deliver them the magazine. Because of the Christmas holidays the January issue was mailed late. As a result, the February issue was similarly delayed and missed its window (mailing the February issue on time would have meant both January and February arriving almost at the same time, cheating the January advertisers). When we delivered the February issue, instead of the normal twelve day turn-around time, we were told it would be twenty-one days, minimum. We hadn't planned on that. (Sigh.)

We plan to bring the issues up to their proper mailing times, but it will be done a few days at a time so that the advertisers will get their money's worth. Please be patient with us.

Other important notes: Last month lots of people called to say that their issue arrived without its customary wrap. Sorry. The person running the bindery machine at the printer made a mistake and dropped the wraps into the wrong position, putting the wraps in the middle of the magazine instead of on the exterior. The printer promised it won't happen again.

Similarly, some people received an issue stating that this was their last issue, even though it wasn't. That was an error at the mailing label house; they put the wrong labels on wrong magazines. (Or is it the right labels on the wrong magazines?) They, too, promise it won't happen again.

There wasn't a winner in our January contest, although we did receive some very interesting letters, including one fellow who not only matched names to faces but also tried to pair us up. Because we have an odd number of people, this necessitated one menage a trois! Fortunately (!) his selections and matches were incorrect.

Terry Kepner

Toolbox

Manuscripts were typed into Microsoft Word 3.0 on a Tandy 1400 LT, where they were edited, spell-checked, and had basic format instructions inserted. From there they were loaded into a Tandy 4000 (80386 CPU, Tandy EGA Monitor, Tandy LP-1000 LaserPrinter) desktop computer and placed into Aldus' IBM PageMaker 2.0a. There they were put into a rough approximation of the magazine's final appearance. Here, pull quotes are placed, headlines, intros, and bylines are sized and positioned, and advertisements positioned.

Next, the magazine was ported over to our Art Director's Macintosh II, using the 1400 LT and Mac-link. She then went over the publication using Aldus Macintosh PageMaker 3.01, making final design decisions on photo, figure, and listing sizes and placements. She precisely placed the text and added all the little things that go into making a nice looking publication.

Page previews were output from her Laserprinter. When everyone was satisfied with the appearance, the Macintosh disk was sent to Colorite Corp. in Wisconsin for final output directly onto photographic paper. The finished magazine was then delivered to the printer, who printed it, labeled it, and mailed it to you.

TERRY KEPNER'S **portable 100**

PRESIDENT/PUBLISHER
Terry Kepner

EDITOR-IN-CHIEF
Terry Kepner

FEATURES EDITOR
David Klein

TECHNICAL EDITOR
Mike Nugent

NEW PRODUCTS EDITOR
Linda Tiernan

CONTRIBUTING EDITORS
Paul Globman
Thomas L. Quindry

CIRCULATION MANAGER
Bob Liddil

ADVERTISING DIRECTOR
Terry Kepner

Portable Computing International Corporation
145 Grove St. Ext., #21, PO Box 428
Peterborough, NH 03458-0428

Editorial	Advertising
603-924-9455	603-924-9455
Circulation	**Bulletin Board**
603-924-9455	603-924-9770
	(300/1200—8,None,1)

Portable 100 (ISSN 0888 0131) is published by Portable Computing International Corporation, 145 Grove Street Ext., P.O. Box 428, Peterborough, NH 03458-0428. *Portable 100* is an independent journal not connected with any hardware, software, or peripheral equipment manufacturer. *Portable 100* is published monthly, except for a combined July/August issue in the summer. Entire contents Copyrighted 1988 by Portable Computing International Corporation, All Rights Reserved. No part of this publication may be reproduced without written permission from the publisher. Portable Computing International Corporation makes every effort to assure the accuracy of articles published in *Portable 100*, but assumes no responsibility for damages due to errors or omissions. Subscription Service: All subscription correspondence should be addressed to *Portable 100*, Portable Computing International Corporation, 145 Grove Street Ext. P.O. Box 428, Peterborough, NH 03458-0428. U.S. subscription rates: $24.97, one year; $45.97 two years. Canada and Mexico: US$27.97, one year; US$51.97 two years. All other foreign (surface mail): US$44.97, one year; US$85.97 two years. Foreign Air Mail, add US$50 per subscription year. All payment, except Canada, U.S. funds drawn on U.S. Bank. Second-class postage paid at Peterborough, NH 03458, and at additional mailing offices.

POSTMASTER: Send address changes to:
Portable 100, Portable Computing International Corporation, 145 Grove Street Ext., PO Box 428, Peterborough, NH 03458-0428.

CORNY EDITORIAL

I have always enjoyed the technical articles in your magazine. As a "normal" subscriber, I read each copy cover to cover the day that it arrives. That is, until I started to read my December copy. *ROM WITH A VIEW* reminded me of my two wonderful (yes, I know this is also corny) daughters. Well, I finished the rest of the issue *a few days later*. And now my 9-year-old, Sarah, is starting to learn *BASIC*.

Thanks for the corny editorial. They are needed once in a while.

Rick Sparber
Wheaton, IL

SEEKING BIOFEEDBACK

I am an occupational therapist in Maine, and much of my case load is comprised of individuals who have incurred a CVA or "stroke." Frequently, an individual will lose control of the arm and leg on the affected side. Through rehabilitation some measure of lower extremity control is often regained. Unfortunately, we often experience less success with the arm and hand.

One method of retraining muscle control which might work is biofeedback. Most of the equipment I've seen is in the $6,000 to $20,000 price range. I'm wondering if anyone knows of a method to turn the Model 102 into such a device. Output from the muscles is in the microvolt range, so an amplifier would be needed to bring the signal up to a level that the 102 could receive. Looking through some of my old issues of *Portable 100*, I came across an article on the DataMite from Jones Service and Design. I have purchased one of these and believe this will help me with the A/D conversion and software display.

I feel many individuals attempt to move their hands and arms, but when no motion is observed, soon give up. I believe that the 102 could become a valuable tool which could help make patients aware of very weak muscle contractions which are sometimes felt by the therapist's fingers. In this way the patients might become more efficient and successful in their attempts to retrain control of muscles which the stroke has incompletely taken away. I would greatly appreciate hearing from anyone who knows of a 102 being used in this way, or a reader who might be able to assist me in constructing the amplifier. Thank you.

Bill Croninger
RFD 1, Box 267
Alfred, ME

GIMME A CLUB!

I hear that there is a Tandy 100/102 user club in New York City but have not been able to locate them. Do you know how I can get in touch?

Stephen W. Lees
Forest Hills, NY

No, but I wish I did. We get many such requests, so we want to compile a user group directory to post on the Portable BBS, publish in a future issue, mail to those who request it, or maybe all three. Any readers who know of any clubs/user groups, please send us the details. Thanks!

-MN

DVORAK FOUND

Months ago, I ordered an NEC 8201A program on cassette tape to convert the standard keyboard to Dvorak. The company (HSI; Monroe, Michigan) was no longer able to supply the product.

Since then, I've checked GEnie's laptop bulletin board but found nothing there. Do you know where I can locate a similar program, either on cassette or printed out for typing into the NEC?

I enjoy your magazine, even though I'm *"BASIC"*ly ignorant and not a programmer. Every issue teaches me something—a good investment.

Gary Sorenson
Fargo, ND

Dvorak Keyboard System, *for Tandy and NEC notebook computers, is now distributed by Granite Street Portables (see their ad).* DKS *is now available on disk as well as tape. Reviews have appeared in* Online Today, PICO, *and the* Dvorak Developments *newsletter. Look for one in a future issue of* Portable 100.

-MN

ACOUSTIC COUPLER

The Datatronics *Discovery 2100 Acoustic Coupler* plugs directly into the RJ-11 modular jack of an internal (or external) modem. It works well at 1200 baud. This was a great breakthrough for me, as I had previously been limited to 300 baud by my Epson CX21 acoustic modem.

This acoustic coupler cost only $45.00 (US) in Hong Kong, and I'm sure it will be a hit with other portable users. I use it now with my Toshiba T1000 and its 1200 baud internal modem. It's made in Taiwan by Datatronics Technology; 172 Nan King E. Rd., Sec. 5; Taipei 10572, Taiwan. Phone (02) 762-3203.

Can your magazine cover products by other manufacturers?

Alex Fane
Tokyo, Japan

Thanks for the tip, Alex! It could be useful to 1400LT owners. Portable 100*'s focus is Tandy laptops, and since the NEC 8201/8300, Olivetti M10, and Kyotronic KC-85 are essentially Tandy clones, we try to cover them, too.*

-MN

DOCTOR'S ORDERS

MENUTL.CO is just what the doctor ordered. Two weird things with my 102: *CODE-V* doesn't make the invisible files reappear (*CODE-C* does), and when the power times out and the screen blanks, switching the power back on puts one in the *Rename* mode for *BASIC* (pressing *SHIFT/BREAK* restores the normal menu).

Using a P.G. Design *Vault* with *POWER-DOS* leads to a cold start if *POWER-DOS* is not turned off before accessing the *Vault*. With *MENUTL.CO*'s ability to make files invisible from the main menu, it is impossible to accidentally access the *Vault*. A lifesaver!

(From: Kent Hansen, London, Ontario)
David Tubman
Portable BBS
Message to SYSOP

Is this message from Tubman or Hansen or both? Either way, it sounds like you got a dirty download, maybe due to noisy phone lines. The program works as advertised; better download it again, or you might get a cold start someday. If possible, use the Xmodem protocol, which automatically corrects transmission errors.

-MN

QUIBBLE, RESPONSE, COMMENT

A quibble, a response, and a comment provoked by your most interesting January issue:

The quibble: Actually, *Powr-DOS* works satisfactorily with the second version of the Portable Disk Drive. Some of the features are disabled—you can't access the Bank 1 directory, and the direct sector access features create problems—but for most purposes it is satisfactory. You can still store 200K on the diskette, which is the main point.

The response: I'm not exactly certain what Dave Terwelp (Jan. '89 "DEFUSR") is asking for, but one or more of these may meet his needs:

Jesse Bob Overholt's *DBASE.BA* (*P100*, Mar. '84) has some possibly useful hints.

Rick Perry and I have posted several files on CompuServe's M100SIG which may directly address the question; Rick's is *LOGON.100*, and mine are variations on *REDIAL.**. I think they're in DL2 (Data Library 2), but I've stopped calling the SIG. Feel free to borrow anything useful.

The comment: I started putting copyright notices on certain CompuServe contributions when someone captured my review of *Powr-Disk*, did some minor editing, and sold it to an earlier version of this magazine. My object has never been to limit distribution (something I sometimes explain in the documentation); I just decided I needed to protect my work.

Phil Wheeler's right about the *XMODEM* program—it isn't Phil's to release to public domain. When J.R. Chenoweth first put the program on CIS, I was helping run a BBS and wanted the program; Chenoweth was very gracious about letting me copy it there. But I did ask and got a response. Phil's versions, as he says, merely enhance the Chenoweth code; if it was me, I'd certainly share his reservations about the copyright. Too bad, though, it's a terrific program.

I also agree with Thomas Quindry that those of us who post software on CompuServe and elsewhere could be more clear about our intentions. One of the problems is that I'm nearly always exhausted at the end of a programming project; I nearly always forget to write everything necessary into the documentation. For the record: Anything I've posted on CompuServe was intended to be public domain, and anyone who wishes to pass it to other users is certainly welcome to do so, provided only that I am credited with authorship and that the copyright notices not be removed from the distribution copies.

Finally: My sincere thanks to you folks for rescuing this magazine, and congratulations on your first anniversary as publishers.

Joel Dinda
Dimondale, MI

HE WHO LAUGHS LAST...

I would like to thank you and congratulate you on *Portable 100's* re-emergence and its quality. I'll have you know that I subscribed to *Portable 100* prior to purchasing the T200, and you convinced me even more of its usefulness and ease. It's lightweight, and the built-in programs make it ready to go.

Powr-DOS *works satisfactorily with the second version of the Portable Disk Drive.*

My wife, who teased me about getting it in the first place, now uses it for letters and craft and recipe ideas. She's even used a few hours of NiCd's playing the December T200 puzzle.

I would strongly urge you to include T200 changes in T102 programs, due to *CALL*'s, *PEEK*'s, and *POKE*'s, and I hope you continue to answer questions in your magazine. I haven't tried your BBS (I hope to soon), but I would hate to have it replace portions of your magazine. I would like to see a compatibility review of various products. I am also waiting for your article on the PDD-2, promised in an October 1987 article. The original article was great.

I enjoy getting *Portable 100* and look forward to every issue. I wish you well. Thanks, Mr. Kepner, Mark, and Nuge. Please don't stop.

Jurgan Almlie
Jordan, NY

Stop? No way—then we'd have to get real jobs! Don't worry, we'll keep answering questions in the magazine: PBBS *is just an extra service. As for the* PEEK's *and* POKE's, *we do it whenever possible, but some programs aren't easily converted, and we haven't time to rewrite 'em. If someone sends us converted programs, we'll try to put them on the* PBBS *and* P100-To-Go. *I assume the October '87 article you mentioned is Tom Quindry's review of PDD-1 DOS's. We're working on an update to that and planning another for PDD-2 DOS's.*

Since Mrs. Almlie's NiCd habit is probably tough on your budget, we've put the PUZZLE.BA *solution in this issue. But be warned—even with the solution, it's not easy!*

-MN

?IO—CORRECTIONS

Highlights of the "January 1989 Issue Screw-ups Competition":

In "CUSTOM 200," we mis-captioned Paul Globman's letter (Figure 2). The bar graphs were produced with Lucid, *not* MSPLAN.

Paul's KILL.BA *(same issue, p. 21) causes problems in some machines (see "DEFUSR"). The solution is to change* POKE AD,10 *in line 2 to* POKE AD,5.

Final score: Globman—1, P100—1. A tie!

-MN

We welcome all letters from our readers, whether critical or complimentary. We print as many letters as space permits (some are edited for space considerations). Address your correspondence to: Portable 100, I/O Dept., P.O. Box 428, Peterborough, NH 03458-0428.

COMPATIBILITY: Tandy 100/102, 200

DOS Patches III

"Stitch" in these improved patches for the PDD-2.

by Mike (The Tailor) Nugent

Good news—I was wrong! In earlier discussions of patching the disk operating system (DOS) for Tandy's Portable Disk Drive 2, I stated that, since the patched DOS cannot be saved as a file, you must re-patch each time you install the DOS. 'Tain't so! Troublemaker/genius Robert Holland, Sr., took me to task on that and showed how to save the DOS as a file. (See "DEFUSR")

Thus chastised and inspired, I've further refined his method and am happy to present the improved patches. Those who've requested more technical explanations, read on. Sane people can skip straight to the patches (see "PATCHING THE DOS" below).

WHY PATCH DOS?

The need arose when Tandy's first DOS, *FLOPPY.CO* for the original Portable Disk Drive (PDD-1), proved hostile to the kind of file containing machine language *packed* inside a *BASIC* program. Such a file must reside at a specific address in RAM—usually the very lowest address—to work. If it gets moved, it malfunctions, usually causing a cold start.

Unfortunately, *FLOPPY.CO* loads *.BA* files from disk into the lowest RAM address, pushing all existing files upward in memory to make room. If one of those pushed files is a packed file—WHAM! System crash.

If one of those pushed files is a packed file—WHAM!

No such problem with cassette, which loads *.BA* files into the "unsaved-*BASIC*-program" area just above existing *.BA* files. The computer maintains a *pointer* to this area to keep track of its position, and the cassette routine simply reads the pointer to find the program. Only files *above* this area (*.DO* and *.CO* files) are pushed upward to accommodate the new file. All existing *.BA* files, being *below* it, remain safely in place.

Why wasn't *FLOPPY.CO* designed to read that pointer and do the same thing? Who knows? It should have been.

So patches were developed to make the DOS do just that. You can save the patched code as a file named *NEWDOS.CO* (to indicate its modified state), and thereafter, you would use *NEWDOS.CO* instead of *FLOPPY.CO* as DOS.

With DOS thus tamed, the world was once again safe for packed programs like Micro Demon's *Supera*, P.G. Design's *MENU.BA*, Tri-Mike's *DVORAK*, etc.

BUT LATER ...

When Tandy introduced the Portable Disk Drive 2 (PDD-2), computers with packed files began going kablooey again. Had the PDD-2 DOS inherited its predecessor's genetic defect?

Precisely. Tandy had blown an opportunity to fix the flaw that made the old DOS hostile to packed programs. What they had not blown, however, was the opportunity to make the new DOS hostile to even *more* programs—including their own!

Was there some logical reason? Well, sort of. To understand it, let's examine the way the computer handles *.CO* files.

.CO DYNAMICS

All files are stored in low memory. Whereas *.BA* and *.DO* files operate right where they're stored, a machine language (*.CO*) file can operate only at the address for which it was written, usually somewhere in high memory. So the machine's designers had to provide a way to execute such *.CO* files.

Remember the term *pointer*? The variable *MAXRAM* is a pointer to the highest address that any program may use (Model 100—62960; Tandy 200—61104). Memory from there on up is reserved as a scratchpad where the computer can track essential data such as file locations, the contents of the LCD screen, and other "housekeeping" information, so the computer is necessarily quite protective of this area. In effect, *MAXRAM* acts as the "ceiling" of memory, carefully guarding what's in the "attic" above.

HIMEM, another pointer, can be set via the *CLEAR* statement to point anywhere from *MAXRAM* on down, acting as a "false ceiling" or "drop ceiling." Most programs normally are not allowed to use memory from *HIMEM* on up, so that area between *HIMEM* and *MAXRAM* provides a safe place for *.CO* files to operate.

Take *MYFILE.CO*, a hypothetical 2959-byte Model 100 program, written to execute at address 60000. The file actually occupies 2965 bytes (2959 bytes plus 6 bytes of *header* information) somewhere in low memory. It can't run there—it's only stored.

To run it, you'd first issue a *CLEAR 256, 60000* to set the *HIMEM* pointer to 60000. This reserves a safe place for the file to operate—between *HIMEM* (60000) and *MAXRAM* (62960). Then *LOADM "MYFILE"* puts a copy of *MYFILE.CO* (minus header)

into the reserved space.

Now two copies of the file exist—*MYFILE.CO* stored in low memory (and showing on the menu) and an executable copy in high memory. *CALL 60000* (its execution address) will now run the executable copy.

So you can see that running a *.CO* file consumes *twice* its normal space requirement. The file on the menu is strictly for storage, while the high memory copy executes. And therein lies the drawback of normal *.CO* files: It takes two to tango, but only one dances!

TANDY'S BETTER IDEA

Well, then, you can save space by having no menu copy. You can just keep an executable copy in high memory and *CALL* it whenever you want to run it, right?

Absolutely. That's what Tandy did. Putting only a tiny driver/checksum program (called *FLOPPY*) on the menu, it parks the executable DOS in high memory where it must be fiercely protected.

Setting the *HIMEM* pointer isn't enough. All manner of other programs change *HIMEM*, endangering the DOS. Some may even *LOADM* other *.CO* files right on top of it, corrupting it, with disastrous results when you *CALL* it later. It needs tougher measures.

The *MAXRAM* pointer, normally fixed, can be lowered artificially by means of special machine language programming voodoo (beyond the scope of this article). The PDD-2 DOS uses this voodoo.

THOU SHALT NOT

MAXRAM is somewhat sacred. Thou shalt not perform *CLEAR* with any address greater than *MAXRAM*; the computer won't let you. *CLEAR 256, MAXRAM*, a common statement in *BASIC* programs, can set *HIMEM* equal to—but never greater than—*MAXRAM*. So *CLEAR* statements can't disturb the DOS in the "attic" above the lowered *MAXRAM*.

But misguided *POKE*s can still pelt it, and *LOADM*s and *RUNM*s can still dump *.CO* files all over it. So there's another safety feature: *FLOPPY*, the driver/checksum program on the menu. When you run it, *FLOPPY* does a checksum on the DOS to detect corruption. If DOS is indeed farkled, *FLOPPY* refuses to run it.

NOT ALL ROSES

What seems a logical memory-saving scheme has some serious flaws. Any program needing DOS-occupied RAM is either refused admittance or corrupts the DOS, depending on its nature. Either case requires *FREMEM.BA* to kill *FLOPPY* and perform the necessary reverse voodoo to normalize *MAXRAM*.

In the first case (refused admittance) you must run *FREMEM.BA* before the offending program, then reboot the DOS afterwards. With the numerous programs that DOS finds offensive, this process becomes a tiresome pain in the neck.

In the second case (trashed DOS), *FREMEM.BA* must already be in RAM to clean up the mess before you reboot the DOS. Otherwise, you'll have to cold start, since the demented DOS can't load it from disk. Keeping *FREMEM.BA* in RAM is necessary for safety but tends to offset the intended memory savings.

Together, *FREMEM.BA* and *FLOPPY* use two directory slots and 411 bytes. *FLOPPY*'s counterpart in the DOS code, unnecessary in a normal *.CO* file, further offsets the perceived savings. And then there's the high memory permanently occupied by the DOS. Is it worth it?

GOOD OLD *FLOPPY.CO*

Actually, for the price of a little "wasted" memory—and only at run time—the more conventional *FLOPPY.CO* has a lot going for it. Foremost is flexibility. It's one file in one directory slot, easily saved to tape and disk or uploaded to other computers. It's unobtrusive when idle, doesn't tie up high memory, and requires no cold starts. And when patched, it coexists peacefully with most any program. Not a bad set of credentials.

PATCHING THE DOS

To turn the arrogant PDD-2 DOS into a nice guy, too, boot the DOS, type the appropriate line below, and press *ENTER*:

M100—*POKE 62116,154: POKE 62117,249: POKE 62118,0*
T200—*POKE 58799,149: POKE 58800,242: POKE 58801,0*

These patches make the DOS load *.BA* files into the unsaved-*BASIC*-program area, for compatibility with packed files.

We're creating a normal *.CO* file, where a fresh copy loads each time it's run. Corruption won't occur, so we can dispense with any checksum-related code (e.g., the first three bytes of DOS, at *MAXRAM*—we'll skip them).

When you save *NEWDOS*, you'll use a statement that resembles this: *SAVEM "NEWDOS", Top, End, Exe*, where you'll save from a *Top* address of *MAXRAM*+3 (M100—60003; T200—56675) to an *End* address one byte before *FLOPPY*'s load point (M100—62700; T200—59393). And when DOS is uncorrupted, *FLOPPY* jumps to *MAXRAM*+3 to execute the DOS; that's our *Exe* address (coincidentally, the same as *Top*).

With this, you're ready to save the code to a *.CO* file. Type the appropriate line below and press *ENTER*:

M100—Type *SAVEM "NEWDOS", 60003, 62700, 60003*
T200—Type *SAVEM "NEWDOS", 56675, 59393, 56675*

That creates *NEWDOS.CO*, weighing only 2704 bytes (2725 on the T200). Now run *FREMEM.BA* (use *FLOPPY* to load it from disk, if necessary) and then kill *FREMEM.BA*.

To run *NEWDOS.CO*, type *CLEAR 256, Top: RUNM "NEWDOS"* (in place of *Top* type *60003* for M100, *56675* for T200) and press *ENTER*. If you can spare a directory slot, you can save that line as a short *.BA* program that runs *NEWDOS.CO* for you, so you needn't remember the *Top* address for the *CLEAR* statement. To reclaim the high memory after using *NEWDOS.CO*, type *CLEAR 256, MAXRAM* and press *ENTER*.

By the way, if you ever forget the starting address of *NEWDOS.CO*, just type *LOADM "NEWDOS"* and press *ENTER*. *NEWDOS* attempts to load, and in doing so, it displays the *Top*, *End*, and *Exe* addresses of the file.

Be sure to save *NEWDOS.CO* to the same disk you use to boot *FLOPPY*. Then if you ever accidentally kill *NEWDOS.CO* or have a cold start (which should happen less often now), just boot *FLOPPY* and use it to load *NEWDOS.CO* and *FREMEM.BA*. Run *FREMEM.BA* and then kill it. Voila! Back to normal.

The drawback of normal *.CO* files: It takes two to tango, but only one dances!

COMPATIBILITY: Tandy 100/102 only

TEXTSTAR

Get WordStar *cursor control for* TEXT.

by Stan Wong

I'm a *WordStar* addict. There is nothing flashy about the program. It's not even sexy. But I love *WordStar* and its arcane cursor movement and control commands. It was, and still is, one of the few word processing programs supported across many different machines and operating systems. I use it on a DEC Rainbow and a 386 PC at work. I use it on my IBM PC/XT. And I bought an NEC 8500 CP/M laptop because it had *WordStar* built in.

My fingers get cramps every time I use my trusty Model 100, and trying to remember the different cursor movement and editing commands drives me batty.

Well, no more!

A DIAMOND IN THE ROUGH

The Model 100 *TEXT* program already implements the famous *WordStar* cursor control-character "diamond"—^S,^E,^D,^X, where the caret (^) signifies holding down the *CTRL* key, as *WordStar* users know—but it doesn't go far enough.

So I've polished up the diamond a bit. I've written a small machine language program called *TextStar* (or *Text** to its friends), which implements most of the *WordStar* cursor movement commands as well as some text deletion and other miscellaneous commands.

TextStar is not a complete program by itself. Instead, it enhances the operation of *TEXT*. Although *TextStar*'s cursor control looks and feels like *WordStar*, it still retains a *TEXT*-like flavor. See the sidebar "Inside *TextStar*" for technical information on how it performs its magic.

TextStar extends the operation of *TEXT* by redefining the meaning of the control keys, where possible, to match those of *WordStar* (See Table 1). Even multiple-keystroke commands, such as ^*QD*, are supported. Control keys that have not been redefined still retain their *TEXT* meanings. For instance, ^*B* still moves the cursor to the bottom of the screen. ^*T* will not, however, move you to the top of the screen but will delete a word instead.

A simple change to *TextStar* could have made it ignore the *TEXT* control keys, but I felt it would be better to leave them in. The program owes its small size (381 bytes) to its method; it traps keyboard input and translates only *WordStar* control codes into their *TEXT* equivalents, passing all other keystrokes directly to *TEXT* for normal processing.

VIVE LA DIFFERENCE!

However, *TextStar* does not perform exactly like *WordStar*. Because it enhances the *TEXT* program, it is bound by certain *TEXT* conventions.

```
Ø 'txtstr
2 GOSUB99
3 PRINT@42,"Enter end address";:PRINT@82
,"<ENTER> for just below current HIMEM"
4 Y="":PRINT@122,SPACE$(3Ø);:PRINT@122,;
:INPUTY:IFY=""THENB=HIMEM-E-1ELSEB=VAL(Y
)-E
5 IFB>=MAXRAM-ETHENBEEP:GOTO4ELSECLEAR25
6,B:B=HIMEM:GOSUB99
6 CLS:A=B:C=B+E:D=D+B:S=99:M=99:PRINT@48
,"Line";:PRINT@88,"Address";
7 GOSUB6Ø:IFO=ØTHENPOKEA,K:A=A+1:GOTO7
8 P=K:GOSUB6Ø:G=P+256*K+B:P=INT(G/256):Q
=G-P*256
9 POKEA,Q:A=A+1:POKEA,P:A=A+1:GOTO7
4Ø SAVEM"TXTSTR",B,C,D:MENU
5Ø U=INSTR(1,Z,MID$(Y,M,1)):IFU=ØTHEN55
51 IFU>16THENU=U-16:O=1ELSEO=Ø
52 V=INSTR(1,Z,MID$(Y,M+1,1)):IFV=ØORV>1
6THEN55
53 K=(U-1)*16+V-1:RETURN
55 BEEP:PRINT@165,"Bad character";:END
6Ø IFM<N*2+5THEN7Ø
61 IFL<>RTHENBEEP:PRINT@165,"Checksum fa
il";:END
62 READY:M=1:GOSUB5Ø:IFK=ØTHEN4Ø
63 S=S+1:PRINT@7Ø,S;:PRINT@11Ø,A;:N=K:M=
3:GOSUB5Ø:M=5:L=K:GOSUB5Ø:L=K*256+L:R=Ø
7Ø M=M+2:GOSUB5Ø:R=R+K:RETURN
99 DEFINTI-X:DEFSTRY-Z:Z="Ø123456789ABCD
EFGHIJKLMNOPQRSTUV":CLS:E=38Ø:D=Ø:RETURN
1ØØ DATA386E1A2ADEFA22N7Ø121J9Ø122DEFA2A
12FB22N9Ø121LDØ12212FB21ØØØ22E7F6C3EE5D
32DFF6F5CDEC65F1DAØ165FE7FCA1861FE2ØDAPC
ØØF53A
1Ø1 DATA38E217NCØ1FEØ1CAM8ØØ3ANBØ1FEØ1CA
KBØØF1C38A6Ø3EØØ32NBØ1F1CDMEØ1FE42CAS8ØØ
FE59CASDØØFE44CALDØ13EØAC3Ø56Ø3EØØ32NCØ1
F1CDME
1Ø2 DATA38431CØ1FE43DAP7ØØCAT2ØØFE44CAT7
ØØFE53CAV8ØØFE58CAVDØØFE52CAV3ØØFE45CATC
ØØFE59CAG7Ø13EØAC3Ø56ØFEØ7CA1861FE14CAI6
Ø1FE19
```

continued

Listing 1. The BASIC *loader for* TXTSTR. *Use this program to create the* .CO *program instead of using the assembly code in Listing 1.*

Cursor Movement	
^D	Right one character
^F	Right one word
^QD	Right to end of current line
^S	Left one character
^A	Left one word
^QS	Left to beginning of current line
^E	Up one line
^QE	Top of screen, column 1
^QR	Top of file
^X	Down one line
^QX	Bottom of screen, right end of line
^QC	End of file
Text Deletion	
^G	Delete character under the cursor
^T	Delete word
^Y	Delete line
^QY	Delete to end of line
Miscellaneous	
^KB	Mark start of block (same as F7)
^KY	Delete block (same as F6)*
^KD	Exit TextStar to menu (same as F8)

* Note: The ^KY command operates differently in TextStar. In Wordstar you would normally have to type ^KK to mark the end of a block before deleting it. However, ^KY, as defined here, is more consistent with the TEXT function key meanings.

Table 1. With TXTSTR, *you can use these* WordStar *keystrokes in* TEXT. *As with* WordStar, *the caret (^) indicates that you hold down the* CTRL *key.*

```
1Ø3 DATA387E15CAI5Ø1FEØBCAR8ØØFE11CASØØØ
C3Ø56ØF53EØ132NBØ1F1C9F53EØ132NCØ1F1C93E
ØCC3Ø56Ø3E15C3Ø56Ø3E1AC3Ø56Ø3E12C3Ø56Ø3E
14CDØ5
1Ø4 DATA384F126Ø3A88F7FEØØC2UEØØ3EØAC3Ø5
6Ø3E11C3Ø56Ø3E17C3Ø56Ø3E11C3Ø56Ø3EØ2CD05
6Ø3E12C3Ø56Ø3EØCCDØ56Ø3E12CDØ56Ø3E15C3Ø5
6Ø3EØC
1Ø5 DATA387C11CDØ56Ø3EØ6CDØ56Ø3E15C3Ø56Ø
3E11CDØ56Ø3EØCCDØ56Ø3E18CDØ56Ø3E15C3Ø56Ø
F3C5D521ØØØ39EBØ114ØØ21ØØØ39Ø9F921I1ØØ
E3Ø1FØ
1Ø6 DATA2DD4155FØ8CAL7Ø1Ø9E3EBF9D1C1FBC9
E52AN7Ø122DEFA2AN9Ø12212FBE1C39757FE61D8
FE7BDØD62ØC9ØØØØØØØØØØØØ
1Ø7 DATAØØ
```

end of listing

For instance, when moving up and down a line, *WordStar* moves the cursor to the end of the line if the destination line is shorter. But *TextStar*, like *TEXT*, leaves the cursor in the same column regardless of the destination line's length.

As a bonus, the text deletion commands ^*T* and ^*Y* store the deleted text in the paste buffer. So if you make a mistake you can recover the text by pressing the *PASTE* key. This gives you a simple "UNDO" facility.

A STAR IS BORN

Listing 1 is a relocating *BASIC* loader. It creates a machine language program, *TXTSTR.CO*, to run anywhere you specify in high memory, or it will default to loading just under the current *HIMEM*. It can thus coexist with your other *.CO* programs. (Thanks to Phil Wheeler for his *BASBLD.PW3* program, which I used to create the loader.)

Listing 2 is the assembly language source code for *TextStar*. I used the *ADSM* assembler by James Yi, which you can find on the CompuServe Model 100 Forum. Minor syntax changes to the source code will also make it compatible with the *ROM2* assembler from Traveling Software, which I used to do the initial program development.

You have no need to use the assembly language listing,

Continued on page 10

INSIDE *TextStar*

The key to *TextStar*'s operation is its ability to intercept keystrokes before *TEXT* sees them, by using a programming mechanism known as a *RAM hook*. Before I describe how *TextStar* uses RAM hooks, let's take a look at what *TEXT* does with your keystrokes.

The *TEXT* program is constructed as one big loop starting at ROM address 5DEEH. *TEXT* waits for your keystroke, gets it, and takes the appropriate action.

It's interesting to note that near the beginning of *TEXT*, the program pushes the address of the top-of-the-*TEXT* loop onto the stack.

Normal loops end with a *JMP nnnn* instruction, where *nnnn* is the address of the top of the loop. But by pushing the top-of-the-loop address on the stack, the program can instead simply execute a *RET* instruction from anywhere to get back to the start.

This has two advantages. First, it saves memory for each loop return point. It needs only one byte for the *RET* instruction versus three for the *JMP*. Second, the code can be more easily relocated. If the code is, for some reason, changed and shifted in memory, then all the *JMP* instructions do not have to be recoded with the new top-of-the-loop address.

GIVE 'EM THE HOOK, THE HOOK ...

TEXT uses a ROM routine known as *CHGET* (located at 12CBH) to wait for your next keystroke. The *TEXT* call to this "Tandy-documented" routine is at address 63F1H.

CHGET makes use of a RAM hook. What's a RAM hook? Is it related to Captain Hook? Not really.

A RAM hook is a programming mechanism by which the system designer provides a way for users to modify or replace certain system routines. RAM hooks are commonly provided for ROM routines, since code in ROM is fixed and cannot be changed.

Specifically, some ROM routines incorporate an *RST 7* instruction to give users access to hooks, allowing them an oppor-

Continued on page 11

however, other than to understand what *TextStar* is doing and how it does it.

A RISING STAR

To run *TextStar*, place the widebar cursor over *TXTSTR.CO* on the main menu and press *ENTER*. You should then see *TEXT*'s familiar *File to edit?* prompt. Specify the *.DO* file to edit, and you're on your way.

If the program doesn't start, go into *BASIC* and type *CLEAR 0, nnnnn* (where *nnnnn* is the load point) and then try again. If you don't know the load point, type *LOADM "TXTSTR"* and use the *Top* address, returned by the *LOADM* command, as the load point (*nnnnn*) for the *CLEAR* statement above.

WHERE TO GET *TEXTSTAR*

TextStar is also available on the *P100-To-Go* disk and on the *Portable BBS* (603-924-9770) as *TXTSTR.100*. If you need the program and listings on a cassette tape, send $6 to me at P.O. Box 28181, Santa Ana, CA 92799. You can also contact me via CompuServe user ID 70346,1267. I support the program on the M100SIG of CompuServe.

Those cramps in my fingers are finally starting to disappear. Happy typing! ☆

```
;TextStar - Wordstar(tm) commands
;for the Model 1ØØ
; Version 1.Ø   11 October 1988
;
;Copyright (c) 1988 by Stanley Wong
;All Rights Reserved
;
;ADSM assembler syntax
;See Table 1 for commands supported.
;See Listing 2 for Basic loader
;
;Equates
;
MENU    EQU     22423D  ;go to main MENU
                        ;($5797)
TEXT    EQU     24Ø46D  ;TEXT program
                        ;entry ($5DEE)
BEEP    EQU     7662H   ;beep routine
GETKEY  EQU     64222D  ;GHGET hook
                        ;($FADE)
;
;Other Rom Routine Addresses
;---------------------------
;$5FFØ - return to TEXT from GHGET
;$6ØØ5 - process control key
;$6Ø8A - process keystroke
;$6118 - DEL routine
;
        ORG     625ØØ
WSTINI  LHLD    GETKEY  ;point to
                        ;GETKEY hook
        SHLD    GKHOOK  ;save original
                        ;value
        LXI     H,TS1   ;point to our
                        ;code to examine
                        ;keystrokes
        SHLD    GETKEY  ;GETKEY hook to
                        ;our code
        LHLD    $FB12   ;save original
                        ;menu hook
        SHLD    MHOOK
        LXI     H,F8    ;set Menu hook
                        ;to our code
        SHLD    $FB12
        LXI     H,Ø
        SHLD    F6E7H
        JMP     TEXT    ;enter TEXT
                        ;program
;
;Examine Key Press
; Code at TS1 inserts a RETurn to here
; if in TEXT
; TS1 is called from GETKEY RST 7 hook
;
WSTXT
;---------------------------------------
;This code is Copyright by Microsoft
        STA     F6DFH   ;Duplicate ROM
                        ;code here
        PUSH    PSW     ;until we detect
                        ; CTRL keystroke
        CALL    65ECH
        POP     PSW
        JC      65Ø1H
        CPI     127
        JZ      6118H   ;DEL key
        CPI     32
;--------End of Microsoft code---------
        JC      CTRL    ;CTRL key
;A regular character received
        PUSH    PSW
        LDA     CQFLAG  ;check ctrl-Q
                        ;flag
        CPI     1
        JZ      CQON    ;^Q in effect
        LDA     CKFLAG  ;check ctrl-K
                        ;flag
        CPI     1
        JZ      CKON    ;^K in effect
        POP     PSW
        JMP     6Ø8AH   ;continue in rom
;
;control-K in effect
CKON    MVI     A,Ø     ;turn off flag
        STA     CKFLAG
        POP     PSW
```


Continued

Listing 2. The assembly code for TXTSTR, a program that makes TEXT emulate WordStar control code commands.

tunity to "splice in" their own code instead. It works like this.

At address FADAH is a "hook table," each table entry being a two-byte address. In effect, the *RST 7* instruction is a *CALL* to one of these addresses. It finds the appropriate entry by adding an offset byte (which follows the *RST 7* instruction in the ROM routine) to the start of the table, and then *CALL*'s the address stored there.

For example, the *CHGET* routine incorporates an *RST 7* instruction at the beginning of the routine. The value of the offset byte is 4. This means that the RAM hook at FADEH (FADAH+4) is used for *CHGET*.

Since most entries in the hook table point to a *RET* instruction in ROM (at 7FF3H), the *RST 7* normally *CALL*'s 7FF3H, which simply returns back to the ROM routine, chewing up a few machine cycles and nothing more.

But since the hook table is in RAM, the user may change an entry to point to his own routine, in which case the *RST 7* performs a *CALL* to the user's routine.

Once *RST 7* locates the two-byte address in the table, it executes the routine located at that stored address. For routines like *TextStar* that use these hooks, *RST 7* effectively *CALL*'s *TextStar*, which does its business on a keystroke and then returns to *CHGET*.

Thus, a hook gives us a chance to sneak in and peek at a keystroke before it is processed and returned to *TEXT*.

CAN YOU DIRECT ME TO ... ?

Okay, so now that we know what a RAM hook is, how do we use it? The first thing to do is save the original contents of


```
        CALL    LW2UP   ;force to upper
                        ;case
        CPI     66      ;^KB?
        JZ      CTRLKB
        CPI     89      ;^KY?
        JZ      CTRLKY
        CPI     68      ;^KD?
        JZ      F8
        MVI     A,1Ø    ;ignore
        JMP     $6ØØ5
;
:control-Q in effect
CQON    MVI     A,Ø     ;turn off flag
        STA     CQFLAG
        POP     PSW
        CALL    LW2UP
        CPI     67      ;^QC?
        JC      IGN2
        JZ      CTRLQC
        CPI     68      ;^QD?
        JZ      CTRLQD
        CPI     83      ;^QS?
        JZ      CTRLQS
        CPI     88      ;^QX?
        JZ      CTRLQX
        CPI     82      ;^QR?
        JZ      CTRLQR
        CPI     69      ;^QE?
        JZ      CTRLQE
        CPI     89      ;^QY?
        JZ      CTRLQY
IGN2    MVI     A,1Ø    ;ignore
        JMP     $6ØØ5
;
;Look for our control characters
;
CTRL    CPI     7       ;ctrl-G ?
        JZ      $6118   ;yes-jump to
                        ;DEL routine
        CPI     2Ø      ;ctrl-T ?
        JZ      CTRLT
        CPI     25      ;ctrl-Y ?
```

Continued

```
        JZ      CTRLY
        CPI     11      ;ctrl-K ?
        JZ      CTRLK
        CPI     17      ;ctrl-Q ?
        JZ      CTRLQ
        JMP     6ØØ5H   ;continue in ROM
;
CTRLK   PUSH    PSW     ;^K received,
                        ;wait for next
                        ;char.
        MVI     A,1     ;set ckflag
        STA     CKFLAG
        POP     PSW
        RET
;
CTRLQ   PUSH    PSW     ;^Q received,
                        ;wait for next
                        ;char.
        MVI     A,1     ;set cqflag
        STA     CQFLAG
        POP     PSW
        RET
;---------------------------------------
CTRLKB  MVI     A,12    ;^L - select
                        ;text mode
        JMP     $6ØØ5
CTRLKY  MVI     A,21    ;^U - cut(F6)
        JMP     $6ØØ5
;---------------------------------------
CTRLQC  MVI     A,26    ;move to eof
                        ;(^Z)
        JMP     $6ØØ5
CTRLQD  MVI     A,18    ;end of line
                        ;(^R)
        JMP     $6ØØ5
CTRLQE  MVI     A,2Ø    ;top of screen
                        ;(^T)
        CALL    $6ØØ5
        LDA     F788H
        CPI     Ø
        JNZ     X1
        MVI     A,1Ø
```

Continued

```
        JMP     $6ØØ5
X1      MVI     A,17
        JMP     $6ØØ5
CTRLQR  MVI     A,23    ;beginning of
                        ;file (^W)
        JMP     $6ØØ5
CTRLQS  MVI     A,17    ;beginning of
                        ;line (^Q)
        JMP     $6ØØ5
CTRLQX  MVI     A,2     ;bottom of
                        ;screen (^B)
        CALL    $6ØØ5
        MVI     A,18    ;right end of
                        ;last line
        JMP     $6ØØ5
CTRLQY  MVI     A,12    ;^L - select
                        ;text mode
        CALL    $6ØØ5
        MVI     A,18    ;^R - beginning
                        ;of line
        CALL    $6ØØ5
        MVI     A,21    ;^U - cut
        JMP     $6ØØ5
;---------------------------------------
CTRLT   MVI     A,12    ;^L - select
                        ;text mode
        CALL    $6ØØ5
        MVI     A,6     ;^F - move to
                        ; next word
        CALL    $6ØØ5
        MVI     A,21    ;^U - cut (F6)
        JMP     $6ØØ5
;
CTRLY   MVI     A,17    ;^Q - move to
                        ;left of line
        CALL    $6ØØ5
        MVI     A,12    ;^L - select
                        ;text mode
        CALL    $6ØØ5
        MVI     A,24    ;^X - move
                        ;down a line
        CALL    $6ØØ5
        MVI     A,21    ;^U - cut (F6)
        JMP     $6ØØ5
;
;
;RST 7 hook from GETKEY
;
;1Ø levels down in stack is the return
; address to the calling routine.  If it
; is 5FFØH then the call is from TEXT.
; Put a return to WSTXT. If not, then
; leave everything alone.
;
TS1     DI              ;don't bother
                        ;me, kid...
        PUSH    B       ;save registers
                        ;used
        PUSH    D
        LXI     H,Ø
        DAD     SP
        XCHG            ;save SP in D
;
        LXI     B,2Ø    ;get return
                        ;address 1Ø
        LXI     H,Ø     ;layers down in
                        ;stack
        DAD     SP      ;HL<--SP
        DAD     B       ;HL<--SP+2Ø
        SPHL            ;SP<--SP+2Ø
;
        LXI     H,WSTXT ;RETurn to WSTXT
                        ;instead of
                        ;5FFØH
        XTHL
        LXI     B,$5FFØ ;Was it called
                        ;from TEXT?
        DB      8       ; (hl<--hl-bc)
        JZ      TSX     ;Yes
        DAD     B
        XTHL            ;No
```

continued

continued

the RAM hook at FADEH. Typically it is just a pointer to a *RET* (address 7FF3H), but someone else might be using it, so we'll politely save it and restore it when we're done.

Next, we put the address of our own code (we'll use the address *TS1*) into the hook. Now every time *TEXT* gets a keystroke, we get a peek at it first.

RELAX AND UNWIND

Now that we can get a peek at a keystroke before *TEXT* does, what next?

Other programs use the *CHGET* routine, too. So we have to make sure that *CHGET* was invoked by *TEXT* and not another program. *TEXT* calls *CHGET* at address 5FEDH. (Actually, it's called at 63F1H, from within a routine called by 5FEDH.) This means that there will be a *RET*urn to 5FF0H somewhere on the stack.

When we get control via the FADEH RAM hook, the return address to *TEXT*, if it was called by *TEXT*, is 10 levels (20 bytes) down in the stack.

So we have to unwind the stack 10 levels and see if the address 5FF0H is there. If it is, then *CHGET* was called from *TEXT* and this keystroke is for us. If not, then we must put everything back the way it was.

How can a keystroke in *TEXT* not be for us? There are functions within *TEXT*, such as the *F1* (*Find*) function, that require keyboard input separate from the main *TEXT* processing loop.

The stack unwinding code is liberally borrowed from the *TXT-OV.ASM* program by James Yi and Paul Globman. The code uses an undocumented 8085 instruction that subtracts *BC* from *HL* and stores the result in *HL*:

HL <— HL - BC

The mnemonic for this instruction is *HLMBC* in the *ROM2* assembler syntax and *DSUB* in the *ADSM* assembler. If defined instead as *DB 8*, as in Listing 1, all assemblers will support it.

Sixteen-bit arithmetic on the 8085 is quite limited. This, and several other undocumented instructions, can save many instructions and make for easier programming.

WHAT A CHARACTER!

At label *WSTXT*, we examine the keystroke. If the character is a control character, then we have to look for "our" control

```
TSX     XCHG            ;Restore stack
        SPHL            ;(old SP was
                        ;in DE)
        POP     D
        POP     B
        EI              ;allow
                        ;interrupts
        RET             ;RETurn from
                        ;hook
;
;Menu hook (F8 keypress to exit WS-TEXT)
;
F8      PUSH    H
        LHLD    GKHOOK  ;Restore
                        ;original GETKEY
                        ;hook
        SHLD    GETKEY
        LHLD    MHOOK   ;Restore
                        ;original Menu
                        ;hook
        SHLD    $FB12
        POP     H
        JMP     MENU    ;return to MENU
;
;Convert lower to upper case
: input/output in A-register
LW2UP   CPI     97      ;lower case is
        RC              ; in the range
        CPI     123     ; 97<=A<=122
        RNC
        SUI     32      ;convert
        RET
;
;Storage
;
GKHOOK  DW      Ø       ;GETKEY hook
MHOOK   DW      Ø       ;MENU hook
CKFLAG  DB      Ø       ;ctrl-K flag
CQFLAG  DB      Ø       ;ctrl-Q flag
;
        END             ;of TextStar
```

End of listing.

characters. Otherwise, we pass control to ROM location 6005H for control characters and location 608AH for "normal" characters and process the keystroke as if we had never been there.

When we do find our control character, we simulate the *Wordstar* control sequence by sending *TEXT* a series of its own commands that will implement the *Wordstar* function. Look at the code for the ^*Y* function (delete line) as an example.

Multiple keystroke sequences, such as ^*KD*, pose a more difficult problem. When a ^*K* is typed, we can't act until we know what the next keystroke is. The solution is to set a flag indicating that the ^*K* has been typed. On every keystroke we have to check the flag to see if a ^*K* is in effect.

On a ^*KD* (or *F8*) exit command, we must restore the RAM hooks to their original values and exit to the main menu by *JMP*ing to location 5797H. Failure to do so may result in a machine that feels very cool to the touch (i.e., COLD START!). ☆

continued from page 21

A PROGRAM YOU SHOULDN'T BE WITHOUT

In the past five months, I have tried various ways to get the most programs at my fingertips in RAM without starting a war among the TSR's. Persoft's *Referee* program has been a wonderful addition. It consists of three programs that allow you to activate/deactivate TSR's either from the command line by invoking *REFEREE.COM* with certain parameters or from a menu-driven TSR (*SIDELINE.COM*).

A program can be activated or deactivated from the command line or batch file. For example, *REFEREE -BW* deactivates Traveling Software's *Battery Watch* program, and *REFEREE +BW* reactivates it. *REFEREE #1400CLK* unloads the 1400LT clock program, if it was the last program loaded into memory. *REFEREE ##* unloads every program in memory back to *REFWATCH.COM*.

For the TSR version, you load a "watchdog" program (*REFWATCH.COM*) before you load other TSR's. I load it immediately after the *AUTOEXEC.BAT* section displayed above. *REFWATCH* pays attention to every program loaded after it. If you wish to deactivate a program, call the TSR *Sideline* with *ALT-/*, cursor through the list of programs loaded into memory, and toggle any or all of them off and on with the space bar.

Referee helps avoid confusion among programs. I use Broderbund's *MemoryMate* extensively in the office. This text-based database lets me keep all kinds of notes at my fingertips. You invoke it by pressing *ALT-Z*, a combination I like because the two keys are close together. Another favorite application, however, uses *ALT-Z* to call up its help screen. Instead of the help screen for that program, I was getting *MemoryMate*. So now I temporarily deactivate *MemoryMate* by including *REFEREE -MEMO* (*MEMO* is the command name for *MemoryMate*) in the batch file that loads the other application. The last command in the same batch file is *REFEREE +MEMO*, which reactivates *MemoryMate* after exiting the application.

When using *RBASE for DOS*, I need all the memory I can get, so I load it with a batch file whose first command is *REFEREE ##*. This command clears out everything before loading *RBASE*.

I experimented with several shareware and public domain TSR managers and haven't found anything that pleases me more than *Referee*. It's a winner!

WORDPERFECT AND THE LITEDRIVE II

My new hard disk overcomes one other limitation of the 1400LT—the inability to upgrade with expanded or extended memory. The hard drive controller board occupies the only slot in the machine.

We use *WordPerfect 5.0* and its accompanying *Library* programs in our offices. I like using the shell, but couldn't move from one program to another without tedious unloads and loads because of memory limitations. *WordPerfect*'s *Library 2.0* lets you use your hard disk as simulated memory, swapping programs in and out from the disk as you move from one to the other with keyboard macro commands. Loading the shell with *SHELL /V-C:\LIBRARY* lets me swap files out to my *LIBRARY* directory. The shell erases its extra swap files if you exit it properly with *F7*.

I now can run WordPerfect's *Shell* program with the appointment calendar, calculator and *WordPerfect 5.0* in memory (or simulated disk memory). In addition, I keep resident in memory *MemoryMate*, *PC Magazine*'s *Cardfile* (address Rolodex), *Battery Watch*, *StopLite*, and *DOSEDIT* (a command line editor). I've packed this little white laptop with about everything it can hold. The resulting experience continues to be a pleasant one.

Jim Berg is Dean of Students at Bob Jones University in Greenville, SC and is an avid computer hobbyist. Tom Berg is PC Coordinator for the University Electronics Department at Bob Jones University.

COMPATIBILITY: Tandy 600, 1400LT (untested).

AUDIT 600

How to manage your utility bills with the Tandy 600.

By Dennis C. Rogers

Utility Bill Audit is a versatile program that lets you check your electric, gas, water, and phone bills for accuracy, or split the costs of these bills among the people living in your household. Also, if you are interested in energy savings (and who isn't these days?), you can monitor your daily electric and gas consumption with this program.

The program listed will run on a Model 600 laptop, and you can easily modify it to run on any *GWBASIC* or other Microsoft *BASIC* interpreter or compile it for speedier results.

PROGRAM MODIFICATIONS

Before you *RUN* this program, you must have a thorough understanding of how each bill is calculated in the program. First, a particular bill is split up according to the values (*2,1,2,4*) given in the *DATA* statement in line 1720. These values are assigned to the variable *N(X)* and represent the number of people who must pay for each bill. In its present form, the program assumes that two people pay for the electric bill, one pays for the gas bill, two pay for water, and four pay for the phone bill. However, it's unlikely that these numbers will correspond to the financial arrangements in your household. So be sure to substitute the appropriate values in this line before you continue. Of course, if one person pays the bills in the household, replace all four numbers in line 1720 with *1,1,1,1*.

You can easily modify it to run on any Microsoft *BASIC*

Since the program works on the actual costs of your utility bills based on local rates, you have to provide certain information about these rates before you can run the program. This information is *READ* in lines 560 and 580 from the *DATA* statements in lines 1730-1760.

Notice that the first three *DATA* statements in this sequence have nine entries and apply to the electric, gas, and water bills respectively. Let's consider line 1730 as an example. The first entry in this line is the name of the utility (*ELECTRIC*) for which the rates that follow apply. The second entry is the unit of measurement for that particular utility (*KWH* for kilowatt hours). The next entry is the minimum service charge for the utility ($5.40 for electricity). The fourth entry is the tax rate based on the sum of the service charge and the rate charge (0 percent for electric use). These first four *DATA* entries are *READ* in as *A$(1)*, *B$(1)*, *M(1)*, and *Z(1)*, respectively.

At this point, the numbers begin to get a little confusing, so

```
10 '-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
-=-=-=-=-=-=-
20 '     B I L L   A U D I T   P R O G R
 A M
30 '
40 '     E A S Y   W A R E   P R O D U C
 T S
50 '            P.O. Box 218778
60 '       Houston,  Texas    77218-877
8
70 '
80 ' Author - Dennis C. Rogers
90 ' Version - 1.02
100 ' Date - 24 Nov 1988          Model 6
00 Version
110 '
120 ' COPYRIGHT 1988  EASY WARE PRODUCTS
130 ' A L L   R I G H T S   R E S E R V E
D.
140 '-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
=-=-=-=-=-=-
150 '
160 '
170 DIM A$(4),B$(4),L1(3),L2(3),M(4)
180 DIM R1(3),R2(3),R3(3),W(50),Z(4),N(1
0)
190 GOSUB 1790: GOTO 520
200 A1=0
210 PRINT "Enter Adjustments to Bill (+
or - ,'0' When Done)"
220 IF IX>12 THEN IY=3:IYY=15:IC=1:ICC=7
8:GOSUB 350:IX=0 'ALLOW 12 ENTRIES
230 INPUT E
240 A1=A1+E:IX=IX+1
250 IF E=0 THEN GOTO 270
260 GOTO 220              'LOOP
270 GOSUB 430             'SKIP 2 LINES
```

continued

This bill audit program lets you check and monitor your utility bills—and keep your roommates honest!

```
28Ø GOSUB 47Ø            'PROMPT TO EXIT
29Ø RETURN
3ØØ CLS
31Ø RETURN
32Ø PRINT "  ";A$(X);" Bill (Cont')"
33Ø RETURN
34Ø ' Programmable Cursor Controlled - C
lear Screen
35Ø FOR L=IY TO IYY
36Ø FOR C=IC TO ICC
37Ø LOCATE L,C
38Ø PRINT " ";
39Ø NEXT C
4ØØ NEXT L
41Ø LOCATE IY,IC
42Ø RETURN
43Ø FOR I= 1 TO 2
44Ø PRINT
45Ø NEXT I
46Ø RETURN
47Ø PRINT "Enter (X) to Exit";
48Ø INPUT C$
49Ø IF C$="X" OR C$="x" THEN 59Ø
5ØØ GOSUB 3ØØ
51Ø RETURN
52Ø FOR I= 1 TO 4
53Ø READ N(I)
54Ø NEXT I
55Ø FOR I= 1 TO 3
56Ø READ A$(I),B$(I),M(I),Z(I),L1(I),L2(
I),R1(I),R2(I),R3(I)
57Ø NEXT I
58Ø READ A$(4),M(4),Z(4)
59Ø GOSUB 3ØØ:IX=2Ø
6ØØ LOCATE  2,IX:PRINT "  Audit 6ØØ - Ut
ility Bill Auditer"
61Ø LOCATE  3,IX:PRINT "       By  Easy
Ware Products"
62Ø LOCATE  4,IX:PRINT "Copyright 1988,
 ALL RIGHTS RESERVED"
63Ø LOCATE  5,IX:FOR I=1 TO 36:PRINT CHR
$(2Ø5);:NEXT I
64Ø LOCATE  7,IX:PRINT "         1. Elec
tric Bill"
65Ø LOCATE  8,IX:PRINT "         2. Gas
Bill"
66Ø LOCATE  9,IX:PRINT "         3. Wate
r Bill"
67Ø LOCATE 1Ø,IX:PRINT "         4. Phon
e Bill"
68Ø LOCATE 11,IX:PRINT "         5. All
the Above"
69Ø LOCATE 12,IX:PRINT "         6. Exit
"
7ØØ LOCATE 14,25:PRINT " Choose an Optio
n ";
71Ø INPUT P
72Ø IF (P<1)+(P>6) THEN GOTO 71Ø
73Ø ON P GOTO 119Ø,122Ø,125Ø,128Ø,167Ø,1
77Ø
74Ø GOSUB 3ØØ    'CLS
75Ø PRINT "      ";A$(X);" Bill"
```

continued

Audit 600, *a program that enables you to check the accuracy of your utility bills, is designed for use on the Tandy 600 portable computer.*

read carefully. The next two numbers are cut-off limits for each electric rate and are represented in the program by *L1(1)* and *L2(1)*. The last three numbers are the actual rates charged per KWH use for each level of use *R1(1)*, *R2(1)*, and *R3(1)* in the program. Thus, the program is presently set up so that the rate charged for electricity is $0.0495 for the first 350 KWH, $0.0565 for the next 950 KWH (i.e., 1300 minus 350), and $0.0541 for any usage exceeding 1300 KWH.

The *DATA* statement in line 1760 is easier to follow and includes this: the utility (*PHONE*), the minimum service charge ($13.50), and the tax rate on the service charge and long distance calls (3 percent).

So get out your most recent bills and read off the various rates (per KWH for electric, per CCF or hundred cubic feet for gas and water). If the rates are not given on a bill, contact the utility company to get a schedule for the latest rates. Then just substitute your local rates for those in the *DATA* statements in lines 1730 to 1760. In this way, you can compute or audit your utility bills.

ANALYZING THE PROGRAM

After inserting the correct rates, *RUN* the program. You will then be asked which utility bill you wish to check. The first three menu choices are electric, gas, and water. Bills for these three utilities are all calculated in the routine beginning at line 740. Let's look at an electric bill as an example.

When the routine at line 740 is executed, you will be required to *INPUT* the present and previous meter readings. These values can be read directly from your latest electric bill. Next, you must *INPUT* the number of days in the billing period. Then you will be asked to *INPUT* any adjustments to the bill, either positive (connection fees, previous balances, or other debits) or negative (credits).

The program next calculates the amount of electricity consumed for the given period (defined as *U* in line 890). Then, depending on the value of *U* relative to the two rate limits, *L1(1)* and *L2(X)* in lines 910 and 920, an amount owed (*T*) before tax and adjustments will be calculated (lines 930, 950 and 970). Next, it determines the tax on this amount (*T1*). And finally in line 1000, it calculates a total electric bill (*T3*)—the sum of the minimum charge, usage cost, tax, and adjustments.

The results are then *PRINT*ed on the screen with provisions for formatting the output to two places past the decimal. Any numbers in the third place past the decimal are simply dropped. If you prefer rounded numbers, you could easily modify the program to achieve that.

The routine beginning at line 740, as mentioned, also calculates the gas and water bills. These are based on the rates *READ* from the *DATA* statements in lines 1740 and 1750. Notice the sets of large numbers (i.e., *99999*) in line 1740. The rates for gas where I live are the same, regardless of the amount used. By using large numbers here for the cut-off limits, *L1(2)* and *L2(2)*, for this utility, you aren't likely to exceed these amounts (see lines 910 and 920). Thus, the charge for this commodity will always be based on the first rate, or *R1*.

The rates for water, as *READ* from the *DATA* statements in line 1750, are based on a single cut-off limit *L1(3)* of 1000 CCF. For less than this, it charges you a usage rate *R1(3)* of $0.144 per CCF. If water use exceeds 1000 CCF, it charges you a second rate *R2(3)* of $0.160. Again, using a very large number (*99999*) for the second cut-off limit *L2(3)* assures that the overall usage cost is based on only two rates.

INTERPRETING THE PHONE BILL

The program checks phone bills in a separate routine beginning at line 1280. This routine initially *INPUT*s adjustments to the bill in the same way as with the electric, gas, and water bills. Next, the person responsible for each long distance charge is required to *INPUT* the amount of each long distance call. A separate routine (lines 1440-1550) allows you to correct any typing mistakes. Finally, it displays the amount owed by one person.

The portion of the phone bill that each person must pay is the sum of his or her long distance tolls, a proportional amount of both the service charge and the billing adjustments, and a proportional amount of the tax levied on the service and long distance calls. Again, if only one

```
76Ø PRINT
77Ø PRINT "Previous Meter Reading"
78Ø INPUT E1
79Ø PRINT "Present Meter Reading"
8ØØ INPUT E2
81Ø PRINT
82Ø PRINT "Input Days in the Billing Per
iod"
83Ø INPUT D
84Ø GOSUB 43Ø    'SKIP THREE LINES
85Ø GOSUB 47Ø    'CONTINUE?
86Ø GOSUB 3ØØ    'CLS
87Ø GOSUB 32Ø    'TYPE OF BILL DESIRED
88Ø GOSUB 2ØØ    'ENTER ADJUSTMENTS
89Ø U=E2-E1
9ØØ Y=U/D
91Ø IF U>L2(X) THEN GOTO 97Ø
92Ø IF U>L1(X) THEN GOTO 95Ø
93Ø T=M(X)+R1(X)*U
94Ø GOTO 98Ø
95Ø T=M(X)+R1(X)*L1(X)+R2(X)*(U-L1(X))
96Ø GOTO 98Ø
97Ø T=M(X)+R1(X)*L1(X)+R2(X)*(L2(X)-L1(X
))+R3(X)*(U-L2(X))
98Ø T1=T*Z(X)
99Ø T2=T+T1
1ØØØ T3=T2+A1
1Ø1Ø GOSUB 3ØØ   'CLS
1Ø2Ø GOSUB 32Ø   'TYPE OF BILL DESIRED
1Ø3Ø PRINT "Use for the Period is ";INT(
U*1ØØ)/1ØØ;" ";B$(X)
1Ø4Ø PRINT
1Ø5Ø PRINT "Use/Day is ";INT(Y*1ØØ)/1ØØ;
" ";B$(X);" or $";
1Ø6Ø PRINT INT(T2/D*1ØØ)/1ØØ'"/Day Inclu
ding Tax"
1Ø7Ø PRINT
1Ø8Ø PRINT A$(X);" Bill:"
1Ø9Ø PRINT " W/Out Tax   :$";INT(T*1ØØ)/1
ØØ
11ØØ PRINT " Tax is      :$";INT(T1*1ØØ)/
1ØØ
111Ø PRINT " Adj'ts      :$";A1
112Ø PRINT
113Ø PRINT "  *Total*    :$";INT(T3*1ØØ)/
1ØØ
114Ø IF N(X)=1 THEN GOTO 116Ø
115Ø PRINT "Split ";N(X);" Ways:$";INT(T
3/N(X)*1ØØ)/1ØØ
116Ø PRINT
117Ø GOSUB 47Ø           'CONTINUE?
118Ø RETURN
119Ø X=1
12ØØ GOSUB 74Ø
121Ø GOTO 59Ø
122Ø X=2
123Ø GOSUB 74Ø
124Ø GOTO 59Ø
125Ø X=3
126Ø GOSUB 74Ø
127Ø GOTO 59Ø
128Ø GOSUB 3ØØ
129Ø X=4
13ØØ PRINT "      ";A$(X);" Bill"
131Ø GOSUB 2ØØ
132Ø FOR K=1 TO N(X)
133Ø I=1
134Ø PRINT "For Person #";K;",Enter Char
ge for each LD Call (Enter 'Ø' When Done
)"
135Ø IF IX>12 THEN IY=2:IYY=15:IC=1:ICC=
78:GOSUB 35Ø:IX=Ø 'ALLOW 12 ENTRIES
136Ø INPUT W(I)
137Ø IF W(I)=Ø THEN GOTO 14ØØ
138Ø I=I+1:IX=IX+1
139Ø GOTO 135Ø           'LOOP
14ØØ GOSUB 3ØØ           'CLS
141Ø FOR J=1 TO I-1
142Ø IX=CSRLIN
143Ø IF IX>12 THEN IY=1:IYY=15:IC=1:ICC=
78:GOSUB 35Ø 'ALLOW 12 ENTRIES
144Ø PRINT "Person #";K;",Call #";J;"  :
```

continued

continued

person in the household foots the bills, the last number in line 1720 should be *1*.

CONCLUSION

In addition to enabling you to catch billing errors and helping you to divide up household bills easily, this program can help you monitor your costs. If you add an energy-saving device that is supposed to save 10 percent of your total electric bill, take a meter reading when you install it and verify the savings with a later reading. You can also project weekly, monthly, and yearly savings for any utility in this way.

I fully intend to support the Model 600 so long as I get enough support from the users of this machine. In the meantime, I will be writing more articles on the Tandy Model 600, specifically. But note that these programs should work with little or no change on almost all PC's and laptops in the marketplace. I fully believe this machine has much more potential than the models before it.

The *AUDIT 600* program is available on 3.5-inch Tandy Model 600 format for $10. It is also available on 5.25-inch or 3.5-inch PC-compatible format for the same price. It is available by writing to the address below specifying the article title with money order enclosed and return address. Easy Ware Products (CLUB 600), P.O. Box 218778, Houston, Texas 77218-8778; (713) 933-0542

Dennis Rogers is president and owner of Easy Ware Products. He is also president of Club 600, and editor/publisher of the Club 600 Newsletter. *He talks with a southern accent and he's friendly.*


```
$";W(J);"  Is this Correct (Y/N) - ";
145Ø INPUT C$
146Ø IF C$="Y" OR C$="y" THEN GOTO 149Ø
147Ø PRINT "Type in Correction - ";
148Ø INPUT W(J)
149Ø NEXT J
15ØØ GOSUB 3ØØ
151Ø GOSUB 32Ø
152Ø T=Ø
153Ø FOR J=1 TO I-1
154Ø T=T+W(J)
155Ø NEXT J
156Ø PRINT "Service:     $";INT(M(X)/N(X)
*1ØØ)/1ØØ
157Ø PRINT "LD Calls:    $";T
158Ø PRINT "Adj'ts :     $";INT(A1/N(X)*1
ØØ)/1ØØ
159Ø T1=T+INT(M(X)/N(X)*1ØØ)/1ØØ
16ØØ T2=INT(T1*Z(X)*1ØØ)/1ØØ
161Ø PRINT "Total Tax:  $";T2
162Ø PRINT "Total Bill: $";T1+T2+INT(A1/
N(X)*1ØØ)/1ØØ
163Ø GOSUB 43Ø
164Ø GOSUB 47Ø
165Ø NEXT K
166Ø GOTO 59Ø
167Ø FOR F=1 TO 3
168Ø X=F
169Ø GOSUB 74Ø
17ØØ NEXT F
171Ø GOTO 128Ø
172Ø DATA 2,1,2,4
173Ø DATA ELECTRIC,KWH,5.4Ø,Ø,35Ø,13ØØ,.
Ø495,.Ø565,.Ø541
174Ø DATA GAS,CCF,4.Ø5,Ø,99999,99999,.49
541,Ø,Ø
175Ø DATA WATER,CCF,3.26,Ø,5ØØ,99999,.14
4,.16Ø,Ø
176Ø DATA PHONE,13.5Ø,.3Ø
177Ø CLS:PRINT "Exiting to HHOS, Wait fo
```

continued

```
r Menu ...":SYSTEM
178Ø REM  ----------- SIGNON MESSAGE ---
-----------
179Ø CLS
18ØØ MSGL=3
181Ø MSG1$="      Bill Audit Program"
182Ø MSG2$="         Presented By"
183Ø MSG3$="      Easy Ware Products"
184Ø MSG4$="       P.O. Box 218778"
185Ø MSG5$=" Houston, Texas  77218-8778"
186Ø KEY OFF:X=1:Y=15:V=13:W=53
187Ø LOCATE X,Y:PRINT CHR$(2Ø1);:FOR T=1
 TO W:PRINT CHR$(2Ø5);:NEXT I
188Ø PRINT CHR$(187)
189Ø FOR J= X+1 TO V
19ØØ LOCATE J,Y:PRINT CHR$(186);:PRINT S
PC(W);:PRINT CHR$(186)
191Ø NEXT J
192Ø LOCATE J,Y:PRINT CHR$(2ØØ);:FOR I=
1 TO W:PRINT CHR$(2Ø5);:NEXT I
193Ø PRINT CHR$(188)
194Ø GOSUB 196Ø
195Ø FOR K= 1 TO 8ØØØ:NEXT K:RETURN
196Ø LOCATE MSGL,28:PRINT MSG1$:MSGL=MSG
L+1
197Ø LOCATE MSGL,28:PRINT MSG2$:MSGL=MSG
L+1
198Ø LOCATE MSGL,28:PRINT MSG3$:MSGL=MSG
L+1
199Ø LOCATE MSGL,28:PRINT MSG4$:MSGL=MSG
L+1
2ØØØ LOCATE MSGL,28:PRINT MSG5$:MSGL=MSG
L+1
2Ø1Ø LOCATE MSGL,28:MSGL=MSGL+1
2Ø2Ø LOCATE MSGL.31:PRINT "    Copyright
 1988":MSGL=MSGL+1
2Ø3Ø LOCATE MSGL,31:PRINT "  All Right R
eserved":MSGL=MSGL+1
2Ø4Ø LOCATE MSGL,31:PRINT "Model 6ØØ Ver
sion 1.Ø2"
2Ø5Ø RETURN
```

End of listing.

Traveling Software. A Powerful

Laptop Classics

Your chance to save on quality items from our popular collection of Tandy Model 100/102 & Model 200 software and accessories.

ITEM	ITEM NO.	REG. PRICE	SALE*
SOFTWARE EXTRA ORDINAIRE			
Ultimate ROM II Four programs: ROM-View 80, T-Word, Idea, and T-base.	R1-UR-01 (M100) R2-UR-01 (M200) NE-UR-01 (NEC)	$199.95	$ 99.95
Sardine3.5" Disk Electronic spelling checker with pocket dictionary.	R1-SD-01 (M100) R2-SD-01 (M200) NE-SD-01 (NEC)	$ 99.95	$ 59.95
Sardine ROM Chip Includes T-Word word processor.	R1-SD-02 (M100) R2-SD-02 (M200) NE-SD-02 (NEC)	$169.95	$ 99.95
Sardine 4 Chip Set Put the Dictionary on Chips!	R1-SD-03	$199.95	$129.95
TS-DOS 4.0 Disk Operating system for your portable disk drives.	R1-TS-01 (M100) R2-TS-01 (M200) NE-TS-01 (NEC)	$ 69.95	$ 49.95
TS-DOS 4.0 ROM Chip Save your RAM!	R1-TS-02 (M100) R2-TS-02 (M200) NE-TS-02 (NEC)	$ 99.95	$ 59.95
ESSENTIAL ACCESSORIES			
Padded Cordura Carrying Case For your Model 100/102 or portable printer	AC BG-06	$ 29.95	$ 9.95
3.5" Disk Wallet Carries 8 diskettes.	AC-DW-02	$ 9.95	$ 6.95

All items above are limited to supplies on hand. Offer expires April 30, 1989

SHIPPING AND HANDLING (U.S.) Surface $5.00 / Air (2nd Day) $8.00

BY MAIL

Send Name, Address, Order Number, and Method of Payment.

BY PHONE

Credit Card/COD Orders
7 Days/24 Hours
1-800-343-8080.
In Washington Call:
206-483-8088.

BY FAX

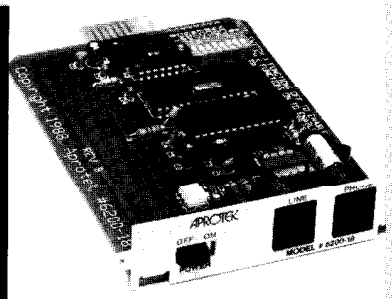
Send Your Completed Purchase Order To Traveling Software via Facsimile Transmission
206-487-1284
24 Hours-7 Days

Traveling Software

Dept P-WW2

18702 North Creek Parkway • Bothell, WA 98011

Connection For Your Tandy 100.



APROTEK MINIMODEM-T

A powerful, affordable modem for your Tandy 1400 LT.

The Aprotek Minimodem-T gives you the easiest access to telecommunication possible. It consumes one-sixth the power of a Tandy internal modem when in use and can be switched off when not in use. Tandy modems always draw current from your battery. The Minimodem-T weighs less than four ounces and has a two-year warranty. It is fully compatible with the Hayes AT command set. It operates at 1200 baud in either Bell 103/212A or CCITT V.21 and V.22 modes (switch selectable). Features include Auto Answer, Auto Dial, Auto Redial and Full Duplex communications.

Aprotek Minimodem-T $119.95

When you're up against hardwired phones, the acoustic coupler can be a lifesaver. Allows you to use your modem with either round or square handsets. A must for the laptop on the go!

Acoustic Coupler ..... ~~$49.95~~ **$34.95**

***Special Order Now ! (limited time offer).**

MAC-DOS II

Connects Your Mac to Your Laptop or Tandy/Purple Computing Portable Disk Drive.

MAC-DOS II gets your Mac and laptop "talking" by the connection of our three-connector cable. It lets you read, edit and print files created with your laptop as easily as regular Macintosh files. Transfers files between your Tandy M100/102/200 or NEC PC-8201/8300 laptop and any version of the Macintosh at speeds up to 19,200 baud instantly. Also includes a Tandy disk drive adapter for optional direct disk access.

MAC-DOS II with cable ..... ~~$129.95~~ **$109.95**

LAP-DOS II

Connects Your IBM Compatible PC to a Tandy M100/102/200, NEC PC-8201/8300 or Tandy/Purple Computing Disk Drive.

LAP-DOS II lets your IBM compatible (including PC, XT, AT and PS/2) transfer files from your laptop at speeds up to 19,200 baud using our specially designed three-connector cable. Also included is a Tandy disk drive adapter for direct disk access (either 100 or 200K format). LAP-DOS II has rename, view and delete functions and supports sub-directories on IBM or compatible PCs.

LAP-DOS II with cable ..... ~~$129.95~~ **$99.95**

TRAVELING SOFTWARE®

ORDER TOLL FREE U.S. 1-800-343-8080 in Washington State 206-483-8088 Visa, MC, AMX Accepted

Five Months With Twenty Megs

Experience has provided a compendium of useful tips for the 1400LT hard drive.

By Jim Berg

Author Stephen Lankton allowed us to commiserate with him as he tried to install CMS Enhancements' twenty-megabyte LiteDrive II hard disk in his Tandy 1400LT (*Portable 100*, Oct. '88). My drive's serial number is 000041, so it is one of the earliest ones shipped. And I was installing mine about the time his article was published, but, unaware of *Portable 100* then, I couldn't take advantage of his experience. Fortunately, I didn't have the same frustrations. My brother Tom, a computer technician, installed my drive without a hitch. I even had DOS 3.2 from a Tandy 1000SX, so when I couldn't format the hard disk with the 1400LT DOS, I immediately had a solution. I've been using it for five months now and have had a few struggles of my own. The effort was worth it, however, and I wouldn't want to be without my LiteDrive II.

My first hitch came when I tried to back up my hard drive with *Fastback Plus*. The program estimated the number of disks needed for the backup (forty-five disks) but gave me strange program errors when I actually tried the backup process. Technical support at 5th Generation was courteous but provided no answers. I next called Dahn Lee, the LiteDrive's designer, at CMS Enhancements. He referred me to Tandy's product manager who oversees the Tandy 1000 product line. He explained that the BIOS in the 1400LT is a proprietary one and could give problems with a backup program that bypassed DOS (which *Fastback Plus* does). He said the BIOS still thinks it is servicing drive B. I phoned this new information back to 5th Generation and went looking for a backup program that used DOS. I found one called *Diskpack* on a local BBS. The demoware version backs up and verifies data, but does not restore it. I obtained a registered version and couldn't be happier. It backs up thirteen megabytes onto eighteen 3.5-inch disks.

The "Mystery of the Missing Drive B" can continue to be a nasty one, however. If a program tries to access drive B, the computer goes "bye-bye." To counteract this, I include *ASSIGN B=C* in my *AUTOEXEC.BAT* file so that any subsequent calls to drive B are transferred to the hard disk.

MORE POWER TO YA'

Once the LiteDrive II is installed, it doesn't take long for the power drain to make the A/C adapter hot enough to fry an egg. Mr. Lee at CMS recommended that I build or purchase a separate power supply that provides twelve volts at 2.5 to 3 amps. My brother went to work again and came up with just the ticket. The schematic in Figure 1 is the result. He installed a one-eighth-inch phone jack in the back panel just above the external keyboard jack. I drilled another hole in the rear plastic flip-up panel just like the original power adapter hole so the new power supply could be plugged in with the back panel in place.

When the power supply is plugged in, the jack disconnects the battery. A diode across the jack terminals lets the battery jump back into action in case the power supply should lose power or get unplugged from the computer. Table 1 is the parts list for the power supply. This is not a project for the faint-hearted or the novice. If you can install the drive yourself, you probably can build the power supply, however. I keep the power supply at home where I use *WordPerfect* and *RBASE for DOS* quite extensively. At the office, where I use my 1400LT mainly for desktop organizing via *WordPerfect Library*, I use the regular charger and try to keep track of battery drain with Traveling Software's *Battery Watch*. Version 1.0G of the program just came in the mail and supports the 1400LT hard drive better than earlier versions. Traveling Software's support staff recommends adding 800 (ma) to the *HI* and *LO* system switches. I load the program for use with the hard disk as follows:

BW LO=928 HI=945 AL=10

Manufacturer's Specifications

Battery Watch—$39.95
Traveling Software
18702 North Creek Parkway
Bothell, WA 98011
(800)343-8080 or in WA (206)483-8088

Diskpack—Demoware ($49.00 registration)
Biologic Company
11982 Coverstone Hill Circle
Suite 1622
Manassas, VA 22110
(703) 368-2949

MemoryMate—$69.95
Broderbund Software, Inc.
17 Paul Drive
San Rafael, CA 94903-2101
(800)527-6263

Referee—$69.95
Persoft, Inc.
465 Science Drive
Madison, WI 53791
(608)273-6000

WordPerfect 5.0—$495
WordPerfect Library—$129
WordPerfect Corp.
1555 North Technology Way
Orem, UT 84057
(800)222-9409

RBASE for DOS—$700
Microrim, Inc.
3925 159th Ave. N.E.
P.O. Box 97022
Redmond, WA 98073-9722
(206)885-2000

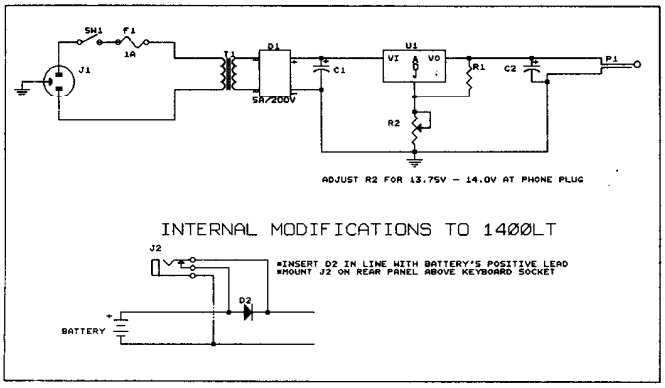


Figure 1. The plans for creating an AC power supply for the 1400LT. See Table 1 for a description of the parts needed.

The *AL=10* tells *Battery Watch* to ring its alarm ten minutes before the program reaches the "empty" reading. I found that the status screen (invoked by pressing *ALT-SHIFT-B, F3, F5*) accurately registers the percentage of use for my floppy drive. It would not, however, register *any* use of my hard drive. Another call to Traveling Software revealed that *Battery Watch* polls the computer's systems every 2.85 seconds to take its readings. The CMS drive's access time is generally less than a tenth of a second, and the program often misses it.

To test if this was really my problem, I set up a batch file with several calls to *CHKDSK* (with calls to *DIR* between each *CHKDSK*). My *Battery Watch* status screen still registered no drive activity. To this, Traveling Software replied that the program seems to be working accurately on CMS drives being marketed by Tandy but not on the drives CMS sells for itself. CMS told me that Tandy makes some internal changes to the power supply when they install drive at a Tandy Service Center. Perhaps these changes account for the different effects these drives have on *Battery Watch*.

In spite of this, *Battery Watch* is working quite accurately for me. Apparently, my use of the hard drive when on battery power corresponds quite closely to the drain represented by the 800ma addition. I am generally quite pleased but will be even more excited when Traveling Software finds out how to read the percentage of hard drive use accurately. These folks provide professional products and professional service. I'm confident they will eventually come up with an upgrade that will work for us CMS drive users.

Speaking of upgrades, Tandy recently sent me a disk with a new *FORMAT.COM* program and a new version of *SL.COM* (*StopLite* ver. 1.12). I simply left my name and address with Radio Shack Customer Service on the PC-LINK network and requested the updates.

The new version of *StopLite* is not memory resident, however. Since I like to change the time-out values frequently depending on whether I am using the battery or the extra power supply, I like to be able to call up the terminate-and-stay-resident (TSR) version and change the values at will, rather than waiting for the program to load from disk.

SCREEN HELPS

I found a useful clock program on PC-LINK as well. Most clocks don't work on the 1400LT's screen—again, because they bypass DOS calls. This one doesn't bypass DOS and seems to work well. I also picked up several other screen utilities there:

SB14.COM—This program turns off the backlit screen after user-defined periods of inactivity, but does not allow the power consumption to drop to "standby" levels. The biggest advantage is that since the power remains higher, it does not lock the COM port when your screen blanks out. Ever had your screen blank out in the middle of a long download? This one's a goody!

User-selected parameters tell the program to scan the keyboard, video calls to the BIOS, or both.

BLOFF.COM—Turns the backlit screen off. *CTRL-ALT-F11, CTRL-ALT-F12* or *CTRL-ALT-INS* will reactivate the light.

VIDM.COM—This third program checks to see which display is currently active—external CRT or LCD—and is useful if you need to determine which display is active so you can call the *MODE* command with appropriate parameters.

SW1	SPST power switch
F1	1-amp fuse with holder
T1	12V/3A transformer
C1	2200µF/50V electrolytic capacitor
C2	1µF/50V tantalum capacitor
U1	LM338K adjustable voltage regulator
R1	120-ohm, 1/4W resistor
R2	5K-ohm potentiometer 10-turn preferred
D1	5A/200V bridge rectifier
D2	1N5402 or equivalent
J1	AC line cord
J2	1/8-inch phone jack with switching contacts
P1	1/8-inch phone plug

Table 1. The parts needed to build your own external AC power supply for your Tandy 1400LT with a hard drive.

TIPS FOR THE RAM DISK

The extra 128K that Tandy makes available for a RAM disk was a thoughtful addition. Once the hard drive is installed, the RAM disk becomes drive D. I use *AUTOEXEC.BAT* to load that 128K with all kinds of goodies.

After setting the path to look for programs in the RAM disk first, and after clearing the screen, I display a brief lost-and-found message, set the *\COMSPEC* environment to look for *COMMAND.COM* in drive D, set the cursor prompt message to display the current directory (i.e., *PROMPT PG*) and then load several useful utilities into drive D. Once in the RAM disk they pop into action instantaneously since they are called from memory, not from the disk.

```
ECHO OFF
PATH=D:\;C:\DOS;C:\BATCH;C:\UTILITY;C:\LIBRARY;
CLS
ECHO   IF FOUND, RETURN TO:
ECHO   JAMES A. BERG
ECHO   555 MAPLE BLVD.
ECHO   ANYTOWN, U.S.A.
ECHO   (555) 555-5555
ECHO   REWARD
SET COMSPEC=D:\COMMAND.COM
PROMPT $P$G
COPY C:\DOS\COMMAND.COM D:\
COPY C:\DOS\MORE.COM D:\
COPY C:\UTILITY\DDIR.COM D:\
COPY C:\UTILITY\MEMORY.COM D:\
COPY C:\UTILITY\WHERE.COM D:\
```

128K isn't much memory, but it is useful for custom tricks like these. I also use it when a program, such as a spreadsheet, uses several overlay files. By loading the spreadsheet with a batch file that copies the *.OVL* files to drive D and changing the program's set-up options to look for them there, I do not notice any appreciable delay when the program switches from one overlay to the next. Of course, they must all fit in 128K for this to be helpful.

continued on page 13

NEW PRODUCTS

TMN ASSEMBLER

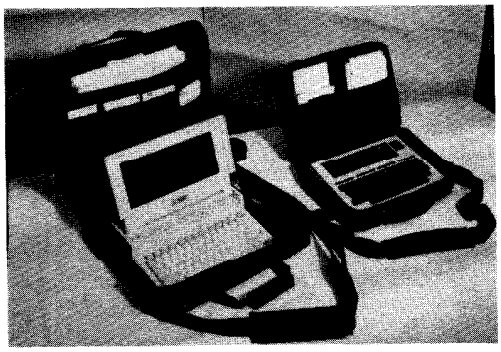
Tri-Mike Network East has acquired all rights to the HSI Assembler for Tandy and NEC notebook computers.

The TMN Assembler requires less than 3K of RAM and is relocatable. Assembles source code from any file or device, thus eliminates "limited memory" problems. Accepts data in hex, decimal, or ASCII. Creates a ready-to-run .CO file directly on the menu or an optional trial assembly. Six built-in macros make programming even easier.

Output all or any portion of the assembled listing to screen or printer, with optional user-inserted pauses. A 56-page manual covers the use of the assembler, the 8085 instruction set, sample programs, information on ROM calls, and a map of reserved high memory.

Available on cassette and priced at only $35.95 complete (includes shipping and handling). For more information, contact Granite Street Portables, P.O. Box 651, Peterborough, NH 03458-0651. *Or circle #73 on your Reader Service card.*

TWO TOP CASES FROM TARGUS



Rugged, lightweight, and versatile Targus cases—well-crafted protection for your laptop and accessories. Comfortable to carry and good looking, too.

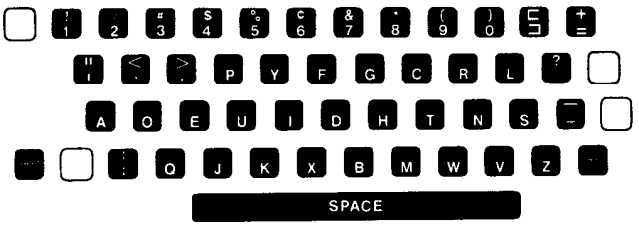
Targus, Inc. the leading manufacturer in carrying cases for portable and laptop computers, offers two fine cases for Tandy laptop and notebook users—the Lappac 2 Deluxe and the Lappac 3.

Rugged "Ziticone" treated 840 Denier Oxford Nylon outer shells provide unsurpassed durability and a non-abrasive, water-repellant, scuff-resistant surface. Padding is provided by impact absorbing, high-density, closed-cell foam covered by a protective inner lining. A 200-lb. burst strength webbing is stitched to nautical standards, reinforced with industrial rivets on all stress points. Special non-skid, shock absorbing rubber feet are securely glued and stitched to the bottom of each case. Targus products use free running, self-healing nylon zippers plus molded positive close, non-corrosive fasteners. And all Targus cases feature cushioned non-slip shoulder pads and foam padded hand grips for user comfort.

The Lappac 2 Deluxe, ideal for the Tandy 1400LT, incorporates a carrying case, a briefcase, and a portfolio into a single portable system. The main section accomodates essential computer equipment, together with an adjustable or removable divider to hold power supplies, cables and accessories. A separated portfolio compartment features individual disk pockets, pen holders, business card holder, and document pouch. Exterior zippered pockets provide space for notebooks, manuals, etc., and a padded, gusseted pocket holds a portable printer, modem, or a notebook computer.

The Lappac 3 for Tandy, NEC, Kyocera, Olivetti, and Cambridge notebook computers, offers the same quality materials in a smaller, more convenient configuration. An exterior 2.5-inch wide pocket with a hook-and-loop flap provides generous space for manuals, cables, power supplies, portable printers, disk drives, and other accessories. Two document pouches inside the case hold notebooks, pens, and other papers.

For more information on these and other Targus cases contact Targus, Inc., 6190 Valley View Drive, Buena Park, CA 90703 (714) 523-5429. *Or circle #74 on your Reader Service card.*



The scientifically designed Dvorak keyboard layout improves typing efficiency up to 90 percent while decreasing errors and fatigue. Dvorak Keyboard System *software puts this "Rolls-Royce" of keyboards on your Tandy or NEC notebook computer.*

DVORAK HAS A NEW DVERSION

Tri-Mike Network East has released a versatile new, single-program version of its tried and true *Dvorak Keyboard System* software for Tandy and NEC notebook computers.

DKS lets notebook computers use the vastly more efficient Dvorak keyboard arrangement for faster, easier typing with far less effort than the traditional keyboard (designed in 1873!). And learning to type is much easier on a Dvorak keyboard.

Studies show that while an average typist's fingers travel about 16 miles per day, a Dvorak typist accomplishes same work with only *one mile* of motion. It's no wonder this fast, efficient keyboard is used by directory assistance operators, military and government agencies, and thousands of others—including the world's champion typist!

DKS requires no hardware. Installation is simplicity itself—type a single command. The Dvorak filter installs invisibly, safely and permanently, in only 300-600 bytes of RAM and works in BASIC, TEXT, TELCOM, etc.—like the machine came from the factory that way. Here's what reviewers say:

"...makes the Model 100 into a powerful Dvorak-equipped system which you can take almost anywhere. Fantastic."—*Dvorak Developments*, Summer 1988

"...let me maintain my ultra high Dvorak speed and took up too little RAM to notice."—*PICO*, Nov. 1987

"...seems an ideal program. Laptop computers need more such software that is so easy to use and so well behaved."—*Online Today*, July 1987

Along with other enhancements, the new *DKS* is now a single program for even easier use and storage. When stored in the personal file area of a BBS or major on-line service, it's immediately available after a cold start or similar disaster while traveling.

Dvorak Keyboard System is available for $32.95 (disk) or $35.95 (cassette). Prices include shipping and handling. For more information contact Granite Street Portables, P.O. Box 651, Peterborough, NH 03458-0651. *Or circle #72 on your Reader Service card.*

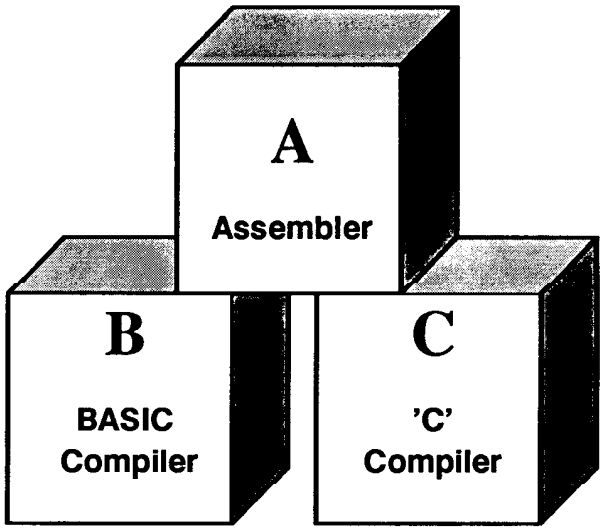
LESS TAXING EFFORT

Software Versand is shipping their Model 100 *WEETAX*™ system of integrated programs for periodic maintenance of finances for personal income tax returns. Keep your records organized and verified. Designed to standardize your reporting format easily and quickly, *WEETAX* helps you avoid the year-end crunch.

The scrolling input panel is table-driven: field labels and default data come from a user-supplied, tax-category *TEXT* file. You specify which categories are to be supplemented with an audit trail of prompted, dated annotations. A 377-byte RAM-resident control program automatically moves supporting programs and data between RAM and disk. The paginated printout is designed for attachment to an annual IRS return and is also available as a year-to-date summary. A reset feature clears all data files but keeps the organization—all ready to begin the next calendar year.

The DVI version retails for $44.95 (ppd); the PDD version is $49.95 (ppd). For further information, contact Software Versand, Points West No. 158, 3131 West Cochise Drive, Phoenix, AZ 85051 (602)997-1523. *Or circle #65 on your Reader Service card.*

OPTION ROM DEVELOPMENT FOR THE MODEL 100/102 IS AS EASY AS . . .



These Cross Development Packages for the Model 100/102 that run on a PC or Clone will make your development project a breeze, whether you work at machine level in Assembler, produce fast maintainable code in 'C' or just want to get your BASIC program to run from an Option ROM.

ASM 100

Package includes a Cross Assembler and linker for the Model 100/102 that allows you to create a .CO or ROMable program.
The Option ROM Library makes it possible to write and test code as .CO files and then burn the final product into an Option ROM. Manual contains detailed information on the Model 100/102 option ROM technology.

$395.00

RBASIC

A Basic to ROM Compiler that allows your BASIC programs to be converted so that they can be run directly out of an option ROM.
At last a simple and inexpensive solution to putting your BASIC program onto a ROM.

$199.99

- C100

The 'C' Cross Development System is a portable, easily maintainable compromise between the slowness of BASIC and the complexity of Assembler. Complete libraries provide Telcom, Graphics, screen control and random record access.

$695.00

The Secrets of ROM Revealed

Concise, easy to understand manual covers in detail, previously undocumented Option ROM features of the Model 100/102. Includes listings of everything you need to call standard ROM routines from an Option ROM. IBM Diskette containing source code of key routines and a search & replace utility for text files.

$39.95, Diskette $10.00

Eproms, Boards, and Eprom Burners available.
Please send check with order or we can ship COD.
California residents add 6.5% sales tax.
$3.50 shipping per item inside the US.
Rush shipping and handling charge $15.00

King Computer Services, Inc.

1016 North New Hampshire, Los Angeles, CA 90029. (213) 661-2063

Circle 111 on reader service card.

COMPATIBILITY: Tandy 100/102

Weetax

A Weekly Tax (Record) Collector

by Don Wood

Now that you either have gone—or will go—through the agony of preparing your tax return, what will you do to make next year's effort a little easier? Just as I ponder (again) how I might personally solve this problem, the phone rings, and I am asked if I would like to try a tax program. I think, "No, not another tax program!" But with the recent tax preparation experience still fresh in my mind, I relent. Sure, why not?

Well, this is *not* just another tax program (sigh of relief). I have tested many of those expensive and hard-to-use programs for filling out your return, but filling out the return is really the easy part. The hard part is getting all those records itemized and totaled. If you are like most people, you will throw everything—maybe—in an old shoe box until tax time, when you will try and figure out which receipt really went with what. And if you are like some other people, you will keep some record of each potentially deductible item on a pad or ledger. Of course, you still have to total them.

Figure 1. The Weetax *main menu shows the various options available.*

If you don't really like either of those options, you may like this program. It is called *Weetax*, which apparently stands for *WEE*kly *TAX*. The program is designed to help you in your record keeping effort by collecting the information as it happens.

SETTING UP

Well, it sounded good to me, so I tried a pre-release copy of the program. When it arrived, I opened the twelve-page, single-sided, comb-bound manual and discovered that I needed a copy of the *Disk Power* disk operating system (DOS). *[Not anymore—see editor's note.—Ed.]* I acquired one and decided to install it on my Booster Pak equipped Model 100. Well, at least I thought is was a good idea. During installation my Booster Pak muttered something too quickly for me to see and then cleared the screen and vanished into the ether.

Some of you know that strange feeling when you find yourself staring at the cold start screen. After I recovered from the initial shock, I realized that one of the nice features of the Booster Pak is that a cold start does not erase the files. So I located the Booster Pak manual, performed the magic listed therein, and everything was back to *NoRmaL* or so it appeared. To make a somewhat longer story short, it wasn't. So I reached for my backups (I really did have some), formatted the Booster Pak and reloaded.

FALSE START

After this excitement, I called the software company to see if *Weetax* could be operated without *Disk Power*. The program's author explained that *Disk Power* was used to allow the *Weetax* program to be broken into pieces (i.e., smaller programs) that load from disk as needed. He told me these programs were loaded and run by *RUN "0:filename"* statements within the programs, where *0:* specifies the disk drive.

Knowing my Booster Pak as I do, I said "no problem;" it allows the same statements to work. So, I loaded the main *Weetax* program into memory and executed it. It displayed a menu of choices and then said something about *File not found*. Being familiar with *BASIC*, I looked at the program and discovered it was trying to *RUN* a program from the floppy disk. I looked on the floppy disk and could not find any *.BA* programs other than the *Weetax* main program. But, I did find a *.DO* file with the desired name. Another call to the author and I found that in order for *Disk Power* to run programs from the floppy disk, they must be in ASCII (*.DO*) form. Since my Booster Pak has no such requirement, I converted all of the *BASIC* programs from *.DO* form to *.BA* form on the disk.

Now I'm cooking with gas. I finally got a full menu (Figure 1) and things appear to be working. So let's put in some information.

ROLLING ALONG

You start by letting the program know what categories you will be keeping track of (i.e., charities, medical etc.), the default amount that each normally incurs weekly, and whether an audit (comment) should be required. You do this by editing a file called *TAXCAT.DO* and changing, adding and deleting lines, using the format shown in the *Weetax* manual. Putting a * beside the category name will cause a comment to be requested when amounts are entered.

Having done the editing, I again start the main program, *WEETAX*. I see from the manual, that each week you're supposed to take about five minutes and record that week's transactions. (Actually, it works best if your recording

frequency matches your paycheck frequency.)

Well, the first step appears to be *F1 (Load Files)*, so I press *F1* and then discover another feature of the program—entertainment. During the loading of files, apparently random (musical?) notes are played. I later found that the more files you have, the more "music" you hear. It would probably attract a lot of attention in the library or on an airplane. Maybe it's subtle advertising. I could find no command to turn it off. Anyway, the music has stopped now and the next step is to select the *Update* choice from the main menu.

This presents a full-screen display (Figure 2) showing the first eight of your categories and the default amount that will be posted for each. Across the bottom are the choices for this menu. *Up* and *Down* page though the categories in groups of eight, beeping when it reaches the top or bottom of the list. You may choose *Change* to adjust the amount that will be posted to the current totals, either by typing a new value or an incremental (+ or -) amount. *Add* will post the amounts displayed to the totals. *Exit* will return to the main menu, and *Files* will take you out to the RAM files so you can edit *TAXCAT.DO*, if necessary, and then rerun the *WEETAX* main program (the data files will still be loaded).

```
gr gross    | ss fica     | fd fedl     | st state
np net      | gr insur    | lt disab    | em expen

gr 673.38   | ss  41.28   | fd 107.66   | st  40.84
np 471.92   | gr  11.88   | lt   2.62   | em*  0.00
  Last Upd: 02/25/89    Today: 03/01/89
u(P),d(WN),c(HNG),a(DD),e(XIT),f(ILES):
```

Figure 2. The update menu displays the default amounts for eight categories at a time, allowing you to modify and post any changes.

I select *Change* and then enter a category code and a new amount. I continue this routine, selecting *Change*, typing the two-letter category code and the new amount or adjustment, until all categories show the amount I want to post for this pay period. When the amounts are correct for this period, selecting *Add* will post them to the year-to-date totals. Some of the categories may have the audit trail option selected, and this will cause the program to ask you for a description of the amount for that category.

This was one thing I found that did not match the way things normally occur for me. I may have more than one charitable contribution during the week for which an audit trail comment is desired. After some thought, I realized I could accomplish this by setting the default for each category to zero, so only the categories which had non-zero amounts would post and possibly request a comment.

After making that change to the *TAXCAT.DO* file I post the rest of my transactions. During this effort I discover that the program will let you post the same entries more that once without objecting. This could allow errors caused by errant fingers or keyboard bounce.

OOPS!

Then the door bell rings...another contribution. I go immediately to my trusty Model 100 to enlighten *Weetax* about this transaction. The program is sitting at the main menu, so I press *F1 (Load Files)*. Oops! I realize that the files were already loaded, and I had made changes and not saved them. Well, *Weetax* does not seem to mind; it is busy carrying out my request to load the files again. Once the files are again loaded, I have to post my first batch of changes again, adding this most recent contribution. All goes well, and I decide to save the files before repeating my previous mistake.

Continuing to test the software, I print a summary of my work. The listing includes the year summary and a printout of the reasons for each amount in categories where an audit was requested. Everything comes out all right, except an extra sheet is ejected. (The program sent two form feed characters to my printer.) Feeling brave now, I decide to find this in the program and delete it. No problem, since the program is structured and has comments. Apparently the author has a DMP-2100 printer (according to the comments), which explains the need for the extra form feed (he has a pull-type forms tractor).

GOING FOR SPEED

Everything else goes well, just more slowly than I would like. My confidence is really up now, so I decide to make the program run entirely on my Booster Pak (which uses *R:* instead of *0:* to reference disk files). Again, no problem at all. Since the programs are well structured, all file references occur only once. Within five minutes I am finished, and *Weetax* runs much faster.

I called the author once more to relate my experience. He expressed great interest, and later I received a letter documenting my call and stating what would be done regarding each of my comments. Most of the changes I suggested would be implemented prior to first shipment of the product. So by the time you read this, most of the inconveniences will be gone, and the product will run on more environments than just *Disk Power*.

THE BOTTOM LINE

In summary, *Weetax* delivers on its promise, and especially with the upcoming changes, this program provides a simple solution to record keeping. Since the Model 100 goes with me as I travel, my tax records can now easily be kept up to date. One more time consuming job eliminated!

Now if we could just write a program to eliminate the IRS.... ●

Editor's note: As of publication time, Weetax *now supports other DOS's and devices, such as Booster Pak and Gold Card. Other changes have been made as well, including an option to disable the sound.*

-MN

Manufacturer's Specifications

Weetax™—$44.95 (DVI), $49.95 (PDD)
Software Versand
Points West No. 158
3131 West Cochise Drive
Phoenix, AZ 85051
(602) 997-1523

BASIC Bits

A Source for Model 100/102 Public Domain Software

By Thomas L. Quindry

If you read the January *"BASIC Bits"* column, remember the controversy caused by the selection of programs I have presented in my column and offered on a distribution disk. After much thought, I have reached a decision, subject to change based on public opinion, regarding the column and distribution disks. I encourage your continued comments on this subject.

THE CONTROVERSY CONTINUES

For the most part, it is not possible for me to contact authors prior to publication of the column to clarify their intent regarding the "implied unrestricted distribution," as stated in my January column in the section called "The Legal Consensus." Therefore, I will work by exception. I will continue as I have in presenting programs and distribution unless notified otherwise by an author. This may mean a little inconvenience to readers who wish to send for a distribution disk, since I may have to remove a program from the disk if the author requests. Include a self-addressed, stamped envelope if you want the disk only if a particular program is included. I will return your check if I can't deliver that program or an alternative.

I invite authors of programs for the Model 100/200 to clarify their intent (either positive or negative) in writing to me even before I would cover their programs. And I also encourage these authors to send me a disk with their programs if they especially would like coverage and distribution through this column. Send me a dollar for return postage and I'll return the disk with other programs I have.

Regarding CompuServe, GEnie, and the others, if they would like me to credit them as the source of a program, I will be glad to. All they have to do is ask (in some official capacity) and give me some sort of access so I can check the programs out.

Mr. Wheeler has expressed his desire that I not distribute his programs. I respect his rights and they will not be distributed. After I received Mr. Wheeler's letter asking me to remove the programs I substituted two other programs. In place of *XMDPW5*, I have substituted *XMD100*, by John Chenoweth. This program was the original program from which Mr. Wheeler's program was based. Mr. Chenoweth made no restriction on his program and in fact, in a source code file called *XMD105.ASM*, which I have, encouraged others to modify and share his program even though copyrighted. In place of *BASBLD.BA*, by Mr. Wheeler, I have substituted *SAVEML.BA*, also included this month. The following is a response to Mr. Phil Wheeler's letter to the editor in January, along with other comments.

> ***Mr. Chenoweth made no restriction on his program***

RESPONSE TO MR. WHEELER

While I respect your right to restrict distribution of your programs, I cannot agree with your reasonings and inconsistency in applying them. You have stated that Mr. Chenoweth's program was copyrighted and not released to the public domain. Mr. Chenoweth, by uploading his program to CompuServe, gave "implied unrestricted distribution," which includes my method of distribution. He did not specifically restrict it to CompuServe as you have assumed. You yourself stated to me in a letter that you have been unable to contact him for clarification. So you can see it is impossible for me to confirm anything other than what is implied by his upload.

Mr. Chenoweth's program, *XMD100*, and your enhancements, *XMDPW5*, and *XMDPW6*, are already widely distributed whether you know it or not, and not by me. These programs and many others on CompuServe are also on many public bulletin boards and on GEnie also. From a directory of GEnie programs the following, *XMDPW5.DOC*, *XMDPW5.SRC*, *XMDPW5.200*, *XMDPW5.LOD*, *XMDPW5.100*, *XMDPW5.INF*, *XMDPW5.ASM*, and *XMDPW5.DVI* were apparently uploaded by a person with the GEnie address of PWHEELER between December 1, 1987 and March 18, 1988. If PWHEELER is you, it is inconsistent with your own reasoning of protecting the rights of other authors as stated in your January letter to the editor.

The original authors of programs always have the opportunity to restrict their programs regardless. You may, in fact, be unwittingly circumventing the other authors' desires for further distribution of their programs by making modifications to them and restricting distribution. You have been very prolific in modifying and enhancing the works of others and providing conversions from the 100 to 200 and vice-versa. Your improvements and contributions may have replaced the original author's programs to the extent that their versions no longer exist. It is indeed unfortunate that you have taken this position as a big chunk of software available for the Model 100 and 200 user may be lost in the process. You're wasting your efforts too, which I'm sure were to enhance the enjoyment of other laptop users.

I hope you will reconsider your position, not just for the two programs from my November column, but for all you have modified that have not been specifically restricted by the original author. Don't get all tied up with the term "public domain." No one is trying to add a label to his or her program that may not exist. Let the original author decide whether he wants to restrict distribution. Don't restrict it for him. For electronic databases (such as CompuServe), the term *copyright* does not imply a distribution prohibition.

OTHER RESPONSES

In addition to sending Mr. Wheeler an advance copy of the January column, I sent copies to Wayne Day (Sysop of the Model 100 Forum for CompuServe), Dave Thomas (Sysop of the GEnie Laptops Roundtable), and Rich Hanson (Founder of Club 100). CompuServe and GEnie apparently want to stay out of the fray with no comment. CompuServe

decided to treat it as a proposal for a new service and said they would evaluate my suggested solution of asking for CompuServe time in the January column. Dave Thomas did not respond. Only Rich Hanson answered the issue full force. Below are CompuServe's response and Rich Hanson's response for Club 100. Rich's response has been shortened to conserve space.

COMPUSERVE'S RESPONSE

Dear Mr. Quindry: This is to confirm the receipt of your proposal for a new CompuServe Information Service product. The proposal has been forwarded to the appropriate CompuServe marketing staff for evaluation with respect to our current marketing strategy, product area needs, and revenue potential. This review process may take 60 to 90 days.

Given the volume of proposals we receive, we will only recontact those individuals whose proposals are most synergistic with our current business goals.

I'd like to thank you again for your interest in becoming a service provider. Sincerely, Sharon Wychoff, Proposal Review Administrator

CLUB 100'S RESPONSE

Dear Tom: For over six years CLUB 100's position has been based on, "the intent of the author." Authors control their own work. It is not right for us, as distributors, clubs and promoters, to make legal decisions for the authors or owners of program code. It is also not right for third parties like you or us to be victims of uncertainty, politics or power struggles.

CLUB 100 operates under one rule for program acquisition and distribution. If a program was ever offered for sale to the open market then it is not in the public domain, and thus, it is not in our library. All else is fair game. It is our contention that the author, or legal owner, not the distribution service, determines whether a program is in the public domain.

The club is our interest and hobby. In fact, CLUB 100 is a registered, non-profit, public benefit corporation in the State of California. We did this to make sure there will be no question as to our intent. If anyone would like to try making a profit doing the things we do for the Model 100 community then more power to them. We work to just break even each month so that our hobby remains interesting and fun. And, we share and swap. We have openly swapped programs with yourself, the Boston Computer Society's lap computer group, and Compuserve, GEnie and Delphi sysops, sponsors and members. We have swapped mailing lists with commercial firms, and selected lists with new user groups, like the Heart of Illinois Model 100 User Group. We share ... we swap ... we support ... we have fun! We are here for the development of all M100, T102 and 200 owners, world wide.

Our recommendation is this: If authors do not want their programs considered "in the public domain" they should keep their programs out of accepted public domain distribution channels. If they [authors] feel that their code is worth money then they should advertise it for sale; making their intent clear. The *Portable 100* Magazine Classified Advertising offers a great place to start. It's working for us.

If commercial online services are concerned, for whatever reason, they should insist that members involved in development work restrict their work to private e-mail and not public forums.

Our community is no longer in the spot light. Support and new products are almost gone. *Portable 100* magazine remains the undisputed foundation of Model 100, 102, 200 and 600 support. Remaining support exists in the underground we create through sharing our knowledge with each other. When the sharing stops, our community stops... sharing stops when tails get short... tails get short through ignorance, insecurity and greed.

Tom: You just keep on doing *"BASIC* Bits," and distributing your disks in any manner you feel is right for our community. You, and *Portable 100* magazine have the complete support and admiration of our club and its membership. Regards, Richard Hanson, Founder, President, sysop

WRAP-UP

Now CompuServe leaves the door open for consideration without any promise of a response. If you are a CompuServe user, how about getting a discussion going there so we can see what develops. Also, how about you GEnie, Delphi, and Source users? I for one would like to see this issue resolved in the open. How about you?

THIS MONTH

I apologize for taking so much space discussing the issue of the types of programs discussed here, though it is one that needs resolution. My descriptions of programs will thus be short so you can retrieve some value out of the column. Programs described this month will be useful to the *BASIC* programmer. They will help you squeeze out those wasteful bytes. Some of these programs may also run on a Tandy 200. I will give my opinion on this when applicable, but I would suggest that you have everything backed up in your Tandy 200 before trying them.

AUTO#.BA AUTHOR UNKNOWN

Auto# gives you an auto line numbering facility when you write your *BASIC* programs. The program automatically gives you the specified first line number and increments to the next line number and as you enter your *BASIC* statements, they are saved in a *.DO* file. This takes away some of the flexibility of your programming if you want to edit a line already entered, because you can't do it right away. You have to exit the program (with *CTRL-C* and go into the text file). Nevertheless, it is a nice little utility. This program should also work on a Tandy 200.

DATE$.BA BY T. L. QUINDRY

This is a handy little utility that I wrote for setting the date and time on my Model 100. Especially when using *TELCOM*, at certain times the year gets advanced. If you run *Date$* and just press the *ENTER* key for the first prompt, the year is set to 1989. In later years, you will have to edit line 10 in *DATE$.BA* to change the default year. This program should also work on a Tandy 200.

HISTLD.BA BY TOM BENNETT, POLAR ENGINEERING AND CONSULTING

Histld and its related file *HISTPT.BA* comes from the Club 100 library of public domain programs. *Histld* is a loader for a machine language program called *HISTO.CO*, which will help you find out the time hogs in your *BASIC* program.

PUBLIC DOMAIN

Just enter an extra line temporarily in your program, *RUNM "HISTO"* and run your program. When you finish, you *RUN "HISTPT"* and it gives the statistics relating to how much time is spent on each line of your program. You can see it graphically or else save the stats to a file. This program will not work on a Tandy 200 because it uses machine language.

KEYBD.BA UNKNOWN AUTHOR —MODIFIED BY T. L. QUINDRY

Keybd is a short program that demonstrates pseudo-keyboard input. You can use this technique in your *BASIC* programs. It puts characters into the keyboard buffer so that keyboard-like entries can be made automatically after your program is finished. You can enter a maximum of thirty-two characters. The first character has to be a space. It's interesting and does have some applications, as you'll see in this demo file. When you run this program, it places characters into the buffer, entering the commands *NEW* and *CLS*, creating a one-line *BASIC* program, and running it. This program will not work on a Tandy 200 but could be easily modified by someone who knows the address of the Tandy 200 keyboard buffer.

KEYWRD.BA BY T. L. QUINDRY

This is just a simple little program that reads the keywords from the *BASIC* ROM and prints them, formatted, on the screen. It is written so you can edit the program to print the keywords to a file if you prefer. This program will not work on a T200 but could be easily modified by someone who knows the address of these *BASIC* keywords in the Tandy 200.

LENGTH.BA BY T. L. QUINDRY

An article on *LENGTH* appears in the September '87 issue of *Portable 100*. It is public domain, Model 100 only, and I have included it here since it fits in with the theme of this article. It creates a machine language program to give you the length of each file in RAM in addition to just about everything else you would like to know about memory usage.

RENUM.BA BY RICH YOUNG

If you have ever wanted to renumber your *BASIC* programs, this will do the trick for you. You must first save your *BASIC* program as a *.DO* file. Though still slow, this is one of the quickest renumbering programs for the Model 100 that I have come across. Still, some programs have *BASIC* statements too long for this program to process. *Renum* is from the Club 100 Library. This program should also work on a Tandy 200.

Let me give you a little tip here. If you have another computer such as a TRS-80 or an MS-DOS PC, you can port your program in ASCII over to that computer using a null modem cable and a communications package and use the renumbering program that comes with DOS for that computer. Just load the program into *BASIC* and enter the command, *RENUM*. Save your program with the command, *SAVE "filename",A* where the *A* saves it as an ASCII file. Then port it back to the Model 100.

SPOUT.BA BY RON GOLINI

Spout is also from the Club 100 Library. As with *Renum*, you must first save your *BASIC* program as a *.DO* file. *Spout* removes extra spaces in your program. Unfortunately, it also removes spaces in a *REM* statement. By removing extra spaces, you will save precious memory, and also your program will run slightly faster. This program should also work on a Tandy 200.

TOOLKT.BA BY PETER STANWYCK

Another program from the Club 100 Library, *Toolkt* provides the functions of *Renum* and *Spout* in addition to a few others, seven functions in all. I included *Renum* and *Spout* in case you didn't want to load this much larger program. You can print a somewhat structured listing of your *BASIC* program. You can also send to a printer a list of which lines branch to other lines. The author calls this *Fyte the Byte*. You can reset your function keys back to their original definitions using this program and also determine the the checksum value of a *BASIC* program. Last of all you can enter data into either an existing text file or a new file. It appends your text to the end of the file. This program should also work on a Tandy 200.

SAVEML.BA BY T. L. QUINDRY

Saveml saves your machine language programs in the form of an ASCII file that you can load into *BASIC* and run to create the original machine language program. The purpose; of this program is to create a *BASIC* loaded program so you can distribute public domain and shareware machine language programs via bulletin boards or direct computer-to-computer transfer using a null modem cable. If you use a desktop computer to store your Model 100/200 files in ASCII format, then *SAVEML.BA* will be useful to you. It runs on both the Model 100 and 200.

SCR200.BA BY T. L. QUINDRY

Scr200 is just a little bonus program I threw in here. It is a Tandy 200 version of my *Scripy Text Processor* described in the January '89 issue of *Portable 100*. If you would rather have one of my other *Scripy* versions for the Model 100, either the regular or DVI (Disk-Video Interface) version, let me know when you ask for the disk I distribute.

VARLST.DO BY WOODS MARTIN

Varlist is a subroutine you merge, using the command *MERGE "VARLST"* at the end of a *BASIC* program. Your program must not have any line numbers greater than 58999. After you run your program, you enter the command, *GOTO 59000*, and all the *BASIC* variables your program uses will be displayed. If your program normally exits to the *MENU*, you must use *CTRL-C* to stop it. If your program uses the *CLEAR* function or *RUN* command from within it, only variables used after the *CLEAR* or *RUN* will be displayed.

FINAL NOTES

As a convenience to readers, this month's programs will be made available to readers on a 100K TPPD1 disk for $7. To order the programs described in this column or any other *"BASIC* Bits" column, send $7 (includes packaging and handling) along with the month and year of the column. The cost per disk is not to pay for the programs but for the cost of distribution. If you would rather have the programs in ASCII format for Radio Shack Model 3 TRSDOS or MS-DOS, please specify them. The cost is the same. A self-addressed label will speed your order. Send to Thomas L. Quindry, 6237 Windward Drive, Burke, VA 22015, Attn: March 100.

DEFUSR appears monthly to answer your questions about Tandy notebook computers.

Send your queries to: DEFUSR, PORTABLE 100, P.O. Box 428, Peterborough, NH 03458-0428. Please enclose a stamped, self-addressed envelope for our reply.

DERANGED KILLER

Upon returning to the menu, *KILL.BA* (*P100*, Jan. '88) appears to generate five carriage returns, *CHR$(13)*, leaving the computer in *BASIC* with four carriage returns having been executed on the screen. No harm done. The menu is only an *F8* away, but I'm sure it needn't happen. Can you check into this? Will this invisible program count toward the maximum number of files allowed? Can I delete it while it is invisible?

Stephen W. Lees
Forest Hills, NY

KILL.BA's strange behavior is caused by the POKE AD,10 *statement in line 2 erroneously telling the keyboard input routine to expect ten characters (the five in* "Menu" + CHR$(13), *followed by five of whatever remains in the keyboard buffer from a previous operation). In tests here the glitch never appeared, because another program in my machine constantly clears the buffer.* KILL.BA *will behave if you change the statement to* POKE AD,5.

An invisible file counts toward the allowable maximum and is killed like any other, using BASIC's KILL "filename.ext" *statement. Type the filename exactly as it exists—if there's no extension, don't type one.*

-MN

NEC DOS?

I have a Model 102 and a NEC 8201A. Will my Tandy Portable Disk Drive 2, which I use with my Model 102, also work with the NEC?

I am a teacher and am especially interested in educationally related programs and information. I thoroughly enjoyed the September issue but must still have typing errors in the programs *LEARN.BA* and *STUDY.BA*, so I decided to go ahead and order your disk.

I am also wondering if there is a way to transfer programs directly between my two computers.

Diane Miller
Madang
Papua New Guinea

Yes, Diane, the PDD-2 will work with your NEC, using Traveling Software's NEC version of TS-DOS (see their ad).

The teaching programs work as published, but the P100-To-Go *disks do make it much easier by avoiding those nasty typos. We have more education programs on the way.*

You can transfer files between machines

Will my Tandy Portable Disk Drive 2 work with the NEC?

via a null modem cable or adaptor. You can purchase an adaptor from Radio Shack or make your own null modem cable ("Ending Your Computer's Solitude," Sep. '88). With the machines thus connected, you can use two methods.

One is to use TELCOM *in both machines with matching RS-232C "stats" (I use* 98N1E *on the Tandy and* 9N81XN *on the NEC) and simply upload from one while downloading to the other. This method transfers* .DO *files only, its main advantage being that it also works with other machines, like desktop computers, which have some form of telecommunications software or a batch file ("Quick, Easy and Cheap Model 100 to MS-DOS Computer File Transfers," July '88).*

Using two notebook machines, it's easier to do directly from within TEXT. *For example, let's send* MYFILE.DO *from the NEC to the Tandy. First, on the Tandy, go into* TEXT *and create an empty file named* MYFILE.DO *(or any name you wish), press* F2 *(Load), type* COM:98N1E *and press* ENTER. *Now on the NEC, go into the* MYFILE.DO *file, press* F3 *(Save), type* COM:9N81XN *and press* RETURN. *After a brief pause, you'll see the file appear on the Tandy's screen. Press* F8 *on the Tandy and* SHIFT-F5 *on the NEC, and you're done.*

You can similarly transfer .BA *programs. For example, let's send* MYPROG.BA *from the Tandy to the NEC. First, go into* BASIC *on both machines. Now on the NEC, press* F1 *(Load), type* COM:9N81XN *and press* RETURN. *On the Tandy, press* F2 *(Load), type* MYPROG.BA *and press* ENTER *to load the program into* BASIC. *Now to send it, press* F3 *(Save), type* COM:98N1E *and press* ENTER. *When the file has been sent, the* Ok *prompt appears on both machines. Now on the NEC, save it as a* .BA *program: press* F2 *(Save), type* MYPROG.BA *(or any name you wish) and press* RETURN.

A word of caution: Since Tandy and NEC BASIC *languages differ in many respects, usually only "plain vanilla" programs work in both machines without some modification. Until you're sure a transferred program works, back up important files before running it.*

-MN

NO-TAB NEC

I imagine you're read by many other NEC'ers. Using an NEC PC-8201A, I value that you at times include as inserts the ROM addresses or programming details for the NEC.

In your reply to Ted Knyszek (Nov.

DEFUSR

'88, p. 26) you refer to the ROM *PNOTAB* routine. Can you print the relevant NEC 8201A ROM address, because I have also encountered that eight *CHR$(32)*'s for a *CHR$(9)* as a nuisance, unable to get around it.

G. van Wessem
Ferntree Gully, Australia

My ROM disassembly suggests 6365 (18DD hex) as NEC's equivalent function. Unlike Tandy's CALL *statement, NEC's* EXEC *won't pass parameters. Instead, you* POKE *desired register values into special RAM locations beforehand (see p. 4-43 of your* BASIC *manual). Since* PNOTAB *expects the printer character's ASCII value to be in the A register,* POKE 63911,value: EXEC 6365 *should do the trick.*

-MN

TSK! TSK!

Nuge, I'm surprised at you! I just bought a Tandy PDD-2 portable disk drive to replace my PDD-1. I looked up your programming sidebar on DOS patches (Sep. '88 *Portable 100*) where it said that *FLOPPY* for the PDD-2 can't be saved to tape or disk. TSK! TSK! Use the utility *CHANGE.BA* to rename *FLOPPY* to *FLOPPY.CO*, enter *BASIC*, type *LOADM "FLOPPY.CO"* (press *ENTER*), and make your changes. Then you can *SAVEM "NEWDOS.CO", 60000, 62737, 62701* and press *ENTER*. You can now save *NEWDOS* to tape or disk, as fitting. Just remember to *CLEAR 256,60000: LOADM "NEWDOS.CO"* and then *CALL 62701* to use *NEWDOS.CO* when you've cleared high memory for other uses.

I then used *CHANGE.BA* to make *NEWDOS.CO* invisible so I wouldn't accidentally kill it. If *NEWDOS.CO* isn't resident in memory, *CALL*ing 62701 will buy you a one-way ticket to the FROZEN NORTH ...

There you go; that wasn't so bad.

Robert J. Holland, Sr.
Sayre, PA

Yeah, okay. (Picky, picky, picky!) You're right, though. After saving NEWDOS.CO, *rename* FLOPPY.CO *back to* FLOPPY, *run* FREMEM.BA *to kill* FLOPPY, *and then kill* FREMEM.BA.

Perhaps a safer way to run it is to type CLEAR 256,60000: RUNM "NEWDOS" *instead. That avoids accidents by eliminating the need to remember to* LOADM *the program. If you can spare a directory slot, make that line into a small* .BA *program, and then you needn't remember the* CLEAR *addresses, either.*

Now that you've opened this can of worms, there are ways to further improve NEWDOS, so take a look at "DOS Patches III" in this issue. Thanks for getting on my case, Robert!

-MN

MAC CONNECTION

David Rowell's articles on using a null modem cable to link the Models 100/102/200 to other computers have been particularly good reading and very informative. I've found it easy to connect my M100 with various TRS-80's, and it's the perfect accessory for my Amiga—but I've been trying for weeks to transfer ASCII files from a Macintosh.

Three computer dealers claim it can't be done.

The Macintosh Plus, SE, and Apple II GS all use a mini-round 8-pin jack for the serial port. The pins are labeled in the instruction book as:

1	HSKo	Handshake out
2	HSKi	Handshake in
3	TXD-	Transmit data
4	GND	Signal ground
5	RXD-	Receive data
6	TXD+	Transmit data
7	GPi	General purpose input
8	RXD+	Receive data

I've called three computer dealers asking for a null modem cable only to be told it either couldn't be done, or I'd have to have a custom cable wired.

Would you run a diagram of the connections to be made for a null modem cable to go between the laptops and the Apples?

Dick Kahoe
Warrensburg, MO

Two solutions. A) Use a Mac-to-Modem cable (as sold by Apple) with a standard null modem adaptor to switch the lines, or B) use an Imagewriter I cable (if you can find one) to connect directly to the Tandy laptops.

-TK

EURO-DATE$

I have a very specific querie. You see, in Europe, dates are expressed in DD/MM/YY format, and so are the *DATE$* on European-made Model 100's. Now most of the listings we get in *Portable 100* are obviously written for the U.S. M100, and those which deal with *DATE$* never happen, due to incompatibility bugs.

So if you could explain how to modify the *BASIC* statements in order to cope with our European way of expressing *DATE$*, I am sure it would help to make more useful the maze of listings which, written in the U.S.A., deal with *DATE$* as MM/DD/YY. Many thanks!

Jean-Luc Michaud
Chertsey, Surrey, England

In general, programs deal with DATE$ *by dividing it into month, day, and year components, usually by means of* BASIC's LEFT$, MID$, *and* RIGHT$ *functions.*

LEFT$(DATE$,2) *or its equivalent,* MID$(DATE$,1,2), *returns the first two characters of the date string.* MID$(DATE$,4,2) *returns the middle two characters.*

RIGHT$(DATE$,2) *or its equivalent,* MID$(DATE$,7,2), *return the last two characters (the year). Since that's the same in both U.S. and European formats, you needn't concern yourself with them.*

That leaves only the month and day to deal with. Since European and U.S. formats simply reverse these portions of the date string, just change all occurrences of LEFT$(DATE$,2), *or its* MID$ *equivalent to* MID$(DATE$,3,2). *And conversely, change occurrences of* MID$(DATE$,3,2) *to* LEFT$(DATE$,2) *or its* MID$ *equivalent. For example, suppose you want to change* A$="MM/DD/YY" *to the European* "DD/MM/YY." *Just rebuild the string like this:* A$=MID$(A$,4,2)+"/"+LEFT$(A$,2)+"/"+RIGHT$(A$,2).

While probably not the most efficient means of translation, it's a starting point. Perhaps readers can suggest alternatives.

-MN

Unpuzzling: The Solution

An addendum to your Tandy 200 manual, plus the solution to our PUZZLE.BA program.

When I first figured out how to access RAM locations in the alternate banks of the Tandy 200, I set certain goals for myself. I was determined to bring cross-bank access to *BASIC, TEXT, TELCOM,* and of course, to the main menu. I feel that I have accomplished (in a limited way) what I had originally set out to do, but I would like to think that it doesn't stop there.

I hope I have piqued the interest of some accomplished programmers who can use the ideas I have presented and expand what I've done. I tend to move to something new immediately after revealing my latest efforts, and I don't really develop my projects to their fullest potential.

The work that has appeared in "CUSTOM 200" was presented for the purpose of demonstrating the significance of what I've found, and by no means should it be considered "perfect." I'll be the first to admit that my projects end when they *appear* complete and that they are presented after some successful beta testing.

NiCd INSTALLATION

This month I am not offering software for the Tandy 200 but instead would like to focus a bit on the hardware. In the four years that I've had the Tandy 200, I have only had to change the batteries twice. But when I replaced the second set of batteries, I decided to install NiCd's (nickel cadmium rechargeable batteries).

The modification for installation is quite simple (according to the documentation). Just install jumpers J301 and J302 on the memory PCB (printed circuit board). The memory PCB is the circuit board that holds the option RAM and ROM sockets. To access that PCB requires removing the bottom of the Tandy 200. The memory board is between the keyboard PCB and the bottom cover. Once you install the jumpers, you can reassemble the unit.

Simple enough? Well here's where Murphy's Law holds true to form! I did the modification, charged up the NiCd's by leaving the unit plugged in overnight, and in the morning proceeded to use the Tandy 200 as usual, except that I kept it on battery power to see how long those NiCd's would let me work without the AC adapter.

Within three hours the low battery light came on. I had been getting at least six hours of use with fresh alkaline batteries, and although the NiCd's were only 1.25 volts when charged (instead of the 1.5 volts of fresh alkalines), I thought I should get more computer use than just three hours.

Well, I was right. Those NiCd's kept the Tandy 200 powered for over two more hours before the low voltage sensors turned the unit off. I might have gotten even more time, but I had used the internal modem, which drains the batteries more rapidly. In any case, I realized that if the warning LED came on two or more hours before auto-shut-off, then what purpose does it serve?

Since the Tandy 200 worked fine until it powered off, I concluded that the only problem was the bias in the LED circuit.

```
Ø REM PUZZLE SOLUTION BY PAUL GLOBMAN
1 REM              COPYRIGHT (C) 1988
2 ONERRORGOTO55:CLS:DIMLC(2Ø):C=1:FORI=Ø
TO9:READSQ(I),TY(I):NEXT:FORI=1TO4:READX
X(I),YY(I):NEXT:FORI=ØTO9:GOSUB45:NEXT:I
=9
3 RV$=CHR$(27)+"p":GOSUB45:GOSUB24
4 READA:IF(A=>48)AND(A<=57)THENRV$="":GO
SUB46:I=VAL(CHR$(A)):GOTO3
5 IFA=28THEN13
6 IFA=29THEN1Ø
7 IFA=3ØTHEN17
8 IFA=31THEN2Ø
9 BEEP:GOTO4
1Ø IFSQ(I)MOD5=ØTHEN9
11 SF=Ø:GOSUB28:IFSF=ØTHEN9
12 GOSUB44:SQ(I)=SQ(I)-1:GOTO3
13 IFSQ(I)MOD5=4THEN9
14 IFTY(I)MOD2=ØAND(SQ(I)+1)MOD5=4THEN9
15 SF=Ø:GOSUB36:IFSF=ØTHEN9
16 GOSUB44:SQ(I)=SQ(I)+1:GOTO3
17 IFSQ(I)<5THEN9
18 SF=Ø:GOSUB32:IFSF=ØTHEN9
19 GOSUB44:SQ(I)-SQ(I)-5:GOTO3
2Ø IFSQ(I)>14THEN9
21 IFTY(I)>2THENIF(SQ(I)+5)>14THEN9
22 SF=Ø:GOSUB4Ø:IFSF=ØTHEN9
23 GOSUB44:SQ(I)=SQ(I)+5:GOTO3
24 FORJ=ØTO19:LC(J)=Ø:NEXT:FORJ=ØTO9:LC(
SQ(J))=1:IFTY(J)MOD2=ØTHENLC(SQ(J)+1)=1
25 IFTY(J)>2THENLC(SQ(J)+5)=1
26 IFTY(J)=4THENLC(SQ(J)+6)=1
27 NEXT:RETURN
28 IFLC(SQ(I)-1)=1THENRETURN
29 IFTY(I)<=2THENSF=1:RETURN
3Ø IFLC(SQ(I)+4)=ØTHENSF=1
31 RETURN
32 IFLC(SQ(I)-5)=1THENRETURN
33 IFTY(I)MOD2=1THENSF=1:RETURN
34 IFLC(SQ(I)-4)=ØTHENSF=1
35 RETURN
```

continued

PUBLIC SERVICE ANNOUNCEMENT: Because of the public outcry, we at Portable 100 *have decided, as a public service, to help prevent assault on batteries. If you (take note, Mr. Almlie) have a loved one who has engaged in NiCd abuse, overusing and undercharging while trying to solve* PUZZLE.BA (Dec. '88, p. 29), *you may direct him or her to this page for the solution. Be aware, though, that like all public welfare programs ours has a catch. The program below shows you the solution to the puzzle, but you have to remember it. Good luck.*

Studying the schematic diagram of the Tandy 200 power supply and noting the circuits involved with the two jumpers, I decided that one jumper was for the trickle charge of the batteries. The second jumper altered the bias of the LED sensor circuit.

At that time, my Tandy 200 was still under warranty—installing the jumpers did not void it—so I brought the unit to a local Tandy repair center. I was using the *Tandy 200 Service Manual* for reference, and I discussed the problem with the service manager.

But to maintain the warranty, I could not make any further changes. After a brief (but technical) discussion with a local service manager, I left the unit with him for two hours. We generally agreed on what had to be done, but he needed the official "OK" from Tandy to proceed with further modifications.

The fix was simple and pretty much like what we had discussed. It involves changing one resistor (R93) on the main PC board from 94.4K ohms to 150K ohms. There was no charge for the service. The *LOW BATTERY* LED now gives me about a twenty-minute warning, I get about five hours of use without the AC adapter, and my Tandy 200 has gone three and a half years on one set of NiCd's.

As a postscript to this story, the jumper at J301 is in series with the 94.4K ohm resistor. So rather than installing the jumper and then changing R93, you could instead simply install a 55.6K ohm resistor at J301 in place of a wire jumper, and leave R93 intact.

UNWANTED COLD STARTS

Have you ever found your Tandy 200 "locked up" in bank #2 or bank #3, with no way to recover the system? Pressing *SHIFT/BREAK/RESET*, and even *CTRL/BREAK/RESET*, will not unfreeze the system. As a last resort you turn power off and attempt to cold start the frozen bank by pressing *CTRL/BREAK* while turning power back on. This time it works. But Murphy's Law prevails, and you find that you've just wiped out bank #1.

As if that wasn't painful enough, when you switch into the previously-frozen bank, your Tandy 200 locks up again. So you perform another *POWER-ON/CTRL/BREAK*, which cold starts bank #1 again, and you're still faced with the problem of unfreezing the locked-up bank. The solution is this: To cold start the corrupted bank, you must start in the *previous* bank. While holding down *CTRL/BREAK*, press *TAB* or *F1* to switch into the frozen bank, and that bank will cold start.

Sounds strange, doesn't it? I've seen this problem time and again while experimenting with the RAM hooks or incorrectly modifying a running *.CO* program. Don't worry, this is not a random occurrence. Although you may seldom learn the reason for the lockup, you can be sure you caused it. Perhaps you were using a corrupt program, for example. So after the cold start, you reload a perfect version of that program into RAM, and everything runs fine.

If "running fine" after a cold start isn't good enough—and it wasn't for me—then an explanation of the power-on sequence may shed some light on the matter. The cause of the lockup certainly varies from user to user, but the mysterious bank #1 cold start is no mystery at all.

Regardless of which bank is currently active, whenever the Tandy 200 is powered on, it jumps into RAM bank #1. Location EEF5H (hexadecimal) in bank #1 stores the number of the currently active bank. This location in bank #1 is updated whenever you switch banks, so at power-on, the Tandy 200 knows exactly which bank you were using. Unfortunately, at power-on, the Tandy 200 looks for keyboard input, and if *CTRL/BREAK* is recognized, it initiates an immediate cold start. That is the reason for the unexpected bank #1 cold start when attempting to recover from a lockup in another bank. This cold start also resets the byte at EEF5H so you remain in bank #1, leaving the problem in the other bank unresolved.

```
36 IFLC(SQ(I)+1+(TY(I)+1)MOD2)=1THENRETU
RN
37 IFTY(I)<3THENSF=1:RETURN
38 IFLC(SQ(I)+TY(I)+3)=ØTHENSF=1
39 RETURN
4Ø IFLC(SQ(I)+((TY(I)<3)+2)*5)=1THENRETU
RN
41 IFTY(I)MOD2=1THENSF=1:RETURN
42 IFLC(SQ(I)+1+((TY(I)<3)+2)*5)=ØTHENSF
=1
43 RETURN
44 C=Ø
45 X=SQ(I)MOD5:Y=INT(SQ(I)/5):LINE(X*48,
Y*32)-((X*48+XX(TY(I))),Y*32+YY(TY(I))),
C,B
46 PRINT@41+Y*16Ø+X*8,RV$CHR$(I+48)CHR$(
27)"q";:IFC=ØTHENPRINTCHR$(8)+"
47 C=1:RETURN:DATA 5,4,15,2,17,2,19,1,14
,1:DATA 9,1,4,1,7,3,2,2,Ø,2:DATA 46,3Ø,9
4,3Ø,46,62,94,62
48 DATA 52,29,51,3Ø,5Ø,28,55,31,56,31,54
,29,53,3Ø,56,28,55,3Ø,3Ø,52,29,31,51,29,
29,56,31,54,31,28,55,28
49 DATA 51,3Ø,3Ø,52,3Ø,3Ø,56,29,5Ø,29,54
,31,31,53,31,31,55,28,52,28,3Ø,56,3Ø,5Ø,
3Ø,49,28,28,48,31,57,31,51,29,29,52,29
5Ø DATA 29,56,3Ø,57,28,28,48,3Ø,49,29,29
,54,29,29,53,31,29,55,31,31,57,28,56,28,
52,28,31,51,28,28,48,3Ø,49,3Ø,54
51 DATA 29,29,53,29,29,5Ø,31,49,28,28,54
,3Ø,53,3Ø,5Ø,29,29,49,31,52,31,28,51,31,
31,48,28,54,3Ø,3Ø,53,29,3Ø,51
52 DATA 29,29,48,31,56,29,29,57,3Ø,52,3Ø
,28,48,28,51,28,3Ø,5Ø,3Ø,49,29,29,48,31,
51,28,28,53,28,54,31,56,29,57,29,52
53 DATA 3Ø,51,28,48,3Ø,49,28,28,5Ø,31,54
,31,53,31,56,31,57,29,29,48,3Ø,53,28,28,
54,28,28,5Ø,3Ø,49,29,29,53,31,29,55,29
54 DATA 51,31,31,52,31,31,48,28,54,3Ø,3Ø
,53,3Ø,3Ø,55,29,52,29,31,48,31
55 IFINKEY$=""THEN55ELSEMENU
```

End of listing.

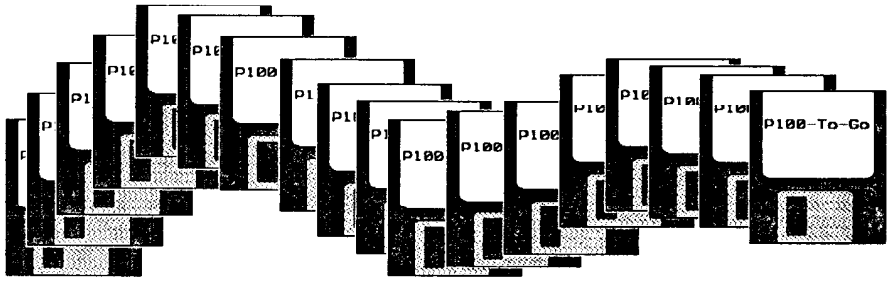
On the other hand, if you don't hold down *CTRL/BREAK*, the power-on sequence does a warm start in bank #1, which looks at EEF5H and then warm starts the active bank. The warm start sequence also looks for *CTRL/BREAK*, but now you are in the active bank. If you hold down *CTRL/BREAK* during a warm start, a cold start is initiated in the active bank. (*CTRL/BREAK* only has this effect during cold or warm starts, so there is no problem should you inadvertently press this key combination at any other time.)

Pressing *F1* or *TAB* at the menu will switch the computer into the next bank and initiate a warm start, so it is no surprise that pressing *CTRL/BREAK* while switching into any bank will cold start that bank.

Getting back to the original problem, how can you avoid having to cold start *two* banks, when only one bank is corrupt? That's a tough one! My personal solution is to use banks #2 and #3 for storage of programs and data, and use bank #1 to actually *RUN* the programs. Now if I get a lockup and it becomes necessary to cold start, I lose only what is in bank #1.

FINGERS TIRED?

Accomplish more... in less time.
Get the P100-To-Go monthly program disk.

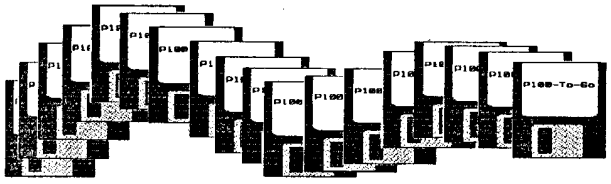


Make your life easier! Save hours of time from typing in lengthy programs. Never worry again about typing errors. Have more fun with *Portable 100*'s new disk-of-the-month service. Every program from the magazine is included on each month's disk. No typing. No trouble.

We make it easy and inexpensive. You deserve value. At *Portable 100* we believe in value. That's why we're not only making each month's programs available on 100K Tandy Portable Disk Drive disks, we're also making sure any extra room on each disk is packed with interesting and fun bonus programs. All files are in *.DO* format. Ready to load. Ready to go.

Not just for the Tandy 100. The *P100-To-Go* disks will work with any computer that uses the Tandy Portable Disk Drive, the Brother FB100, and the Purple Computing Portable Disk Drive. This means your NEC-8201/8300, Tandy 100/102, Tandy 200, Olivetti M10, and Kyotronic KC-85 can all load these programs into memory (once loaded, changes for specific computer models may be required, refer to corresponding program article for information about compatibility with particular computers).

Subscribe and save! You can order *P100-To-Go* disks individually at only $9.95, but for maximum savings sign up for our **Six-pack Special**, one disk each month for six months, six disks, for only $47.70. Save over **30%**, a savings of **$2.00** per disk.



❑ Yes! Save me hours of typing time. Sign me up for your **Six-pack Special** (6 monthly disks of the Portable 100 programs, plus the special bonus programs included with every disk) for only $47.70.

❑ ***P100-To-Go*** 1988 disk set (Jun.-Dec. 1988) $47.70.
❑ January '89 single issue ***P100-To-Go*** disk, $9.95.
❑ February '89 single issue ***P100-To-Go*** disk, $9.95.
❑ March '89 single issue ***P100-To-Go*** disk, $9.95.

To place your order immediately, call:
603-924-9455

Prices include postage and handling. Please allow 4-6 weeks for delivery. Foreign airmail add $2 per disk. All foreign, US Funds drawn on US banks only.

❑ Payment enclosed (check or money order)
❑ Mastercard ❑ VISA ❑ Amex

CARD NUMBER _____ EXP. DATE _____

SIGNATURE _____

NAME _____

STREET ADDRESS _____

CITY _____ STATE _____ ZIP _____

Mail to: Portable 100, P100-To-Go,
P.O. Box 428, Peterborough, NH 03458-0428

Chips-To-Go: *Chipmunk owners—Club 100 will convert your* P100-To-Go disk *to Chipmunk format. Send your original* P100-To-Go *disk, with $5.00 plus First Class return postage, to Club 100,* Chips-To-Go, *P.O. Box 23438, Pleasant Hill, CA 94523.*

CUSTOM 200

IPL PROGRAMS

A neat feature of the Tandy 200 is its ability to set IPL (Initial Program Loader) to run a program automatically at power-on time. The command *IPL "filename"* designates the file to execute (e.g., *IPL "SETUP.BA"*). You can either type the command at *BASIC*'s *Ok* prompt or include it within a *BASIC* program. Any type of RAM file will execute, but you must remember to include the extension (*.BA*, *.DO*, or *.CO*) in the command. To clear the IPL, so the program will no longer execute on power-up, simply type (or program) the *IPL* command with a null filename (i.e., *IPL""*).

Execution of the IPL takes place during a warm start, so if you bank switch the Tandy 200, you execute the IPL in the new bank (assuming it had been set). You can also warm start a RAM bank from *BASIC* with *CALL 0*. (Be careful not to press *CTRL/BREAK* or you will cold start that bank and wipe out all files.)

I wanted to discuss IPL because it's part of the warm start procedure. The significant, but once again undocumented, aspect of IPL is that before the Tandy 200 executes IPL, it scans the keyboard for the *SHIFT* key. If you hold *SHIFT* down, then the IPL is bypassed. This is useful when switching into a RAM bank that has IPL set but unwanted at the moment.

If you have found a way to use the IPL feature, as in program chaining (discussed in detail in *Portable 100*, June '88), then knowing how to bypass this feature might be of equal importance.

Incidentally, *IPL CHR$(13) + CHR$(0)* makes the computer run the unsaved *BASIC* program (if one exists) as the IPL program. If no unsaved *BASIC* program exists, the computer enters *BASIC* and goes straight to the *Ok* prompt, never displaying the Microsoft logo. I haven't found a practical application for this; it's just an interesting discovery.

USING *MSPLAN* WHEN *HIMEM <> MAXRAM*

Have you ever tried to enter *MSPLAN*, only to get the *MP work area in use* message and a quick return to the menu? You attempt to solve this mystery by referencing the *Tandy 200 Owner's Manual*, but you cannot find any information relating to this problem. The Tandy 200 also comes with the *Tandy 200 Multiplan Manual*, so you may have a momentary feeling of relief, until you discover that the *Multiplan Manual* does not discuss this problem either!

To enter *MSPLAN* successfully, you must set *HIMEM* equal to *MAXRAM*. You do this by entering *BASIC* and typing *CLEAR 256, MAXRAM*. *MAXRAM* is normally set to 61104, although a few software packages alter *MAXRAM*. For this discussion, let's assume that *MAXRAM* defaults to 61104.

Suppose you attempt to enter *MSPLAN* using the above *CLEAR* statement to set *HIMEM* equal to *MAXRAM* and you get the *Insufficient free space* message. This occurs whenever you have fewer than 4,213 bytes of free RAM.

Not only do you need the minimum 4,213 bytes free to enter *MSPLAN*, but you need additional free bytes for the *MSPLAN* file you are about to create. If the file already exists and is on the menu, then at least 4,213 bytes must be free in order to enter this file. These RAM requirements are not mentioned in the Tandy manuals.

When *MSPLAN* is being used, it requires an area of RAM for miscellaneous calculations. This "scratchpad" area is from address 59638 to 61103, so if you have set *HIMEM* lower than this area, say, to run a *.CO* program, *MSPLAN* respects this protected area and will not run until you unprotect it and reset *HIMEM* to equal *MAXRAM*.

However, you can make *MSPLAN* work with *HIMEM* set lower than *MAXRAM* if you meet two conditions: First, *HIMEM* must be *less* than 59638. And second, memory locations 61102 and 61103 must contain the ASCII values for *M* and *P*.

This information will be of significant value to machine language programmers. If you've written a program that requires *HIMEM* to be set below 59638, you can allow the user to access both *MSPLAN* and your program without having to alter *HIMEM* by simply storing the ASCII *MP* at 61102 and 61103.

This is a safe technique if your machine language program resides on the menu and is loaded into the *HIMEM* area with each use. If your program is loaded into *HIMEM* and then accessed by other programs, then you would *not* insert *MP* at 61102 and 61103 (to ensure that *MSPLAN* will not overwrite your machine language code).

And if you haven't guessed by now, Tandy has not documented any of this information, either. But I thought you would like to know.

by Paul Globman

Paul can be reached by modem on CompuServe (72227,1661) and GEnie (P.GLOBMAN). Or by mail at 9406 N.W. 48th St., Sunrise, FL 33351 (please enclose SASE if you're requesting a reply).

The Portable 100 Classifieds

SERVICES

MODEL 600. Programs/Newsletter. Send S.A.S.E. for information. Easy Ware Products; P.O. Box 218778; Houston, TX 77218-8778

SOFTWARE

TRANSFER.COM MS-DOS/ Tandy Laptop $9.95 plus $1.00 shipping and handling (Min. $18 for MC or VISA orders; Calif. residents add 6.5% tax).

Easily transfers .DO files both ways between MS-DOS computers and your Model 100, 102, or 200 laptop. MS-DOS formatted 5.25-inch disk contains the transfer program, complete documentation and several unique files, including laptop programs not found anywhere. Write to: Hanson-McBride Services, 984 Hawthorne Drive, Walnut Creek, CA 94596. Order phone number (415)939-5285. 3/89

BY Teachers for Teachers: "Grade Management System" Personalized... Stay organized. You set up to: 12 grade values-9classes-50 students. 3.5-inch disk for PDD. Tandy 100/102, 200. We also customize 5 *BASIC* programs. $49.00+$2.00 (S&H). Software by Steffens', 1731 William Ct., Little Chute, WI 54140.

ENCRYPTION SOFTWARE. Highest security. Stronger than DES. Tandy 100 and compatibles. D. Vogel, 808 Post, #915, San Francisco 94109.

HARDWARE

TANDY 600 MEMORY-$129. We make the 96K memory expansion boards that fit inside your Tandy 600 Portable. You can expand your 600 yourself to 224K with 2 boards. Complete installation instructions, 90 day warranty, and 30 day money back guarantee. We also make memory for the Model 100 & Tandy 200. Visa, M/C. Purple Computing. Call (503)479-8087. 3/89

"DATAMITE" analog to digital converter for Tandy laptops. Compact, single channel unit connects to cassette jack. High resolution, 0.04% accuracy. $45.00 Companion chart-recorder software. $20.00. Jones Service & Design; 1842 S. Nugent Rd; Lummi I., WA 98262. (206) 758-7258. Visa/Mstrcd. 3/89

Tandy Disk Video Interface $200. Zenith Amber B/W monitor 12" $100. (203) 379-7864 3/89

Model 102 with SuperROM, Tandy Drive 1, 3-32K RAMpacks, printer and modem cables, cassette and cable. $600. Call Doug Nelson at (w) (617) 577-1717 or (h) (617) 876-3117 3/89

Model 200 w/disk drive (PDD-2), Extra 29K bank, manuals-- $600 or best offer. (916) 722-9786 3/89

M100 (32K), PDD-1, CCR-82 cassette, power supplies, modem cable, TS-DOS, 10 programs, P100/PICO collection. All like new. Original boxes. $650 or best offer. (407) 479-4995. 3/89

WANTED

WANTED: CGP-115 Plotter for Model 100. Will Pay $200 in good condition. Call Alice Ann at (407) 425-1001. 6/89

HOW TO PLACE A CLASSIFIED AD

Categorize your advertisement (Hardware, Software, Services, Wanted, Etc.) and carefully type your message. We are not responsible for correctly interpreting handwritten advertisements. Phone numbers, street numbers, dimensions, and any abbreviations count as one word. Logos, company or product, are not allowed, neither are display advertisements. Business rates are $1.60 per word; non-business rates (individuals advertising) are $1.30 per word. Add up the cost and send the advertisement copy with a check, money order, Visa, or Mastercard number to: Portable 100 Classifieds, 145 Grove St. Ext., P.O. Box 428, Peterborough, NH 03458-0428, c/o Linda Tiernan. Make checks payable to Portable 100 Classifieds. Include your complete return address and phone number (phone number is printed only if it is included in the advertisement itself). Materials due the 1st of the month, two months prior to the magazine cover date (example: materials for the February issue must be received by December 1st.) Advertisements received after the deadline will appear in the next scheduled issue. Payment must accompany order. No refunds for advertisements that miss deadlines, regardless of reason. We reserve the right to change advertisement categories, and to reject, edit, or cancel any advertisement we deem improper. There are no agency discounts for classifieds. **For faster service call 603-924-7949.**

_____ _____ _____ _____ _____

_____ _____ _____ _____ _____

_____ _____ _____ _____ _____

_____ _____ _____ _____ _____

_____ _____ _____ _____ _____

Ad Category _____ No.of words x (1.60)(1.30) _____ #Issues to run _____

Name _____

Address _____

City _____ State ____ Zip _____

Total Enclosed x (Ad Cost x Issues) _____ Phone Number _____

Visa/Mastercard Number _____ Exp. Date _____

Signature _____

ADVERTISER'S INDEX

Free Information

For free information on products advertised in this issue of Portable 100, locate the Reader Service number corresponding to the advertisement that interests you. Circle the number on the Reader Service Card at page 25 (or on the wrapper protecting the magazine if you are a subscriber) and drop it into the mail. The literature you've requested will be forwarded to you without any obligation. Please allow 3-5 weeks for delivery.

ADVERTISERS

NEW PRODUCTS

Don't forget to say you saw it in Portable 100!

SUPER ROM

Lucid Spreadsheet **Write ROM** **Database** **Outliner**

NOW YOU CAN REALLY HAVE IT ALL!

All on one ROM. Truly the finest four programs available for the Model 100 — guaranteed. Try it for 30 days. If you aren't blown away by the excellence return it for a full refund.

$199.95

The four best programs for the Model 100 all on one ROM. 32K of power without using any RAM for program storage. This is the PCSG Snap-In ROM that just presses easily into the little ROM socket in the compartment on the back. You access the four right from the main menu like built-ins.

Write ROM — the definitive word processor for the Model 100. Function key formatting or dot commands. Search and replace. Library feature — inserts words, phrases or whole documents into text from just a code. MAP lets you see a picture of your document. In all there are 60 features and functions. No one can claim faster operation. FORM lets you create interactive forms with on-screen prompts that you can answer from the keyboard. Nothing else for the Model 100 compares with the features of Write ROM. Exactly the same as the Write ROM sold as a single program. Infoworld says it "makes the Model 100 a viable writing unit ... surpassed our highest expectations for quality and clarity."

Lucid Spreadsheet: This is the one PICO magazine says "blows Multiplan right out of the socket" and Infoworld performance rated as "excellent" and said "makes the Model 100 compute." Gives you features you cannot get with Lotus 123. Lets you build spreadsheets in your Model 100 that would consume 140-150K on a desktop. Program generating capability with no programming knowledge required. Variable column widths. Includes find and sort with function key control. It's fast, recalculates like lightning. No feature has been taken from the original, only new ones added.

Database: This is a relational data base like no other. You can do everything from mailing lists to invoices. No complicated pseudo-coding, you create input screens as simply as typing into TEXT. You are not limited by size; you can have as large an input screen as you wish. Prints out reports or forms, getting information from as many files as you like. Complete math between fields. Total interface with Lucid worksheets.

Outliner: Does everything that Thinktank does on a PC but a whole lot better. Includes a Sort for your headlines. Lets you have headlines of up to 240 characters. Has cloning, hoisting and sideways scroll up to 250 characters. Like Lucid, this one sets a new standard for outliners. This is the way to plan and organize your projects.

Present Lucid and Write ROM owners can upgrade for $150. If you have both it's $125.

As usual PCSG sells the Super ROM on a thirty day guarantee. If for any reason you are not satisfied, simply return it for a full refund.

We are excited about this product. Super ROM gives the Model 100 the true power of a desktop. No other multi-program ROM has software that compares. But don't take our word for it. We invite you to make that comparison yourself. Priced at $199.95 on Snap-In ROM.

Got stuck with somebody else's multi-ROM? We'll upgrade it for $150.

(214) 404-4030

PORTABLE COMPUTER SUPPORT GROUP

4540 Beltway Drive, Dallas, TX 75244 • 1(214) 404-4030 • 1-800-544-4699

MC, Visa, American Express, Check, or C.O.D.

Circle 66 on reader service card.

The sky's the limit... ...with Portable 100 magazine.

On the road . . . at home . . . in the office. . . in the air. . . the variety of tasks you can perform with your Tandy portable is expanding constantly.

To keep fully informed and up-to-date about current trends, new products, and new uses for your laptop, you need **Portable 100 magazine.**

From sophisticated input/output (I/O) calls and their applications, to simple disk drives, **Portable 100** magazine covers it all.

Portable 100 gives you features, news, columns, and reviews that are thorough and timely. And they are written in a fast-paced, easy-to-read style, by leading experts in the computer field.

In upcoming issues you will discover. . .

- how new peripherals can make you more productive.
- more efficient ways to communicate with your desktop.
- where to buy low-cost public-domain software.
- how your business can be more profitable with your portable at your fingertips.
- where to find the best bulletin boards and information services.
- and much more!

Don't miss even a single valuable monthly issue. Fill out the coupon below, or to charge it to your credit card, call

1-603-924-9455

Portable 100, the magazine for Tandy 100, 102, 200, 600 and 1400LT users

SAVE 42% OFF THE COVER PRICE!

☐ **YES!** I want to be more productive with my Tandy portable... and save 42% off the newsstand price. Send me a year's subscription (monthly except for a combined July/August issue) of Portable 100

☐ Payment enclosed ☐ M.C. ☐ Visa ☐ Bill me
Make check or money order payable to Portable 100

Name _____

Street Address _____

City _____ State _____ Zip _____

I have a Tandy ☐ 100 ☐ 102 ☐ 200 ☐ 600 ☐ 1400LT
or an ☐ NEC8201/8300 ☐ Olivetti M10 ☐ KC-85

Canada and Mexico, US$29.97, Foreign surface, US$44.97, Foreign Air Mail add US$50. All foreign, except Canada, US Funds drawn on US bank. P892

MONEY BACK GUARANTEE:
If you are not completely satisfied with Portable 100, you may cancel your subscription and receive a refund on the remaining unmailed issues. Please allow 6-8 weeks for delivery of your first issue.

Mail to: ***Portable 100***, Circulation Department, PO Box 428, Peterborough, NH 03458--0428