

\$2.50

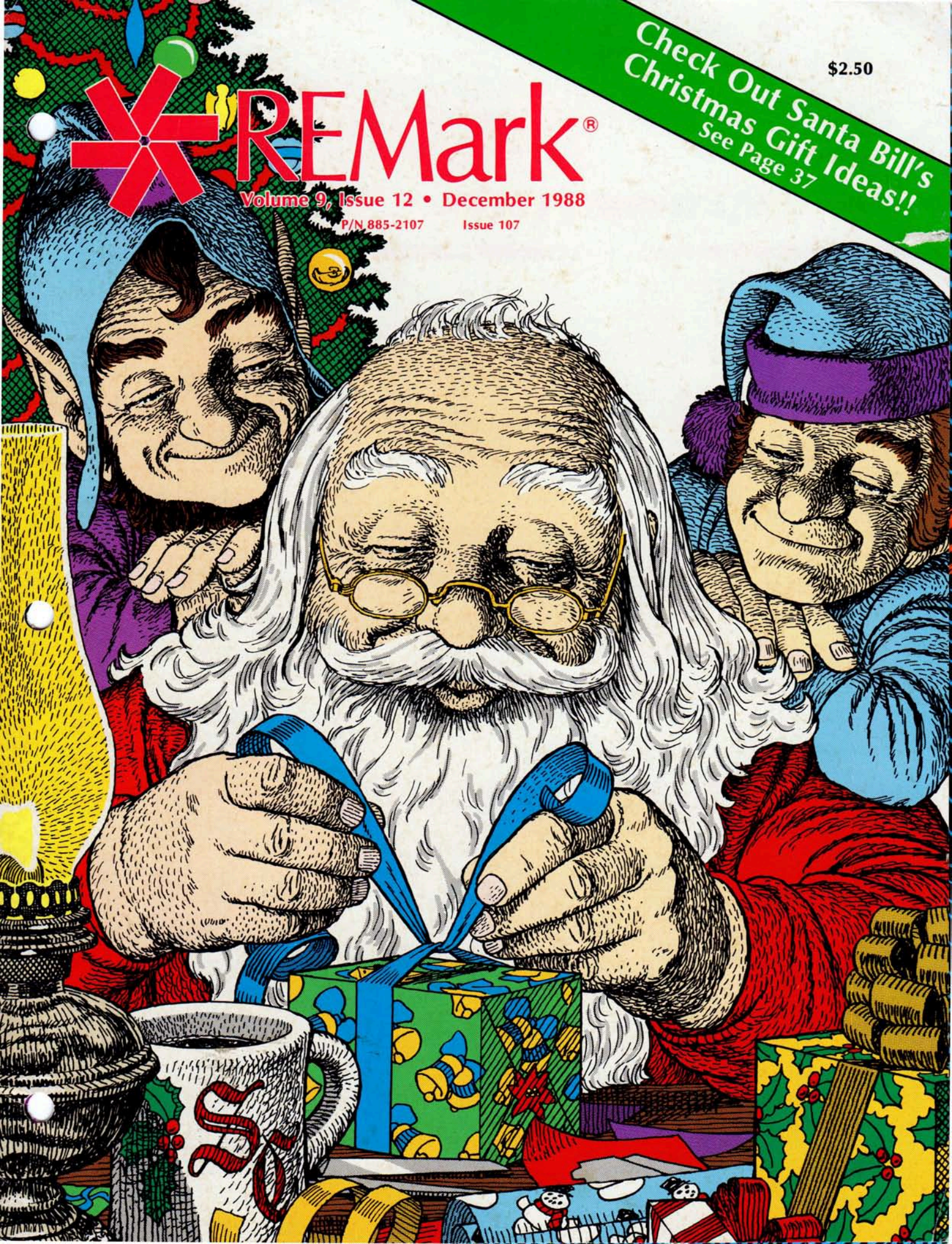
Check Out Santa Bill's
Christmas Gift Ideas!!
See Page 37

REMark®

Volume 9, Issue 12 • December 1988

P/N 885-2107

Issue 107





"I've Got Your Number Now . . .!"

"Now you have no reason for us to be strangers. I know each and every one of you personally! My human gave me the entire HUG Database to study and now there's no need for **any** HUG member to register. Just have your ID Number handy when you call . . . I do! I'll bet you didn't know that my Bargain Centre has discounts on HUG products, as well as other computer **and** non-computer related items. And, did you know that I can renew your HUG membership . . . at a discount?!! I can give you over 35 megabytes of free software. All you have to do is take it! Now no-one will **ever** have to know about the two of us. Just set your modem to 300, 1200, or 2400 baud, 8-data bits, no-parity, and call me anytime your computer needs someone to talk to. I can be contacted at (616) 982-3956. Although no longer required, my human can be reached at (616) 982-3837.

MOC

On The Cover: We, here at the Heath/Zenith Users' Group, wish you and yours a very happy holiday season.



Managing Editor Jim Buszkiewicz
(616) 982-3837

Software Engineer Pat Swayne
(616) 982-3463

Production Coordinator Lori Lerch
(616) 982-3794

Secretary Margaret Bacon
(616) 982-3463

HUG Bulletin Board (616) 982-3956

HUG Parts Ordering (616) 982-3463

Contributing Editor William M. Adney

Printer Imperial Printing
St. Joseph, MI

	U.S. Domestic	APO/FPO & All Others
Initial	\$22.95	\$37.95*
Renewal	\$19.95	\$32.95*

* U.S. Funds

Limited back issues are available at \$2.50, plus 10% shipping and handling — minimum \$1.00 charge. Check HUG Product List for availability of bound volumes of past issues. Requests for magazines mailed to foreign countries should specify mailing method and appropriate added cost.

Send Payment to: Heath/Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49022
(616) 982-3838

Although it is a policy to check material placed in REMark for accuracy, HUG offers no warranty, either expressed or implied, and is not responsible for any losses due to the use of any material in this magazine.

Articles submitted by users and published in REMark, which describe hardware modifications, are not supported by Heath/Zenith Computers & Electronics Centers or Heath Technical Consultation.

HUG is provided as a service to its members for the purpose of fostering the exchange of ideas to enhance their usage of Heath equipment. As such, little or no evaluation of the programs or products advertised in REMark, the Software Catalog, or other HUG publications is performed by Heath Company, in general and HUG, in particular. The prospective user is hereby put on notice that the programs may contain faults, the consequence of which Heath Company, in general and HUG, in particular cannot be held responsible. The prospective user is, by virtue of obtaining and using these programs, assuming full risk for all consequences.

REMark is a registered trademark of the Heath/Zenith Users' Group, St. Joseph, Michigan.

Copyright © 1988, Heath/Zenith Users' Group

Features

The Ultimate Cheap Desktop Publisher <i>Pat Swayne</i>	17
Know Your abCEDs <i>Ed Schindler</i>	22
A Shot in the Arm for the H/Z-100 <i>Richard L. Mueller</i>	27
Easy Extra is Easy WordStar <i>Ralph Seiler</i>	30
Sabotage — Viruses and Trojan Horses <i>Frank Starr</i>	43
A Really Fast Word Processor for the Zenith 171 <i>Daveed Dov Shachar</i>	45

Columns

Mainstream Computing <i>Joseph Katz</i>	7
On the Leading Edge <i>William M. Adney</i>	37

Series

Bootable EPROM Disk for the H-100 — Part Three <i>Richard F. Hassard</i>	9
Getting Started With . . . Turbo C <i>Alan R. Neibauer</i>	11
How to Get the Most From a Zenith Laptop Computer — Part 6 <i>Joseph Katz</i>	25
Turbo Pascal — Part 2 <i>Matt Elwood</i>	48
ENABLE — Part 12 <i>George Elwood</i>	51
POWERING UP <i>William M. Adney</i>	57

Reader Service

Reader Service No.		Page No.
104	FBE Research Co., Inc.	56
105	First Capitol Computer	4,5
105	First Capitol Computer	6,24
***	HUG Members Only	67
***	HUG PBBS	2
107	Paul F. Herman	21
137	Jay Gold Software	63
111	KEA Systems	29,33
136	Lindley Systems	29
114	Micronics Technology	63
117	Payload Computer Svcs.	36
121	Scottie Systems, Inc.	63

Resources

Buggin' HUG	5
HUG Price List	34
H/Z Related Products	64
Classified Ads	66

Attention Advertisers!!

Advertising for REMark Magazine will now be handled through:

Rupley's Advertising Service

Dept. REM, 240 Ward Avenue
P.O. Box 348
St. Joseph, MI 49085
(616) 983-4550



*We Couldn't Find a Tower that
was Designed to Our Specs...
So We Built One Ourselves!*

The *NEW* First Capitol 386 Tower

High-Speed 20MHz Operation ...and Lower Cost!

We're not a nationally recognized brand - YET - but we do have one of the *fastest* 80386-based computers on the market! Our CL-386TW line of computers incorporates a 20MHz (no waits) 80386 processor chip fully compatible with Big Blue's finest. *That's 25% faster than than 16MHz units!*

Price-wise our CL-386TW units begin selling at less than \$3,000 configured with video, monitor, drive, and 1Meg of RAM! *Compare with prices of \$4,500 and up for comparable units!*

Designed as a Tower for Space Savings and Convenience

It seems like the faster the system, the more desk space the computer will absorb. Not so with our CL-386TW units! The unit is floor-mounted in a ruggedized case. Only the keyboard and monitor need be located on your desktop. Imagine the joy of having a free desk top once again!

Incredible Expansion Capability

Included on the motherboard are 2-RS232 standard serial ports, 1-Centronics parallel port, and a real-time clock/calendar with battery backup. Talk about expansion slots - - this baby has 8 slots, with 6 slots open when fully configured!

The chassis contains an over-size 200 watt power supply, plus mounting for *up to 3 floppy devices and 2 hard drives!*

...Built the Way YOU Want It!

Unlike the "Big Boys", we don't limit you to the models that we want to build. We build our models the way YOU want them! Every unit is built to your precise order, and burn-in tested to insure quality prior to shipment - all at no extra charge to you.

We'll Even Sell You a Stripped- Down Unit to Use to Upgrade Your Present System!

If you'd prefer, we'll sell it to you without drives, controllers, or

memory so that you can use it to upgrade an existing system. Not to be confused with our Z-386 upgrades.

One Year Limited Warranty

First Capitol Computer provides a full one year's limited warranty on parts and labor via module replacement. That is, if your system should prove defective at any time within one full year from shipment, we will replace the defective module with a new one free of charge (other than shipping charges).

Immediate Availability!

You don't have to wait, they're available NOW! Avoid the Rush! *Call for details today!*

**First
Capitol
Computer**

#16 Algana, St. Peters, MO 63376
Order Line: 1-800-TO-BUY-IT
Other Calls: 1-314-447-8697

BUGGIN' HUG

POWERING UP

Dear HUG:

Just want to say a quick thank you for running William Adney's series, POWERING UP. Though I have been using a computer for a little while already, I found that most of the articles in HUG were somewhat beyond my comprehension. I recognized much of the information, but had no idea as to what to do with it. I was about to let my subscription lapse when Mr. Adney's series started. I decided to give it another try. While many things he discusses I am already somewhat familiar with, there are many things that are new. Even the things that are a little repetitious to me have been helpful seeing them presented in a clear, coherent manner.

Just wanted to let you know that there were some of us beginners out there and we appreciate being recognized. Thanks to you and to Mr. Adney.

P.S. I even venture into the other articles and am finding that I understand more each time.

Sincerely,
Steve Waxler
8200 Indian Hill Road
Cincinnati, OH 45243

No Computer to Use it On

Dear HUG:

I have been a Z-181 user since January of 1987 and have thoroughly enjoyed using this machine, my first computer. For the most part, I use WordPerfect but also use other well known software, including an earlier version of Autocad. There are certainly drawbacks to running this sophisticated CADD software on a monochrome 640k machine with no hard disk, but I've been able to learn quite a bit, and the CGA graphics and backlit LCD screen of the '181 allow use of most of Autocad's features.

This letter is really intended to share my experience with the purchase of a mouse for use with this computer. After reading about the various mouse brands and features in PC Magazine and Info World I unwittingly purchased a Logitech C-7 Mouse bundled with their Paint software. By all reports this mouse offered the best features and has a good price. Unfortunately, I had not been warned of one very important fact: the Z-181's serial port will not support the Mouse due to an

inadequate power supply!

I was informed of this fact upon requesting 3.5 inch software from Logitech. After haggling with the customer support and sales people, I was informed that it was my responsibility as a potential purchaser to research the product more fully than I had. There was no mention of this problem in any trade literature that I had seen, and Logitech admits that this is in no way a common problem. I feel that I am a cautious consumer and that some mention of this incompatibility should have been made in one of several articles regarding this product. The seller, of course, offers absolutely no warranty of hardware compatibility.

At this point, I have not approached Zenith with this problem; I am really not expecting very much at this point. I am open to suggestions outside of tearing down the computer. As the situation stands now, I have a mint condition Mouse and software waiting for a computer to use it with.

Sincerely,
Jon Linton
3709 7th Avenue #4
San Diego, CA 92103

TMSI

Dear HUG:

Here are some good words about Lee Hart and his new venture, TMSI. In November of last year, I ordered a soft-sector controller from the now bankrupt Technical Micro Systems, Incorporated. That order eventually found its way to Lee Hart and even though there was a delay of several months, he delivered the equipment. Also, his technical expertise helped me to get my whole H-89 hard- and soft-sector controller system working beautifully.

I wish Mr. Hart well with his new venture and hope that those of us with old computers, like the H-89, will be able to enjoy his support

well into the future. His address and phone number are: Lee Hart, 28612 Middle Crossing Road, Dowagiac, MI 49047, (616) 782-3980.

Sincerely,
Richard A. Schmidtke, Ph.D.
372 Fairway North
Tequesta, FL 33469

Calculate Moving Averages

Dear HUG:

Mr. Dick Stanley's original functions to calculate moving averages were OK (REMark, Vol. 7, Issue 3, March 1986, Page 21). However, the follow-up article (REMark, Vol. 10, Issue 7, July 1988, Page 61) must have been typeset by gremlins. Neither of the functions in Figure 1 and Figure 2 will fill an array.

The functions in Figure 1 will calculate the moving average for the first LENGTH entries of an array of length 60 or less. The array itself must be filled by some other means prior to invoking AVGM.

The function in Figure 2 doesn't deserve to calculate anything because there is no way to pass values of ARRAY into the



Monster Memory

From RAM add-ons to board products to SIMM's for the new Z-248/12 computers, First Capitol has it ALL, and our prices are *unforgettable*! Call us for assistance in selecting the most economical means of upgrading your memory, anything from chips to monster 11 megabyte conversions!

First Capitol Computer is a division of Software Wizardry, Inc., the pioneer of the RamPal series upgrades in use in over 40,000 Heath and Zenith systems around the country. Let us show you why we're one of *Inc. Magazine's 500 Fastest Growing Companies* (Dec. 1987 issue)!

 **First
Capitol
Computer**

#16 Algona Drive
St. Peters, MO 63376
Orders: 1-800-TO-BUY-IT
Other Calls: 1-314-447-8697

Circle #177 on Reader Service Card

function. In some compilers or operating system, undefined variables are preset to zero. In this case, the function of Figure 2 would return an AVGM equal to zero divided by LENGTH. In others, it would just bomb out.

To fill an array, the array element must appear on the left side of a replacement statement or must be the argument of a READ statement.

```

SUBROUTINE FILAR (ARRAY,LENGTH)
DIMENSION ARRAY (LENGTH)
DO 3000 I=1,LENGTH
WRITE(*,*) ' Give the value of array element ',I
3000 READ(*,*) ARRAY(I)
RETURN
END

```

or, if you must use a function

```

...
DO 4000 I = 1, LENGTH
4000 ARRAY(I) = FILAR(I)
...
...
FUNCTION FILAR(I)
WRITE(*,*) ' Give the value of array element ',I
READ(*,*) X
FILAR = X
RETURN
END
...

```

The program should, of course, be revised to have the statistical data read from a file. Users soon get tired of entering such data every time they rerun a program.

Sincerely yours,

George C. Driscoll
1117 Marble Circle
Bethlehem, PA 18017

Bothered and Bewildered

Dear HUG:

When the H-88 was in vogue, I built one, and then I built the '89. Some time later when the H-100 series was on the horizon, I recall that Heath promised "to continue support for the H-88/89". Well, the H-100 series came along and eventually support for the '88/'89 was forgotten, completely banished forever.

Wishing to progress with the times, I subsequently succumbed to the H-120. Time presses on . . . "Heath? I want to upgrade the speed on the H-120 . . . That much, huh?" . . . Well, so much for that. "Heath? I want to upgrade my memory on the H-120 to 756K . . . What? . . ." Well . . . "Heath? I want to install a PC compatible

board . . . yeah, the H-120 . . . Ye Gods! For that I'll go buy a Bentley!" And just for the fun of it . . . "Heath? . . . I sure would like to acquire the computer Voice Kit you offer . . . H-120 . . . can't use it on the H-120??" And meanwhile, Heath support for the H-100 series is history.

I think you get the idea.

When I bought the H-120 kit, it cost something around \$2700 including 192K memory, and a second disk drive. Considering the cost of the upgrades in the H-120 a person can buy any number of top quality computers, complete, with 1 Meg memory, 10 Meg plus disk drives and speed for less than it costs to upgrade the H-120.

And cost of upgrades on these items are so much less than Heath products.

One other gripe, then I'll let you go back to your lunch. I rather resent the fact that HUGCON visitors have the advantage of a great many "bargains" that can be picked up at those meetings.

Being one the of the family of 'retirees' on limited income, we are unable to afford



Ask Us to Beat Any Advertised Price!

**First
Capitol
Computer**

#16 Algana Drive
St. Peters, MO 63376
Orders: 1-800-TO-BUY-IT
Tech: (314) 447-8697

Circle #178 on Reader Service Card

the expense of the trip to the West Coast and the lodging to attend the event. The fact that the so-called "bargains" are limited to those attending seems to be a bit unfair to those of us Peons who, for some reason or another, are unable to attend. I note that other "bargains" are being regularly offered on HUGPBBS. Well, perhaps a modem and membership to HUGPBBS might be the way to go . . . if one can afford the freight on the phone bill! It would be great to be able to consider acquisition of some of those so-called "bargains".

The bottom line? Well, I greatly enjoyed building the '88, the '89, the H-120, Hero Jr., Hero 1, along with several other Heath kits. Alas!! The only real kit left with any challenge is the Hero 2000. Well, that's a bit beyond my budget. But how I would like to get my hands on one!

Yours truly,

Edward P. Markey
813 West 20th Street
Carroll, Iowa, 51401

*

Mainstream Computing

Joseph Katz
103 South Edisto Avenue
Columbia, SC 29205

There are good reasons why I'm so enthusiastic about microcomputers in general, laptop computers more specifically, and desktop publishing in particular. Maybe a glance into my workshop might help you see some of those reasons. Maybe it will even give you some ideas for approaches to your own situation.

For the last several years I've been working towards a new biography of Stephen Crane. You probably know him as author of *The Red Badge of Courage*, which you may have read in school. In fact, he wrote much more. But because he started writing as one of America's most obscure newspaper reporters when he was sixteen years old, nobody really knows how much more he wrote or what it was. Nor does anyone really know much about his life. Although there have been several Crane biographies, starting in the 1920s, they present a mythical figure who wasn't the real Stephen Crane. The facts are wrong. So, therefore, is the interpretation. What I've been doing for years is detective work to uncover the facts, bit by bit. That's the only way to get them out from under the dust of a hundred years. It's a lengthy process that takes years and years to perform. From time to time I've published books and articles intended to help straighten things out: *The Portable Stephen Crane*, an edition of his writings that is available in many bookstores, is mine and presents Stephen Crane for a serious general reader, in case you're interested in what goes on under one of the other hats I wear. Many of those publications have benefited from the use of

microcomputers. A few weeks ago, I gave preliminary shape to a book on an unknown aspect of Stephen Crane's life and works — an offshoot of the biography — and this book is entirely the product of microcomputers. I don't think it could have been done without them.

This book has its foundation in notes and transcriptions I've made during research trips to places that have unexplored or barely tapped resources: university libraries, of course, and also public libraries, historical societies, newspaper archives, and many other kinds of places, some of which I can't discuss. For a long time I carried a Kaypro transportable computer on those trips. Then I used a Zenith Z-171 for a while, and now I'm using a Z-183. Each of those trips resulted in a bunch of individual files on disks. I have scores of them. If you know your microcomputer history in any depth, you'll recognize that they're in a variety of "incompatible" formats: 5.25" Kaypro double-sided, double-density CP/M, 5.25" 360KB MS-DOS, and 3.5" 720KB MS-DOS. In fact, my variety of "incompatible" disk formats is even greater because I've used many different desktop computers since almost the beginning of the microcomputer age: a Radio Shack TRS-80 Model II with TRS-DOS and a few different versions of CP/M, for example, and a Zenith Z-100 in all its CP/M and MS-DOS modes, and of course the mainstream computers — such as the Z-248 and Z-386 — I write about here. So the sources for my book have roots that go all the way back to 8" diskettes and 5.25"

diskettes in various obsolete formats.

The reason why my labor wasn't made obsolete by all the changes in microcomputers over all these years is that I don't let myself get backed into a corner. I have a few rules that might interest you in that respect.

First, I make sure to use application programs — word processing, database, and spreadsheet, for example — that store their files in accessible file formats. By "accessible format", I mean the kind of disk file that lets me get at my data when the rush of time dictates a change of application programs.

That's why I now prefer XyWrite III Plus to some of the more popular word processing programs with better publicity: XyWrite's files are straight ASCII (American Standard Code for Information Interchange), so I can move them to other programs with little difficulty. It's also why I stay up-to-date with WordStar, even though some dudes consider it a mark of fashion to proclaim any version of Word-

Copyright © 1988, by Joseph Katz. All Rights Reserved. "Mainstream Computing" is a trademark of Joseph Katz. Please address all correspondence to me at 103 South Edisto Avenue, Columbia, SC 29205. I'll try to answer letters accompanied by a self-addressed stamped envelope, but my volume of mail is too heavy for me to promise. Unless it specifically says otherwise, I'll assume the right to publish your letter (edited as I think appropriate).

Star obsolete. It's pretty easy to convert WordStar files to straight ASCII without using WordStar itself. (There are all sorts of WordStar conversion programs available. In a pinch, I can write one myself in a few minutes. I've done it often, when I've misplaced an existing conversion program. It takes less time to write such a program than to look for one in my files.) Should WordStar disappear tomorrow, I can still benefit from the work I did yesterday and today. Early on, I used Radio Shack's Scripsit for the TRS-80 Model II. When the time came to switch word processing programs, I had such difficulty converting my Scripsit files (what I had to do was develop my own complex conversion program) that I decided to avoid such things forevermore. I really don't mind being the last kid on my block to have the latest toy. My toys last longer.

Database managers and spreadsheet programs do not store data in ASCII, so that easy escape hatch is closed. What I look for in them are other kinds of accessible file formats. I won't use a database management program that can't export its files to either dBASE DBF files or comma-delimited and carriage-return terminated files. (The latter, incidentally, is often called "mail merge" format, a reference to the format used by WordStar's old MailMerge function. Hey kids, can you say "de facto standard"?) Any database manager that wants my patronage will allow both kinds of conversion in an eyeblink: PC-File+ does, which is one reason I use it. Where spreadsheets are concerned, I won't use one that can't export to either a Lotus 1-2-3 (WKS or WK1) or a Visicalc DIF file format. Paperback Software's VP-Planner Plus uses the Lotus format and exports the DIF. Obviously, exports are only half the story: I won't consider a new database manager or spreadsheet that can't also import files in at least one of those standard formats.

As with word processing programs, so with these other major applications programs: what I look for first is easy data accessibility. I really don't care what nifty features it has that puts it ahead of the pack or breaks new ground or establishes a new generation. Those exciting metaphors make my scalp tingle and my fingers twitch with desire, but there isn't a feature around that could tempt me to consign my labors to eventual oblivion.

My second firm rule is that I want a computer with at least one serial port and the availability of at least one communications program that supports XMODEM protocol. With them I can pipe files from the last computer to it in the event there's no other way to carry along the work in which I've invested so many years. Neither of these considerations imposes much of a burden right now, because all major IBM-compatible microcomputers come standard with a serial port and all

major telecommunications programs support XMODEM protocol. Some kitchen table clones, though, still let you save a few dollars by omitting a serial port. Just a couple of years ago, when serial boards cost much more than they do today, some people actually chose to do without one. Then, I used to call them "short-sighted." Today, I call them "stuck," if they own an obsolete machine. Their data files may be obsolete, too. Pity.

My third firm rule is to get at least one "disk conversion" program for my new computer. That's another important lesson I learned from the TRS-80 Model II. There were several versions of the CP/M operating system available for it and one thing they had in common with the rest of microcomputer civilization was the ability to convert files from their own unique formats (Lifeboat's or Pickles & Trout's double-density, for example) to the soft-sectored, single-side, single-density 8" diskette format that was the passport to all other CP/M computers then. But you had to pay extra — almost \$200 — for CP/M and Radio Shack's own TRS-DOS operating system came free for nothing. That's what it turned out to be worth. By the time I figured out its value, I had boxes of TRS-DOS diskettes on which my valuable files were locked tighter than an editor's heart.

First, I had to convert my files to a format that could be imported by programs other than Radio Shack's. In the case of some programs, the job was so difficult that I gave up and left the files behind. All that work was wasted. Then, for the files I could convert and did, I had to transport them — "convert the files" — from TRS-DOS diskettes to CP/M diskettes. Unfortunately, although the TRS-80 Model II had a serial port, it had eccentric characteristics that required special programming. The only Model II TRS-DOS communications programs for it then came from Radio Shack itself, but Radio Shack's communications programs didn't use a protocol compatible with anyone else's communications programs. So my machine could talk, but not fluently, and not at length except to other TRS-80 Model II microcomputers operating from Radio Shack's proprietary TRS-DOS.

I still bless Mr. Pickles and Mr. Trout for the inclusion of a TRS2CPM disk conversion program with their version of CP/M for the TRS-80 Model II. There were a few other disk conversion programs around, but Pickles & Trout's had one significant advantage over them: it worked. TRS2CPM first taught me the importance of disk conversion programs, and subsequent experiences with microcomputers since then have reinforced my convictions about their importance. When MicroSolutions first introduced Uniform for the Kaypro, therefore, I was an early customer for that first major commercial disk

conversion program. Now I have MicroSolutions' Uniform-PC for my mainstream computers. Years from now, when the microcomputer world has erected still more digital versions of the Tower of Babel, and MicroSolutions produces the appropriate descendant of Uniform, I'll be standing somewhere near the head of its line of buyers. In the interim, I've acquired and will continue to acquire most major, and quite a few minor, programs to convert various kinds of disks — many of them to or from machines so obscure I have difficulty in believing they're not fictional.

My fourth rule is to use only first quality floppy diskettes. Sure, I like bargains, and sure, I buy as inexpensively as I can. But I differentiate between the words "inexpensive" and "cheap": I'll buy an inexpensive diskette anytime, but I won't buy a cheap diskette anytime. Obviously, any diskette I use now has to have its contents accessible years from now. I've taken some time to study the visible signs of diskette quality, and I try to keep in touch with the grapevine about it, so often I can get bargains that make me happy. When push comes to shove, however, I won't try to save a few cents a diskette at the risk of getting a batch that is unreliable. There is no reason today to play digital Russian roulette. On those rare occasions that I wind up with a box of diskettes in which one has even a single bad sector, I'll discard the entire box. Draconian, right? Remember, though, that I'm still able to read diskettes older than the length of time most people have used microcomputers. To paraphrase William Faulkner, my data does not endure: it prevails. I invest a great deal of work in creating that data. It is irreplaceable.

Which leads me to my fifth rule on the subject and the last of those that I'll discuss this month. There'll always be someone with a way to pack more contents onto a microcomputer disk than anyone else can pack onto it. It's a cute trick and a great way to save money: by apportioning the cost of a disk among many more bytes of data stored on it, you come up with a lower cost-per-kilobyte. Cost-per-kilobyte is indeed a valid way to measure storage costs. Someone at Janet's work place discovered that it's possible to format a 720KB 3.5" floppy diskette in a 1.44MB 3.5" floppy diskette drive on an IBM PS/2 there. That is a grand discovery because the savings on every single diskette are staggering. A real double-sided, high density diskette — the kind made and designated for formatting to 1.44MB capacity — sells for several times the price of an ordinary double-sided, double-density 3.5" diskette. So if you can get as much on the less-expensive double-density diskette as you can on the more-expensive high-density diskette, why not?

Continued on Page 29

Bootable EPROM Disk For the H-100 Part Three

Robert F. Hassard

3466 Tice Creek Drive, #4
Walnut Creek, CA 94595

This is the final part of a three part series on the construction and installation of a Bootable Eprom DISK board. In this part I will explain how to modify the BOOT ROM so that the BEDISK will automatically boot. My BOOT ROM is Version 2.9. Use the HAND prompt and V to find out what yours is. If it is different, we may have a problem.

If your version is different, load a disk with DEBUG and call up DEBUG. Use the S command as follows:

```
-SFC00:0 3FFF C3 FF FF FF FF
FC00:3BC9
```

If your answer was 3BC9, there is probably no problem. If it was different, then use the D Command to display FC00:3BC0. If there are not at least 12 lines of all FF, then you are out of luck. Sorry!

To prepare a modified H-100 Monitor we will use Debug on disk REVIO which was prepared in the second article of this series. Load DEBUG and then use the M command as follows:

```
-MFC00:0 3FFF 8000:0
```

Use the D command and inspect 8000:3FF0 3FFF and you must see the following:

```
8000:3FF0 EA 00 00 01 FE 5A 29 CD-3E 07
D F5 3E 80 D3 FE
```

Remember, the BEDISK Board address is 0CEH. So the following monitor revision (in contrast to an earlier version) will not disturb the support for either floppy disks or a hard disk. The Boot Drive designator (switch S101) for BEDISK is 3.

In the instructions that follow I will use conventional assembly language remarks for your edification. Please do NOT attempt to type in these remarks when using the A Command of DEBUG. It just will not work.

The first change will be to add a fourth device to the BOOT_DEVICE ta-

ble. To do so, however, requires overwriting CRC_STRNG. I have been unable to find where CRC_STRNG is used. If you have ever seen the Monitor print the word "Old " on your screen, then after this change you will see ":" instead.

Each Device Indicator in the BOOT_DEVICE table is a five byte phrase. The first two bytes are the address of the Device Routine. The address in the table is one paragraph less than the actual address of the routine because the monitor operates AT Segment FC01. The third byte is a modifier for the device select byte. The fourth byte is the primary Device Port address, and the fifth byte is the secondary Device Port address. The important thing here is that the fourth byte is used by B_DIAG to test for a valid default controller. There is no room in B_DIAG to add a test for BEDISK. So in the BEDISK BOOT_DEVICE phrase we will use the Z207 Port address and remember to correct it in the BEDISK routine. The fifth byte is "don't care" for our purposes here since whatever it is, it will be revised in the BEDISK routine as stated above.

```
-A8000:038D
8000:038D DW 3BBA ; AT Segment FC01
8000:038F DB 1 ; Device Select byte modifier
8000:0390 DB B0 ; Pseudo Device Port address
8000:0391 (RETURN)
```

Now use the D Command to check this.

```
-D8000:0380 039F
8000:0380 04 B0 B8 70 1E 04 B0 B8-42 31 04 AC A8 BA 3B 01
8000:0390 B0 4E 65 77 A0 42 6F 6F-74 00 43 6F 6C 6F 72 20
```

If that isn't what you see, something is wrong! The second change is in INIT_COMMAND where some data cells are pre-loaded. This change has to do with

the test to determine if the device is legal. This is necessary because we now have four devices (0, 1, 2, & 3). Again, use the A Command.

```
-A8000:0461
8000:0461 MOV BYTE PTR [032A],4
; Device test byte
8000:0466 (RETURN)
```

Now check it to be sure that this is what you have.

```
-D8000:0460 046F
8000:0460 03 C6 06 2A 03 04 C3 FF-1E 7B
00 B0 00 A2 32 03
```

The third change is in the C_BOOT routine for the same purpose as stated in the second change above.

```
-A8000:0722
8000:0722 CMP AL,4
; Device test byte
8000:0724 (RETURN)
```

Next use the D Command to check this.

```
-D8000:0720 072F
8000:0720 08 00 3C 04 72 06 B8 DA-03 E9
80 01 B0 00 A2 31
```

If that isn't what you see, something is wrong. The fourth change is also in the

C_BOOT routine and it is for the purpose of permitting use of the F4 key. The F4 key will manually boot BEDISK. Use the A Command as follows:

```
-A8000:0756
8000:0756 CMP AL,9B
; Permits use of F4 key
8000:0758 (RETURN)
```

Now test this using the D Command.

```
D8000:0750 075F
8000:0750 B1 97 3A C8 77 30 3C 9B-73
2C 2C 96 FE C8 A2 08
```

Now you are ready for the main routine which will be placed in an area in the Monitor EPROM which is presently not being utilized (in other words, it is now all FFs). Again, use the A Command and do NOT type the remarks.

Now save it with the W command:

```
-W8000:0
To be absolutely sure that you have it
correct, use the N and L commands to
load MMON.DOC back in. It will now re-
side at CS:0100 (remember that it is now
offset by 0100H). Use the D command to
get the following displays:
```

```
-DCS:0100 010F
xxxx:0100 06 07 05 C2 02 00 DB F5-E6
02 C2 06 00 C3 F8 FF
```

```
-DCS:40F0 40FF
xxxx:40F0 EA 00 00 01 FE 5A 29 CD-3E
```

If your displays compare exactly to the above, then your modified monitor file is most probably correct. If not, then you have made a mistake or your monitor is different than mine, which is Part No. 444-276-1.

You may check the checksum by using the N and L commands to load MMON.DOC back into debug and prepare the check program shown below. To run it, use the command G=8000:0 14. When the register display comes up, DX should be 0000. First use the R command to get a register display, then load the following:

```
-A8000:0
8000:0000 MOV AX,xxxx
;xxxx is the CS register

8000:0003 MOV DS,AX
8000:0005 MOV DX,0
8000:0008 MOV SI,0100
8000:000B MOV CX,2000
8000:000E CLD
8000:000F LODSW
8000:0010 ADC DX,AX
8000:0012 LOOP 000F
8000:0014 JMP 0014
8000:0016 (RETURN)
```

If the file MMON.DOC is correct, then all that remains to be done is to transfer it to a 27128-20 EPROM. Exchange the MMON EPROM with the ROM that is in U190. Find Switch S101 and change section 0 to 1 and 1 to 1. The first four switches should be 0 = 1, 1 = 1, 2 = 0, and 3 = 1 for autoboot. It might be wise to save the original BOOT ROM in a place where you can find it should you ever have trouble with BEDISK and want to remove it.

After reassembling your computer, turn it on and if all is well, it will boot without any disks in A: or B:. You will still be able to boot drive A: by holding down DELETE when you turn the computer on. You will get the HAND prompt. Type B, then F1, and then RETURN and if there is a disk in Drive A:, it will boot. Use F3 to boot your Hard Disk.

There is one final phase to this project that remains to be completed. I hope to be able to find someone who still makes plated through S-100 boards for hobbyists. If the price is right, I'll design a circuit layout and have one made. *

```
-A8000:3BCA
8000:3BCA MOV BYTE PTR [0009],CE ; Revise Board address
8000:3BCF MOV AL,0 ; Set to Sector 0
8000:3BD1 MOV BL,0 ; 0 offset
8000:3BD3 CALL 3248 ; Output reg AL to 0CEH
8000:3BD6 CALL 3243 ; Input to AL from 0CEH
8000:3BD9 CMP AL,EB ; Test first byte
8000:3BDB JNE 3BE4 ; If not 0EBH, is an error
8000:3BDD CALL 3243 ; Input next byte
8000:3BE0 CMP AL,1F ; Test second byte
8000:3BE2 JE 3BEB ; If it is 1FH move on, else
8000:3BE4 POP AX ; Clear stack & get message
8000:3BE5 MOV AX,3C12 ; AT Segment FC01
8000:3BE8 JMP 08AC ; & exit
8000:3BEB MOV BYTE PTR [032B],0 ; Boot status flag
8000:3BF0 CALL 08C9 ; Check for User abort
8000:3BF3 OR AL,AL ; Test for 0FFH
8000:3BF5 JNZ 3C21 ; If yes, then leave
8000:3BF7 PUSH ES ; Save for later
8000:3BF8 XOR AX,AX
8000:3BFA MOV ES,AX ; Set to Segment 0
8000:3BFC MOV DI,0400 ; & address 0400H
8000:3BFF MOV CX,0005 ; Transfer 5 sectors
8000:3C02 CLD ; Clear direction flag
8000:3C03 PUSH AX ; AL is sector 0
8000:3C04 PUSH CX ; Save for now
8000:3C05 MOV BL,0 ; Port address offset
8000:3C07 CALL 3248 ; Set Sector
8000:3C0A MOV CX,0200 ; # Bytes/sector
8000:3C0D MOV BL,0 ; Port address offset
8000:3C0F CALL 3243 ; Input
8000:3C12 STOSB ; Store in memory
8000:3C13 LOOP 3C0D ; Tsf all 512 bytes
8000:3C15 POP CX ; Obtain # of sectors
8000:3C16 POP AX ; Obtain sector #
8000:3C17 INC AL ; Move to next sector
8000:3C19 LOOP 3C03 ; Tsf all 5 sectors
8000:3C1B POP ES ; Restore original
8000:3C1C MOV BYTE PTR [032B],FF ; Set status flag
8000:3C21 RET ; Finished
8000:3C22 DB 7,0D,0A,'BEDISK Board '
8000:3C32 DB 'does not respon',E4
8000:3C42 DW DB10 ; Checksum adjustment
8000:3C44 (RETURN)
```

```
07 D3 F5 3E 80 D3 FE
```

When the BOOT ROM does its diagnostics, it takes a check sum of itself and compares it to 0000. If the checksum is not 0000, you will be rudely told. So we must make an adjustment. That is the purpose of the word at 8000:3C42.

You are now ready to save the modified monitor to a disk file. Use the R command to change BX to 0000 and to change CX to 4000. Use the N command to give the file a name. I suggest the name MMON.DOC (Modified MONitor). The command would be:

```
-NMMON.DOC
```

```
-DCS:3CC0 3D4F
xxxx:3CC0 C8 F6 26 DE 02 50 E8 75-D9 C3 C6 06 09 00 CE B0
xxxx:3CD0 00 B3 00 E8 72 F6 E8 6A-F6 3C EB 75 07 E8 63 F6
xxxx:3CE0 3C 1F 74 07 58 B8 12 3C-E9 C1 CC C6 06 2B 03 00
xxxx:3CF0 E8 D6 CC 08 C0 75 2A 06-31 C0 8E C0 BF 00 04 B9
xxxx:3D00 05 00 FC 50 51 B3 00 E8-3E F6 B9 00 02 B3 00 E8
xxxx:3D10 31 F6 AA E2 F8 59 58 FE-C0 E2 E8 07 C6 06 2B 03
xxxx:3D20 FF C3 07 0D 0A 42 45 44-49 53 4B 20 42 6F 61 72
xxxx:3D30 64 20 64 6F 65 73 20 6E-6F 74 20 72 65 73 70 6F
xxxx:3D40 6E E4 10 DB FF FF FF FF FF FF FF FF FF FF FF
```

Getting Started With . . . Turbo C

*Alan R. Neibauer
11138 Hendrix Street
Philadelphia, PA 19116*

Over the last few years, the C programming language has earned quite a following. Many C advocates are professional programmers who find C a powerful development tool. Others are enthralled hobbyists who just enjoy working with the language. But perhaps the most interesting phenomenon is the growth of C fanatics, professionals and amateurs alike who tout C as the "language of the future", the one language that can do everything. I wouldn't be surprised if we started seeing C bumper stickers and encounter groups along with the myriad of articles pitting C against Pascal, Fortran, COBOL, and, of course, BASIC.

Well, friends, C is not the perfect language. Sure it's powerful; it compiles to tight, fast code; it lends itself to structured techniques. But if you're already used to BASIC or Pascal, you'll have quite a bit of new learning to do before you can program in C -- especially if you've never worried about libraries, linking, memory models, include statements, and the like.

One of the most highly touted advantages of C is portability -- the ability to move your source code from one machine to the next, one compiler to the next. This is true as long as the receiving compiler's library contains the same functions as those you developed the program on. Otherwise you'll find that some time-saving function you used turns into a long list of compiler errors.

But whatever camp you're in, things became much easier when Borland International came out with its own version of C. Here was a C compiler in the Borland tradition -- inexpensive, relatively easy to operate, complete with editor and compiler. For example, all of the sample prog-

rams used in this article were compiled without specific include statements, without worrying about which libraries were needed, and with taking advantage of Turbo C's error reporting abilities.

So armed with Turbo C, the Borland manuals, and a good C programming text (a real necessity if you've never used C before), you're ready to see what this C language is all about.

Like Pascal, C can be a "goto-less" language. Certainly, structured programmers frown on the uncontrolled use of Gotos as a cheap tool to make up for lack of planning and good habits. But it is more than that. C is a language that encourages good programming skills. It does so by providing a full range of structures (like the three repetition statements For, Do, and While), and by requiring the programmer to declare variables and constants before writing the program.

If you've only used BASIC you might wonder why declaring variables fosters better programming habits. Well, like I tell my programming students, it forces the programmer to think out the program before sitting down at the keyboard.

In this article you'll learn the basics of C. You will learn how to enter, edit, compile, and run some simple programs. You'll also learn the fundamental structure of C programs and the most common language elements.

Keep in mind that Turbo C is a complete language that offers very few constraints to the professional developer. So we'll only have space and time to cover the most rudimentary elements.

If you have another version of C, you'll still be able to use this article for learning about the C language. The way

the Turbo screen appears, and how you interact with Turbo to edit and compile programs will be different than your own system. But the underlying program concepts are the same.

Installing Turbo C

The Turbo manual explains how to install C on your system. Basically, if you have a hard disk system, you are instructed to create a directory called "turbo" then copy all of the C files onto that directory from the distribution disks. You can create separate subdirectories for the library and include files. But Turbo will work just the same from one subdirectory.

If you have a floppy disk system, first format a blank diskette as a system disk (Format b:/s). Then copy the files TC.EXE and TCHHELP.TCH onto that disk from the distribution disks. Call this your program disk.

You'll need another floppy disk to hold the include (files ending with the .H extension) and library files -- COX.OBJ and those with the .LIB extension. I'll show you how to use these in a moment.

In either case, after you install Turbo, store the original distribution disks in a safe location.

Starting Turbo C

1. If you have floppy disks, put the Turbo C system disk in drive A. With a hard disk system, log onto the C directory with the command `cd\turbo` (or whatever you called your Turbo C directory).
2. Type TC then press Return.
3. Press the spacebar to remove the copyright message from the screen and to see the main Turbo menu shown in Figure 1.

Take a minute to look at the four

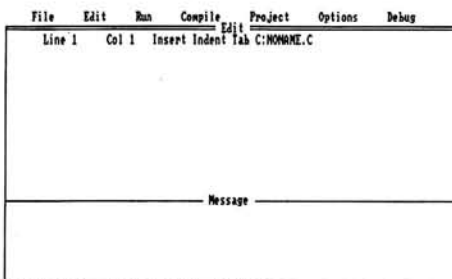


Figure 1

parts of the screen. Across the top is the menu bar where actions are selected or options set. The File option is selected as the default when you start Turbo. Below that is the main editing window. It is here where you display, enter, and edit your programs. (The status line at the top of the window shows you the position of the cursor.) The bottom window, under the line marked *Messages*, shows you any error messages generated when you compile or execute your program. The prompt line at the bottom shows how Borland has programmed some of the function keys for special uses.

The windows and pull-down menus provide a complete environment in which you can enter, edit, compile, and debug C programs. However, it might take a little while to feel comfortable getting around in this so review the basic keystrokes needed.

Select from the menu bar by either pressing the first letter of the option or by using the arrow keys to highlight your choice then press Return. If a pull-down menu appears, make your choice the same way -- the initial letter or highlight the option, then press Return. Press ESC to return to the main menu bar.

If you are working on a program in the edit window, press F10 to return to the menu bar. After a little practice these keystrokes will seem automatic. Press F6 at any time to return to the edit window.

C Structure

Every C program has the same fundamental structure. Take a look at the outline below:

```
(include files and global declaration)

main()

    (local declarations)
    {
        (program statements)
    }

(function definitions)
```

That's what a C program looks like. The items in parentheses are optional, although you'll need at least some program statements, and probably some variables,

if the program is to do anything. For you Pascal programmers, the brackets perform the same function as Pascal's begin and end statements. Use { where you would normally have Begin, and } for end statements. No program name is needed.

Now let's enter the simple C program shown in Listing 1.

LISTING 1

```
main()
{
    puts("My first C program");
    puts("is a simple one but it works");
}
```

1. Press E or F6 to select the Edit option from the prompt line. (You also could have used the right arrow key to first highlight the word Edit, then press Return.) The cursor moved into the edit window.
2. Type main() then press Return twice, once to end the line, again to enter a blank line on the screen.

Turbo provides a full-screen editor. So if you make any mistakes, use the directional arrows, delete and backspace, to correct them -- just as you would with a word processor.

3. Type { then press Return.
4. Press Tab then type puts("My first C program"); then press Return.
5. Press Tab then type puts("is a simple one but it works"); then press Return.
6. Type }. Your program should look like Listing 1.

Main() is the name of a C function which identifies the core of your program. In fact, every program is a collection of functions, either those provided by your compiler (like MAIN and PUTS) or ones you create yourself -- similar to Pascal's functions and procedures.

Puts() is also a C function. It is a low level function, closely related to DOS function calls, that displays text on the screen. Think of it as meaning "put string."

Setting the Environment

When you compile a program, Turbo must collect sections of code from library and include files, and link them with your own program. If you're using a floppy disk system, or subdirectories on your hard disk, you must let Turbo know where these files are located. (This isn't necessary if you have a hard disk and put all of the C programs in one directory.)

If you're in either of these situations, let's change the C environment so you can compile your program.

1. Press F10. This takes the cursor out of the Edit window and returns you to the menu bar on top of the screen.
2. Press O for the options command, then E to select Environment from the pull down menu. Another box of options appear. You're interested in the first three: Include directory, Output directory, and

Library directory.

Each of these must be set for your particular system. If you have two floppy disk drives, each of these should be B: followed by the specific subdirectory. With hard disk systems, the subdirectory should be listed.

3. Highlight each option in turn, press Return, and enter the appropriate drive and directory.

For example, if you set up your system exactly as recommended in the Turbo manual, change your environment as shown here:

Floppy disks:

```
Include directory: B:\INCLUDE
Output directory: B:\
Library directory: B:\LIB
```

Hard disks:

```
Include directory: C:\TURBOC\INCLUDE
Output directory: C:\TURBOC
Library directory: B:\TURBOC\LIB
```

Compiling and Running A Program

Unlike BASIC, an interpreted language, C programs must first be compiled to run. With most other languages this can mean a time consuming wait while each line in your program is translated into machine code. Fortunately, Turbo C can compile your source code at an incredible rate of 7,000 lines per minute. So for short programs you can easily forget that Turbo is a compiler at all.

When you compile a program, without executing it, Turbo just checks it for syntax and creates an object code (OBJ) file on your disk. However, to execute it, the compiler must actually create the binary (EXE) file. Unlike Turbo Pascal, C cannot execute a program from memory. That's because code for the commands and functions you use must be retrieved from library files.

1. If you just set your environment, press the left arrow key twice to highlight the compile option. If you are still in the editor, press the F10 function key then C to select the Compile option. You'll see a pull down menu with a list of compiler options. The line Compile to OBJ C:NONAME.OBJ is highlighted.

2. Press C to compile the program. For a brief second, a status box will appear on the screen showing the name of your program and the number of lines being compiled. If there are no errors, the message *Success Press any key* will appear.

3. Press any key to return to the C windows.

If you looked on your disk now you'd see the file NONAME.OBJ. That's the object code for the file in your editor. When you compile a program this way it will not be executed, or run, automatically. This is a good way to test a long program for errors before actually running it. So let's

now try running the program. To do this, Turbo has to create the EXE file, linking your object file with the appropriate code from the C libraries.

4. Press R for the run option. The compilation box appears again but this time it reports which Library files are being linked with your object file. When it is completed, the screen will clear and your program will be run. The message *Press any key to return to Turbo C* appears on the bottom of the screen.

5. Press any key.

You can also create the EXE file without actually running the program from the compile menu. Just select the option *Make EXE file*.

Now let's edit the program, intentionally adding an error so we can see how Turbo handles mistakes.

6. Press F6 to edit the program code.

7. Use the directional arrows to move the cursor to the semicolon at the end of the first puts command.

8. Press Del to delete the semicolon from the screen.

9. Press F10 then R to compile and run the program.

Because an error was detected, however, the output is not displayed. Instead, the message Errors appears at the bottom of the compile window.

10. Press any key to remove the compile box. The cursor is in the bottom window and you'll see the error message listed:

```
Statement missing ; in function main
```

The line with the error, or one near it, will also be highlighted in the edit window.

11. Press F6 to edit the source code, then the up arrow and END to move the cursor to the end of the line in question.

12. Type the semicolon then press F10 R to compile and run the program successfully.

13. Press any key to return to Turbo.

Move between the edit and message windows by pressing F6. If your program has more than one error, return to the message window to scroll and display each error message. Then press F6 to correct your code.

Saving Your Programs

Get into the habit of saving your programs frequently as you work on them. You never know when some hardware, or software, problem may lock up your system, causing you to reboot and lose anything you've typed without saving.

1. Press the F S from the menu bar. A small box will appear labeled, Rename NONAME.

2. Type Program1 then press Return. Turbo will add the extension .C to the program and save it on the disk.

3. Press F N to select the new option

from the file menu. This erases the program from the edit window so you can start over, like the NEW command in BASIC.

Recalling Programs

You can recall a program for editing, running, or compiling at any time. Turbo makes this easy for you in several ways. First, it stores a list of the programs you most recently loaded into the editor in a special Pick area. Second, you can display a directory of all C programs then use the cursor keys to point to the one you want loaded into the editor.

Let's try both methods now.

1. Press F10 F P for the File Pick option. A small box labelled Recent Files appears.

This box will contain the names of the last several programs you loaded into the editor. To recall a file from the Recent Files box, use the arrow keys to highlight the program name then press Return.

2. Press Esc for now to return to the File menu.

3. Press L for the Load option. The Load File Name box appears with the prompt *.C. If you know the name of the program you wish to load, type it here then press Return. But let's say that you just can't remember what you called it.

3. Press Return. A directory listing of all C programs appears.

4. Press the arrow keys to highlight the file you want to load, then press Return.

While these are the basics you'll need to enter, run, and compile C programs, there's actually a lot more to the Turbo environment. But for now, let's spend some time on the C language itself. Start by pressing F N for the New option. (This clears the screen and puts you into the edit mode).

Elements of C

You've already seen the fundamental structure of a C program. So let's use it to examine what programming in this language is like.

Program Structures

Every programming language has three fundamental structures, or way of doing things: sequence, repetition, and selection. Sequence means a series of steps that are performed one after the other. With repetition, a series of instructions are performed several times. Selection makes a choice, it determines whether some instruction is performed or not. Not every program has to have all of these three structures, but all have at least one and usually others in combination.

C not only has these structures but several types of each. For instance, there are two selection techniques (IF and SWITCH), three repetitions (FOR, DO, and WHILE) and a number of ways to program in sequence using functions.

Let's start by writing a program that

demonstrates the use of declared variables and a simple sequence. You should still be in the editor. If not, press E from the menu bar to select the Edit option.

1. Type the following program. Pay particular attention to the punctuation. The indentation, while not necessary to C, just makes the program easier to read and follow.

```
main()
{
    float grade1, grade2, grade3, sum, average;

    grade1 = 95;
    grade2 = 90;
    grade3 = 85;
    sum = grade1 + grade2 + grade3;
    average = sum/3;
    printf("Your average is %f", average);
}
```

2. Press F10, then R to compile and run the program. If you made any typing mistakes, the editor will reappear with the appropriate error message. Check your program against the one shown here, make your corrections, then try running the program again. Once it compiles successfully you'll see this on the screen:

```
Your average is 90.000000
```

This is the output of your program. Now let's take a good look at what you've typed and see how to display the average a little better. Press any key to return to Turbo, then E to enter the editor.

The line

```
float grade1, grade2, grade3, sum, average;
```

is the variable declaration. It is here where you are stating that your program will use five variables that are all type float, decimal number. Turbo C recognizes many variable types. The most common are Float (decimal numbers), Int (whole numbers), and Char (single characters and character arrays-strings). The variable declaration always follows the same pattern:

```
type-of-variable variable-names
```

Some other valid variable declaration are:

```
int principle, years
float interest_rate
char rating, name[10]
```

When declaring a string, an array of characters, you must tell Turbo the maximum length it can be. Here the variable name can be no more than 10 characters long.

This program then has five "assignment" lines, similar to BASIC Let commands. Two of the assignment lines assign a value through a mathematical formula. In one case, the variable SUM is assigned the total of the three grades. Then the sum is divided by three (the number of grades) and assigned to the variable AVERAGE.

Finally, the results of the program are displayed on the screen with the Printf command, a formatted output command. Take a good look at this line.

```
printf("Your average is %f", average);
```

The text to be displayed is surrounded by quotation marks. However, %f (called the conversion character) is used to specify that the value of a float type variable will be displayed -- the variable listed after the quotation mark. When the line is executed, the value of average is substituted for %f and displayed on the screen. If you use variables of other types, the output specification must agree:

```
%d integer numbers
%u unsigned integer
%e floating point in exponential format
%c character
%s string -- an array of characters
```

Use the PUTS function when you want to display text only, no variables. If you have variables to display along with the text, use PRINTF.

Now let's look at a slight modification of the same program, shown in Listing 2. Here a constant definition is used to store the number of grades 3. A constant is different from a variable in that its value cannot change in a program. The printf line has also been changed slightly. The output specification is now %5.2f. The numbers tell Turbo how to display the value of the variable. In this case, the average will take up five spaces with two decimal positions -- 90.00.

LISTING 2

```
#define number_of_grades 3

main()
{
    float grade1, grade2, grade3, sum, average;

    grade1 = 95;
    grade2 = 90;
    grade3 = 85;
    sum = grade1 + grade2 + grade3;
    average = sum/number_of_grades;
    printf("Your average is %5.2f", average);
}
```

SELECTION

C has two selection commands, IF and SWITCH (like the Pascal CASE statement). While they are generally interchangeable, the SWITCH statement is normally used when more than two

choices are possible. Let's first take a look at the IF statement.

own { and } punctuation as shown in Listing 3.

```
IF (condition-is-true) perform-this-instruction
```

The word THEN is not used in C.

The condition can be any of these type:

The { and } mark a compound statement -- more than one instruction that must be performed if the condition is true.

```
== Equal to as in AGE == 42
< Less than as in A<10 (A is less than 10)
> Greater than as in B>0 (B is greater than 0)
!= Not equal to as in C! = 8 (C doesn't equal 8)
<= Less than or equal to as in A<=10 (A is less than 10 of equal to it)
>= Greater than or equal to as in B>=0 (B is greater than 0 or equal to 0)
```

The sample statement given above is called a simple If statement in that if the condition is true, only one C instruction is performed. Before looking at more complex statement, modify your program by editing the printf statement and adding a line as shown here:

```
printf("Your average is %5.2f \n", average);
if (average > 60) puts("You passed!");
```

Now press F10 R to compile and run the program. What appeared in the output? It should be the message You passed! That's because the statement IF AVERAGE > 60 is true. If the average had been 60 or less, the condition would have been false and the message would not print.

Also notice the change to the printf line. The \n is a special instruction to the compiler that stands for "new line" -- a carriage return line feed (like the LN in Pascal's Writeln command). Using printf, this is needed so the words You passed appear on a separate line than the average message. PUTS, by the way, performs its own carriage return/line feed.

Now let's look at more complex IF statements, where more than one statement must be executed. These are called compound statements and require their

Modify your program as shown in the listing, then run it. In this case, the average and the message is only printed when the condition is met. Of course there's one major problem with this program -- it doesn't tell you if you failed.

We'll take care of this with the Else statement:

```
if (average > 60)
{
    puts("You passed!");
    printf("By %5.2f points", average-60);
}
else
    puts("You failed");

The Else statement is performed only if the condition is false. While it is a simple statement in this case, a compound can also be used such as:

if (average > 60)
{
    puts("You passed!");
    printf("By %5.2f points", average-70);
}
```

LISTING 3

```
#define number_of_grades 3

main()
{
    float grade1, grade2, grade3, sum, average;

    grade1 = 95;
    grade2 = 90;
    grade3 = 85;
    sum = grade1 + grade2 + grade3;
    average = sum/number_of_grades;
    printf("Your average is %5.2f \n", average);
    if (average > 60)
    {
        puts("You passed!");
        printf('By %5.2f points', average-60);
    }
}
```

```

else
{
    puts("You failed");
    puts("See your advisor immediately");
}

```

Switch Statement

The switch statement does about the same thing, just in a different way. It is a menu-type approach where all possible situations -- all cases -- are listed along with the C instructions to be performed. It's a little more complicated so I've shown a sample program in Listing 4. Notice that an integer variable, AVE, has been added along with the statement `ave = average / 10;` just before the switch statement. Real types cannot be used for the switch conditions so this line converts the real type AVERAGE to the integer type AVE.

The switch statement itself is relatively easy to understand. SWITCH (ave) starts it off saying that the value of the variable AVE is being used in each condition. The possible states of that variable are then listed below, using the syntax:

```

CASE value_to_be-checked: C statements;
                        Break;

```

After each possible state is a list of C instructions to be performed if the condition is true. Unlike the Pascal CASE statement, C will not automatically perform the statements then end the SWITCH command. Without the Break command ending each switch case, every C statement from the correct case down would be executed.

The Default is used in the event none

of the individual cases listed are true. This prevents the program from "falling through", passing the Case statement without doing anything.

REPETITION

Now let's try repetition. Repetition is a little more complex to demonstrate because there are several ways to repeat instructions. You can repeat lines using the FOR, DO, or WHILE instructions. Each of these have their own uses and powers, so it's really not difficult to pick the wrong one.

The FOR instruction is always used when you know exactly how many times you want to repeat a step or series of steps. If you don't know the number, then use either WHILE or DO. In this case, use DO when you want to perform the commands at least one time. Use WHILE when you might not want to perform the lines at all.

This will be easier if we look at some examples. Listing 5 is a program that lists the numbers from 1 to 10, along with their squares and cubes. FOR was used because the program was designed to list the numbers from 1 to 10, never more or less.

The syntax of simple and compound For statements follows:

When the program begins, it assigns the starting value to the variable counter, called the control variable. It then performs the following C instruction (or compound statements surrounded by { and } as long as the continuation condition is true. In this case, it prints the value of counter, its square and cube, as long as counter is less than or equal to 10.

Before repeating the instruction again, it adds 1 to the value of counter, making it 2. It checks to make sure the continuation condition is still true and, if it is, repeats the lines. It does this again to assign counter the value of 3, then again to assign it 4, then again to assign it 5. It repeats this until the continuation condition is false, when it stops the repetition.

The value of counter is incremented in the statement `counter++`. This is similar to the PASCAL command `Counter := Counter+1` or Basic's `C=C+1`. Because the plus marks follow the word counter, it is incremented after the first repetition. If it had been written `++counter`, the variable would have been incremented first, so the first line displayed would have been 2 4 8 instead of 1 1 1.

But what if you don't know how many times you'd like to repeat the loop. Say, for example, that you want to start

```

FOR (control-variable=starting-value, continuation_condition,
    increment_index or modification_of_index)
statement;

FOR (control-variable=starting-value, continuation_condition,
    increment_index or modification_of_index)
{
    statement;
    statement;
}

```

LISTING 4

```

#define number_of_grades 3

main()
{
    float grade1, grade2, grade3, sum, average;
    int ave;

    grade1 = 85;
    grade2 = 90;
    grade3 = 85;
    sum = grade1 + grade2 + grade3;
    average = sum/number_of_grades;
    ave = average / 10;
    switch (ave)
    {
        case 9: puts("You got an A"); break;
        case 8: puts("You got a B"); break;
        case 7: puts("You got a C"); break;
        case 6: puts("You got a D"); break;
        default: puts("You failed!");
    }
    printf("Your average is %5.2f \n", average);
    if (average > 60)
    {
        puts("You passed!");
        printf("By %5.2f points", average-60);
    }
}

```

LISTING 5

```

main()
{
    int counter;
    for (counter=1; counter <=10; counter++)
        printf("%2d %4d %6d \n", counter, counter*counter,
            counter*counter*counter);
}

```

LISTING 6

```

main()
{
    float f,c;

    puts("Enter a fahrenheit temperature");
    scanf("%f",&f);
    c = (f-32.0)/(5.0/9.0);
    printf("\n%5.2f degrees fahrenheit is %5.2f centigrade", f,c);
}

```

at 1 and keep on squaring numbers until the square is greater than 1000. In this case you know that the square of one is not over 1000 so you want to square at least one number. This calls for the DO command:

```
Counter = 1;
DO
{
    printf("\n %4d    %6d", counter, counter*counter);
    counter++;
}
WHILE (COUNTER*COUNTER < 1000)
```

The first line starts the variable number at the value 1. This is necessary since, unlike BASIC, variables do not automatically start at 0 when you start programming.

The word DO starts the repetition in which all instructions between it and the WHILE line are performed as long as the condition is true.

The DO command is like Pascal's REPEAT.

Now we've used the repeat command here because we knew that one time one was less than 1000 -- we were sure we wanted to perform the look at least one time. But what if we weren't sure of that first answer? In this case, we'd use the WHILE instruction:

```
counter = 1;
WHILE (counter * counter < 1000)
{
    printf("\n %4d    %6d", counter, counter*counter);
    counter++;
}
```

Here's how this works. The condition in the WHILE command is first tested. If the condition is true then the statement following (or the compound statement as in this example) is performed. The condition is tested again before each repetition.

Here, the repetition starts because the condition is true. It performs the compound instruction of outputting two numbers then incrementing the variable Number. This repetition will continue until the condition is false, or Number squared is not less than 1000.

The main difference between DO and WHILE is when the condition is tested, before the first execution of the loop or after.

Before getting out of Turbo, let's take a brief look at some slightly more advanced concepts. Every program actually has three parts to it -- the input, the process, and the output section. The input

section assigns or otherwise gets our information into the computer. The assignment COUNTER = 1 is really a way of getting the number 1 into the computer. The Output section displays the results of what your program does on the screen. PUTS and PRINTF statements are output lines. The process part is what must be done to convert the information put into the computer into the information sent

out. These lines are usually mathematical formulas. As example, take a look at Listing 6.

This program gets its input from the keyboard while the program is being executed. The SCANF command allows the user to enter data, meaning that the values of the variable can change each time the program is run. This makes the program much more useful than one in which the values are assigned.

SCANF is a formatted input command, so you must enclose the proper conversion characters in quotation marks, followed by the variable to be input. However, notice the variable is given as &f. The ampersand means "the address of" -- so the value entered at the keyboard goes into the address held by the variable f. Use of the ampersand is closely related to pointers -- a subject that's important to C but one too complicated for an initial Getting Started article.

SCANF pauses the program, waiting for the person running the program to enter something from the keyboard, in this case a number. When the person types a number then presses the Return key,

Turbo places the number in the address of the variable f, the same as assigning the value to f.

Anyway, after the value is entered, the program calculates the centigrade temperature and displays the results.

Try typing the program and running it. You'll see the prompt appear that you typed in the PUTS command. Type some number then press Return.

Now let's take this example one step further. Suppose you might want to do the temperature conversion more than once -- a repetition. But which repetition do we use? Since we don't know the number of repetitions, we can't use the FOR command. So we have to select either WHILE or DO.

Listing 7 first asks the user if they want to convert a temperature. If they enter any character other than Y or y, then they don't want to perform the loop even one time. Notice that the question is repeated at the end of the loop (but inside of it) to determine if the user wants the program continued. This program contains several new elements so let's go through them.

The function getch() is a way to input a single character from the keyboard. In this case, the character input is assigned to the character variable cont. It is then tested in the WHILE statement:

```
while ((count == "Y") || (count == 'y'))
```

The double line between the conditions is C's OR syntax. (AND is &&.) Using this command, the loop is repeated as long as the variable cont is either Y or y, the two valid positive responses. Remember that two equal signs (==) are needed to test for equality in C.

Listing 8 is another version of the same program. See if you can tell the difference.

Here, only a capital Y is accepted as the positive response, in the line:

```
while ((cont=getch()) == 'Y')
```

This one line inputs the character, assigns it to the variable cont, and tests it at the same time. Why couldn't you use the same approach for both upper- and lower-case letters such as:

Continued on Page 50

LISTING 7

```
main()
{
    float f,c;
    char cont;

    puts("Do you want to convert a temperature, Y or N?");
    cont=getch();
    while ((cont== 'Y') || (cont== 'y'))
    {
        puts("Enter a fahrenheit temperature");
        scanf("%f",&f);
        c = (f-32.0)/(5.0/9.0);
        printf("\n%5.2f degrees fahrenheit is %5.2f centigrade\n",
            f,c);
        puts("Do you want to convert a temperature, Y or N?");
        cont=getch();
    }
}
```


The Ultimate Cheap Desktop Publisher

A Review of NewsMaster II

Pat Swayne, HUG Software Engineer

In the March 1987 issue of REMark, I reviewed a program called Fontasy, which was one of the first successful attempts at a low budget version of desktop publishing. With Fontasy, you could compose fairly nice looking newsletters and print them on an ordinary dot matrix printer. It used the graphics mode of the printer to produce characters that resembled, at least somewhat, those produced by a printing press. And you could also put pictures in your newsletter, selected from clip art libraries provided by the manufacturer. But Fontasy was difficult to use, and it was also very memory hungry, making it useful only for producing small newsletters.

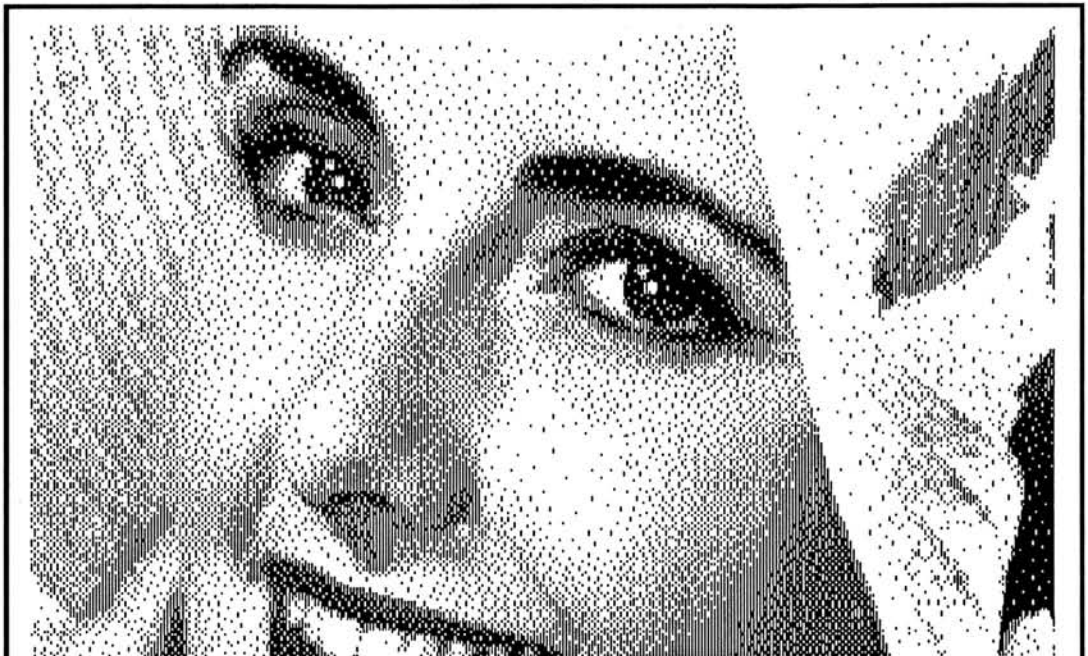
Since then, other low-end desktop publishing programs have appeared, including NewsRoom Pro, PFS:First Publisher (also known as Logitech Publisher), and NewsMaster. Among these, PFS:First Publisher is closest to the high-end programs (see "Z-100 Desktop Publishing: Logitech Publisher" in the September 1988 REMark), but I have three things against it. The first is that it is somewhat difficult to learn. The second is that it is slow, and you cannot scroll quickly around a page.

The third thing I don't like is that it forms the various fonts using a fairly coarse pixel size, and then attempts to smooth the fonts when you print (if you want it to). The smoothing algorithm used in the program, quite frankly, stinks. It sometimes distorts text characters, and since it is applied to both text and artwork, you may have to sacrifice picture quality in order to get smooth text.

Of the programs mentioned above, NewsMaster is by far the easiest to use, and it is also very fast. However, it is very limited compared to other desktop publishing

programs. Although it can divide a page into a headline area and columns, the headline area the margins, and the "gutter space" between columns are of a fixed size. It only prints in one mode, and it can only use its own clip art, or clip art from PrintMaster, another program from the publisher, Unison World.

Now Unison World has released NewsMaster II, a greatly enhanced version which is just as fast and easy to use as the old NewsMaster. NewsMaster II allows you to alter the gutter space, headline area, and margins, and it provides



With the NewsMaster II CAPTURE utility, you can capture images produced by other programs, such as this picture from PC Palette, and use them as illustrations.



CAPTURE works in the CGA, EGA, and Hercules video modes. This picture was created by PC Paint in the 640x350 EGA 16-color video mode.

four print modes ranging from draft quality to high quality with smoothing. The smoothing can be applied to both the text and graphics, or just to the text, and it seems to work a little better than the smoothing done by PFS:First Publisher. (Perhaps that is because the default pixel size is smaller.) NewsMaster II also comes with an absolutely fantastic utility called CAPTURE, that allows you to capture screen graphics and turn them into NewsMaster graphics. It supports CGA, EGA (it runs in a 640x200 graphics mode on both of these cards), and Hercules graphic modes. The combination of NewsMaster II and CAPTURE work so well at translating screen images into printed images that you may want to buy NewsMaster II just to use it as a screen dump utility. In addition to making all of these improvements, Unison Word has lowered the list price from \$99.00 to \$79.00.

Installation and Set Up

NewsMaster II comes on 2 360k 5.25" disks, and comes with an 86 page manual, a printed sheet showing samples of all the fonts and supplied clip art, and various other printed sheets including one containing the installation instructions. Installation is very simple, especially if you have a hard disk.

A supplied utility called HARDDISK makes the required sub directories for you and copies all of the files you need to them. If you have a high capacity disk drive, you can use HARDDISK to install the files on it. Just remember that they will be placed in subdirectories, and you will have to change to a sub-directory before you can run the program.

After you install NewsMaster II on your system, you run a utility called NMCONFIG to select your screen and printer types. Actually, NMCONFIG usually figures out what kind of video card you have and indicates so with an arrow pointing to your type within a list, so all you have to do is press Return. Then NMCONFIG allows you to select your printer type from several "pages" of different types. NewsMaster II probably supports as many different printers as any program of this type. Even the old MPI in my office is supported -- though the resolution is so poor that you would not be able to read this font if I had used the MPI to print it.

You may have to experiment with different printer types to get the best result. Even though the Alps P2000 printer is listed in the menu, the Alps P2000G is not, and I found that the Alps P2000G

printers we have here at HUG headquarters did better when I selected Epson FX85 from the menu than when I selected Alps P2000. Many printers are Epson compatible, so you may want to try the Epson FX85 selection with your printer in addition to its own selection, if it is listed.

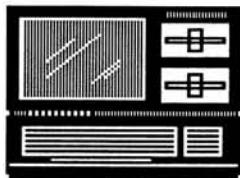
As I mentioned before, NewsMaster II can print in four different modes, which are Draft, Quality, Quality with smoothed text, and Quality with smoothed text and graphics. However, the resolution of your printer is what really determines how NewsMaster II will print. On an MPI-150, for example, it does not matter which mode you select, because they all come out the same -- not very good. On an Epson MX-80 with Graf-

trax, the Quality mode is darker than the draft mode, but the resolution is not any better, and the smoothing modes have no effect. If your printer has the resolution of an Epson FX-85 or better, you will be able to take advantage of the smoothing modes. One curious thing I found is that the draft mode of an MX-80 is better than the draft mode of an FX-85. So if you have a printer that is FX-85 compatible, it will also most likely be MX-80 compatible, and you can re-configure it as an MX-80 if you want a printout that is faster than the quality mode but better than your normal draft mode.

NewsMaster II supports laser printers made by Kyocera and Hewlett Packard. Laser printers are driven in the graphic mode, and laser fonts are not used (NewsMaster fonts are printed as graphics).

Operation

NewsMaster II is entirely menu driven, with the function keys used to select items from the menus. Some menus, such as those used for selecting files, are text menus, but most of the menus use a system of icons (pictures) to represent the items. For example, two of the icons on the main menu are a picture of a floppy disk, which selects the file load/save



Z-100 owners aren't completely left out. NewsMaster II runs under ZPC with no patches, but, alas, CAPTURE does not.

option; and a picture of a printer, which is (obviously) the icon to select if you want to print a newsletter. Some of the icons are not so clear as those examples as to their function, but they are easily learned. If you do not care to learn them, there is an excellent tree diagram of all of the menu functions in an appendix in the manual, with the functions indicated in English, not pictures.

There are two edit modes that can be selected from the main menu: the text edit mode and the artwork edit mode. In the text edit mode, you can type in text, select fonts, select justification of the text (left, right, full, and center), and set up the layout of the current page. A page can be divided into a headline area and 1 to 10 columns. You can adjust the size of the headline area, or eliminate it completely, and you can adjust the size of the margins around the page and the gutter space between columns. The adjustment unit is inches, and the program allows 3 places after the decimal, but I doubt if an adjustment of only .001 inch would make any difference in the output.

Text is justified according to your selection as you type it in, and if you go back and make a correction in an area before the end of the text, everything beyond the text is automatically adjusted, even if there are other pages beyond the correction. NewsMaster II has no provision for any kind of automatic hyphenation, but you can easily hyphenate manually. Just place the cursor where you want the hyphen to go, and type - and a space. This should be done when there is a long word at the beginning of a line, and a lot of "white space" in the line above. When you add the hyphen and space, the first part of the word will jump up to the end of the line above (if it fits), and reduce the white space.

NewsMaster II has the ability to import plain ASCII text, so if you have a word processor or editor capable of producing plain text files, you can use it to prepare text for your document. That will give you the opportunity to correct spelling, etc., before you commit the text to NewsMaster. Although it is faster than most other desktop publishing programs, it is not anywhere near as fast as a good word processor.

In the artwork edit mode, you can select and manipulate artwork and draw lines and boxes on your page. The artwork you can select from includes several small drawings included with NewsMaster II, the drawings included with PrintMaster, and any drawings you have saved from the screen using the CAPTURE utility. Once you have selected a drawing, you can change its size, crop it, alter its aspect ratio, or flip it horizontally or vertically. You can also control how text will flow around (or over) each drawing.

You can draw lines or boxes anywhere on your page, and you can select from any of 30 patterns to be used for drawing the lines or filling the boxes. You can also adjust the width of your lines. One nice feature is that you can go back to a line or box that you have previously completed and re-adjust its size, shape, or pattern. That way, you can "tweak" lines that you have drawn around artwork until everything is as you want it.

One menu item that is available in just about every menu is a selection that allows you to change the page view. The normal view shows about two thirds of the width of a page on the screen and about two inches vertically (three inches vertically on a Hercules display). The other views available are full width, full page, and zoom. The zoom view starts out as a two times magnification of the normal view, and you can select it repeatedly to double the magnification again each time, for a total of six times. At the highest magnification, each pixel is a rather large block on the screen.

There are short cut key strokes available to select many of the menu items. For example, you can change the page view by typing Alt-Z for the zoom view or Alt-N



This CGA color picture was converted to monochrome, and then CAPTURED.

for the normal view, etc.

The Capture Utility

One of the best things about NewsMaster II is the CAPTURE utility. It is a memory resident program that captures screen images and converts them to NewsMaster artwork files. To activate CAPTURE, you type Alt-G, and some instructions appear on the screen, along with a small X-shaped cursor. You then use the arrow keys to move the X to the lower left corner of the picture you want to capture and press Return. As soon as you start moving the X, the instruction message disappears, allowing you to see the entire screen. When you press Return, you can then move the X to the other corner of the picture and press Return again. You can therefore capture any part of a screen, or the whole screen. Before you press Return the second time, you can press the Ins key to switch the cursor between corners, so that you can adjust the selected area. After the second Return, CAPTURE asks you to select whether the background of the picture is black or not, and to select whether you want the aspect ratio of the picture altered to look right on the printer or left as is. Then it saves your selected picture.

The CAPTURE utility stores captured pictures in a NewsMaster II clip art library. Each library consists of three files with the extensions .SHP, .SDX, and .SDR. (Not all libraries have an .SDX file.) The CAPTURE library is called, as you might expect, CAPTURE. If you use CAPTURE repeatedly without changing the currently logged drive, each new picture will be ad-

ded to a single library file. You should check the size of the file occasionally to see that it is not getting larger than you would like it to be (you may want to copy it to a floppy disk). The .SHP file will be the largest one. If it is getting too large, just rename the three files to something else, for example, MYART.SHP, MYART.SDX, and MYART.SDR.

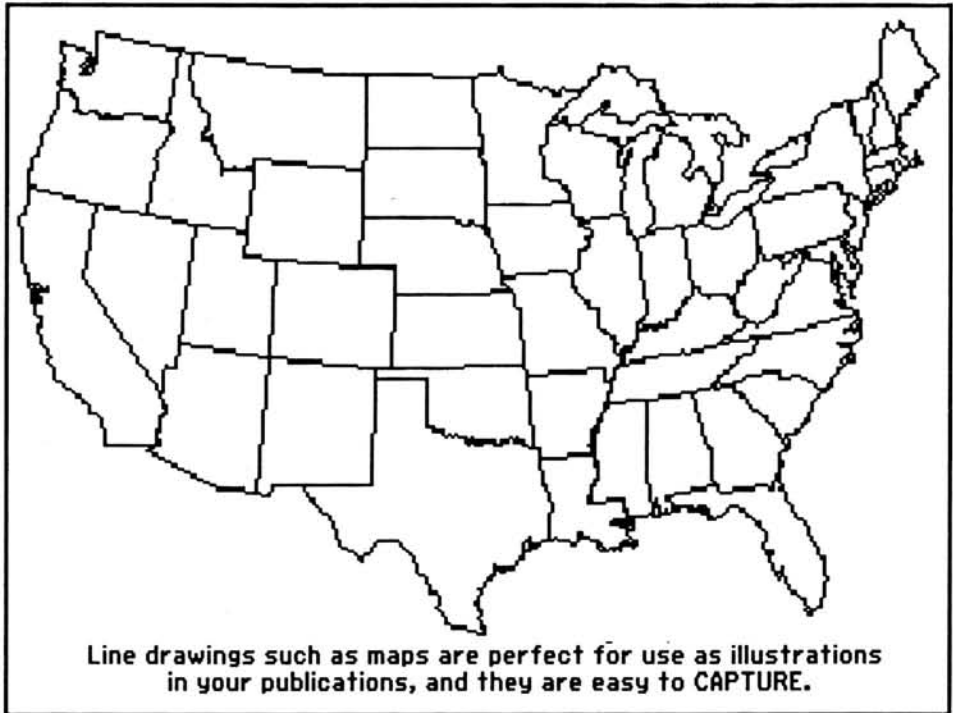
Tips and Tricks

While I was preparing this article, which was done entirely with NewsMaster II, I learned some things that should help other prospective users of the product.

As I stated before, you can control how text flows around a picture, but text will always flow over a line or box drawn in the text area of a page. If you want to put a box on the page and prevent text from flowing through it, there are two ways you can do it. One is to use a "blank" instead of a box. There is a clip art library called Blanks which contains some blank boxes, circles, etc. You can use one of those instead of drawing a box, and re-size it as needed. Or you can go ahead and draw your box manually, and then fill it with a drawing from the Blanks library called "Blank Box", which can be sized to fit your box.

If you want text to stop a certain distance from a box, you can use the Crop function of the artwork editor to extend the borders of the picture or blank inside of the box to beyond the box's sides. The borders of pictures are always shown as dotted lines on the screen, so you always know exactly where text will stop flowing with reference to each picture.

The borders around each picture are rectangular, even those around ovals and circles in the Blanks library. If you want to put a circle or oval on your page and have text flow right around the edge of it, here is a trick you can try. First set up a blank page by setting the number of columns to 1 and removing the headline area. Then get a circle or oval from the Blanks library and put it on your page. Then use the CAPTURE to capture several thin horizontal slices of the circle or oval, and then assemble the slices on the target page. You can use that technique with other odd shapes,



also, including drawings made with other programs.

NewsMaster II does not allow you to stretch, shrink, or otherwise alter font characters the way you can artwork. However, you can CAPTURE text from the screen, which in effect turns it into artwork, and then you can manipulate it anyway you want to. When you capture text, the page view mode should always be set to normal. (It's a good idea to be in the normal mode when capturing graphics, as well.) The reason is that in the normal view mode, the pixel usage of characters on the screen corresponds exactly to the default printer pixel usage. In other words, if a character will be printed using a matrix of, say, 10 pixels vertically by 8 pixels horizontally, it will be displayed using that matrix on the screen. (Since screen pixels are bigger than printer pixels, the characters on the screen are bigger, and they are also distorted in aspect ratio). The default printer resolution is 120 x 72 dots per inch, which is a resolution common to Epson compatible printers. If a printer supports a higher resolution, that higher resolution will only be used if smoothing is used. 120 x 72 may not sound like a very high resolution, but the fonts have been carefully designed to look good at that resolution.

If you want to put a caption under a picture that is wider than one

column on your page, you will have to use the trick of capturing text and depositing it as artwork, since a line of text cannot flow across the gutter between columns. I used that technique to caption the pictures in this article.

When you capture text from the NewsMaster screen, you should always tell CAPTURE to save the screen as it is, rather than letting it adjust the aspect ratio.

When you try to capture a color picture while your screen is in an EGA graphic mode, CAPTURE will try its best to determine what part of the picture to make black, and what part to make white. It is a good idea to capture EGA pictures twice, and indicate to CAPTURE that the background is black one time, and that it is not the second time, regardless of the actual background color. There are many ways to program an EGA card, and CAPTURE cannot always figure out when black is black. But sometimes it does a remarkable job at translating a color picture to black dots on white paper, as it did with the Mickey picture in this article.

When you capture a picture while the screen is in the CGA color mode, the CAPTURE will probably not be able to distinguish between the colors. But if the picture was made using GW-BASIC, it can be saved to disk using BSAVE, and then it can be re-loaded in the CGA monochrome mode. In this mode,

the colors will be interpreted as dot patterns, and the CAPTURED picture will look quite good when it is printed. Here is an example of the BASIC code that can save a CGA color picture and then reload it in the monochrome mode. It is assumed that the screen is in the CGA color mode and that there is a picture on the screen upon entry to this code.

```
10 REM SAVE A CGA COLOR PICTURE"
20 DEF SEG = &HB800
30 BSAVE "PICTURE",0,&H4000
40 REM SWITCH TO MONOCHROME
50 SCREEN 2
60 REM NOW, LOAD PICTURE
70 BLOAD "PICTURE",0
80 REM WAIT FOR CAPTURE
90 A$=INPUT$(1)
```

A Technical Trick

This information is intended for people familiar with DEBUG. The fonts used by NewsMaster II are stored in a file called NEWS.FLB. I snooped around in that file and discovered where the line spacing and column spacing for each font is stored. You can alter the spacing using DEBUG to add more space between characters or between lines for each font. I figured this out because the Park 8 font, which is used in this article, seemed to be a bit "crowded". All you have to do to alter the spacing for a font is to load

NEWS.FLB into DEBUG and then use the Search command to locate the name of the font. (I had trouble finding the Park 8 font at first, because there are two spaces between the "Park" and the "8" in the NEWS.FLB file.)

After you DEBUG finds your font, add 50 (hex) to the address of the name to get the column spacing byte, or 52 (hex) to get the address of the line spacing byte. The NEWS.FLB file is larger than 64k, so if DEBUG does not find your font, add 1000 (hex) to the DS register and try the search again, starting with 0 instead of 100 as you do with the first segment. Here is an example of altering the line spacing in the Park 10 font.

```
D>DEBUG NEWS.FLB
-S100,FFFE,"Park 10"
2305:010A
2305:0166
2305:01C2
-H10A,52
015C 00B8
-E15C
2305:015C 01.02
-W
Writing 3712D bytes
-Q
```

Notice that DEBUG found Park 10 three times. That is because there are standard, bold, and italic Park

10 fonts. If you alter one of them, you should alter the others as well.

If you have the HUG program HADES (Hug Absolute Disk Editing System), you can use it instead of DEBUG for modifying your fonts. Use the File mode to load NEWS.FLB, and put it in the ASCII mode to search for the font name. Then move 50 (hex) bytes (five lines down) from the name to find the character spacing, and move two bytes forward to find the line spacing.

Where to Get It

NewsMaster II is available from Unison World, a division of

Kyocera Unison, Inc.
P. O. Box 3056
Berkeley, CA 94703
ph. 1-800-443-0100, ext. 238

It retails for \$79.95, but I got mine for \$59.99 plus \$7.00 shipping from mail order house called Computer Mail Order (CMO). Their phone number is 1-800-233-8900. Their catalog no. for NewsMaster is ICUW01. Specify NewsMaster II when you order, because although the picture in the catalog is of NewsMaster II, the description just says NewsMaster. Perhaps they have both in stock, so be sure you get the right one.

Printed on an HP LaserJet II. *

Z-100 Graphics Software

DOODLER-V Graphics Package	\$99.00
with D-MOUSE Mouse Driver	\$109.00
with Logitech C7 Mouse & Driver	\$189.00
Font Library Disk for DOODLER-V	\$29.95
44 Additional Ready-to-use fonts	
Texture & Symbol Library Disk	\$29.95
Hundreds of pictures, symbols, textures	
ScreenPro Utility	\$59.00
The ultimate screen print/capture utility	

FREE Catalog available on request

From the Leader in Z-100 Graphics...



3620 Amazon Drive
New Port Richey, FL 34655



Order toll-free...
800-346-2152

Or in Florida... 813-376-9347
Unconditional money-back guarantee
if your not satisfied!

Know Your

abCEDs



Ed Schindler
3 Pacer Lane
Norristown, PA 19401

Command Editor (CED) is an auxiliary program for MS-DOS which enhances some common command entry functions at the MS-DOS prompt, as well as in certain other programs. The program allows command lines to be edited before they are "transmitted" to the operating system; allows previously issued command lines to be recalled, edited, and reissued; automatically supplies some runstring parameters; and lets you define short synonyms for common operations.

There are two especially nice things about CED. The first is that it works with MS-DOS so well and so naturally that you will wonder why Microsoft did not put similar functionality into MS-DOS. The other especially nice thing about CED is that it is available on the HUG Bulletin Board for free.

CED is a TSR (terminate and stay resident) program written by Chris Dunford way back in 1985. Once installed, it intercepts command input and services requests by MS-DOS or by other programs for what MS-DOS calls Buffered Keyboard Input. MS-DOS programmers will know this as function call 0AH, and others may recognize it if the F3 key recalls the line last typed. CED works with input into MS-DOS or other programs that use Buffered Keyboard Input, and knows enough to differentiate between the two, so that you would, for example, never recall MS-DOS commands when inside another program.

The following sections will outline the functions of CED. The last section illustrates the use of some CED features to build a basic menu system to automate the use of the assorted application programs that accumulate on many home systems. This article is not intended to be complete documentation of the CED program, but rather a brief introduction to it. The manual which comes with the program is complete and clear, and is certainly recommended browsing.

Command Editing

MS-DOS provides the backspace key for your convenience. If you notice a mistake on the beginning of a line, all you have to do is backspace to that point, erase the latter part of the line, retype the incorrect portion, and then retype all of the command you had to delete to reach the error. If you have already entered a bad command, MS-DOS has several other functions to recall the last command character by character, correcting as you go (if you can keep track of where you are, that is) until you rebuild a corrected command.

Picture now the ability to take that same command line at the beginning of which you noticed the error, move the cursor to the point of the error, correct it by backspacing, inserting characters or typing over what's already there, and then hit Return to enter the corrected line. That's what the command editing function of CED does for you.

With CED installed, command editing is accomplished by a combination of moving the cursor and erasing, inserting or changing characters. The keys which control cursor movement are left arrow, right arrow (with obvious functions), the Home and End keys to move to the beginning and end of the line, and control/left arrow and control/right arrow, which move the cursor word by word, instead of character by character.

Once the cursor is positioned at the point at which the correction is to be made, Insert can be used to toggle the Overstrike/Insert function. In Overstrike mode (denoted by an underline-style cursor), a typed character replaces the one at the cursor position. In Insert mode (half-block-style cursor), typed characters are inserted before the character at the cursor position.

Further corrections are made using the delete character functions. The Delete key removes the character at the cur-

sor position, while the Backspace key removes the character to the left of the cursor. Pressing control/End deletes all characters from the cursor position to the end of the line. The Escape key erases the entire line. Control/Home is supposed to delete characters from the beginning of the line to the cursor position, but does not function on a Z-148 in version 1.0C of CED.

Command Recall

Aside from operating horizontally on the current command line, CED has a "vertical" mode, using the up arrow and down arrow, which allows you to recall previously issued commands for editing or re-entry. This is similar in function to the MS-DOS F3 key, which redisplay the last command. However, unlike MS-DOS, CED allows continued operation of the recall command so you can scroll through previous commands, and, of course, allows changes to be made in the selected command most conveniently.

CED saves commands in what is termed a stack. This is simply a list (of fixed size) of previously entered commands. When the list fills, the oldest command drops off the end to make room for the most recent one. Actually, there are two stacks in CED: one for MS-DOS commands, and one for application program commands. CED can sense the context of the entries, and thus keep you from recalling old MS-DOS commands into your text adventure game, where they probably wouldn't be understood. It also helps keep you from telling MS-DOS something like "Drink champagne from sneaker," which MS-DOS probably wouldn't understand.

There may be some commands that you don't want stacked, however. Format and delete might be likely candidates. CED lets you tell it what commands you never want stacked. Furthermore, you can flag command lines with a special control

Contents of menu text file HOME.TXT:

```
Home Menu
MISC      - go to miscellaneous functions submenu
FC        - run First Choice
ACCOUNT co - run PC-GL accounting for co(mpany)
```

Contents of menu synonym definition file HOME.CED:

```
syn menu  cd c:\ ^cls ^type home.txt
syn misc  cd -fc:\misc.ced ^menu
syn fc    cd c:\choice ^first ^menu
syn account cd c:\acct ^pc-gl \acct\11 ^cd c:\
```

Figure 1
Files Defining Home Menu

```
ced -fhome.ced
cls
type home.txt
```

Figure 3
Lines Required in AUTOEXEC.BAT
to Initiate Menu

Contents of submenu text file MISC.TXT:

```
Miscellaneous Functions Submenu
LIST file - list a text file page by page
BU mask  - backup files to floppy
MODEM    - set com port for modem use
PRINTER  - set com port for printer use
RETURN   - return to the previous menu
```

Contents of submenu synonym definition file MISC.CED:

```
syn menu  cd c:\ ^cls ^type misc.txt
syn return cd c:\ ^ced clear syn from menu ^menu
syn list  more <11 ^pause ^menu
syn bu    backup %1 a: /v/g ^menu
syn modem mode com1:1200,n,8,1 ^mode lpt1: ^menu
syn printer mode com1:9600,e,7,1,P ^mode lpt1=com1 ^menu
```

Figure 2
Files Defining Submenu MISC

character to exclude them from the stack on a one-by-one basis. (As with all CED control characters, you can choose anything convenient, if you don't like the default.)

Command Conveniences

Parameter Recall. CED has a limited parsing ability that lets it pick out the parameter(s) from a command line and automatically insert it on the end of subsequent, selected, commands. For example, in the program development edit, compile, link, and run cycle, the program name can be automatically entered on each line, simplifying the overall process. You just need to tell CED the specific commands with which parameter recall is to work.

Synonym Definitions. You can create shortened versions of commonly used commands with CED. One simple example might be the use of the letter "d" for "dir." You can also use synonyms for relatively complex commands that you use often (see the definitions for "modem" and "printer" in Figure 2). CED substitutes the definition for the synonym before sending the command line along. To be recognized by CED, a synonym must be the first word on a command line. CED keeps separate lists of synonyms for MS-DOS and application programs.

Parameter Substitution. After a synonym substitution, the remainder of the line is unchanged, effectively appending the original parameters to the new command line. In an additional feature analogous to that found in MS-DOS command file processing, synonym parameters can be substituted anywhere they are needed into the synonym definition. This lets you, for example, specify a file name in a word processor run string embedded in a chain of commands defining a synonym. A simple example is found in the definition of "list" in Figure 2.

Chaining. CED has defined a "chain" character (which defaults to the caret,

^^) that allows you to string commands along on a single line. The program then feeds them to MS-DOS one at a time, as MS-DOS asks for them. This is an especially nice complement to the I/O redirection and piping features of MS-DOS, and makes it possible to have much more flexibility in a single-line command entry. Synonyms become especially powerful when used with chained commands; they then have the power to replace batch command files (.BAT files). You could easily define a synonym to change the default directory to an application directory, run the application program, and then change the directory back to your home directory (see the definition of "fc" in Figure 1). Since these definitions are stored in memory, they run much, much faster than batch files. They are more flexible, too, as synonym definitions can contain other synonyms (to a limited depth, though).

Installing Resident Commands. CED allows MS-DOS programmers to install any number of "resident" commands into the operating system. The advantage is that the programs need not reside on an available disk to be accessed from the MS-DOS prompt, a feature especially attractive to users of diskette-based systems. Because the program must call CED to effect its own installation, it is only possible to install programs written for this purpose, unless the source code is available and can be modified. The new program simply makes an interrupt function call (which CED traps) and tells CED its name and where to find it. It then exits, staying resident in memory, so that CED can run it whenever its name is recognized in an input line.

Operational Features

Size of Buffers. When CED is activated, you can specify the size of all six of the lists CED keeps for its operation. These lists are for 1) the MS-DOS command stack, 2) the application program

command stack, 3) synonym definitions, 4) commands not to be stacked, 5) commands on which parameter recall is to operate, and 6) programs installed as resident commands. The minimum size of each list is 128 bytes, and the maximum size is more than big enough. This lets you tailor CED for your use and possibly conserve memory, as reducing the buffer sizes reduces the resident size of CED.

Status Display. You can display the status of CED at any time by pressing a single key. CED shows the loaded size of each of the six buffers, how much of each buffer is free, and lists the definitions of all synonyms and the contents of the parameter recall and ignore lists. You can change the key which activates this display, which defaults to control/T.

Clearing Unneeded Items. CED recognizes commands to clear any of its buffers completely, or to clear specific synonyms, parameter recall commands or ignore commands. This lets you recover wasted space which might otherwise only be available through rebooting and reloading CED (or worse — installing another copy).

Multiple Synonym Definitions. It is possible to define a synonym more than once. CED will always use the latest definition, and will clear only the latest one when a specific clear command is entered.

CED Command Files. Complex or routine CED commands can be placed in a file for a sort of batch execution mode, either at installation time or at any other time. Specifying the option and command file name tells CED to accept commands from the file, and is most useful for initial configuration of the system at boot time. All of your favorite synonyms can be installed and ready automatically.

Example Application

Using the features of CED described above, it is possible to put together a basic menu system for managing access to your system and its programs. While the

menu system described below was developed mainly to illustrate the power of CED, it is as well a useful, working mechanism. It works by defining synonyms as needed for the context of the current menu, defining new synonyms for sub-menus, and clearing sub-menu synonyms when returning to a parent menu. One advantage of a system implemented in this manner is that you always have the full range of normal MS-DOS commands available, since all input is at the MS-DOS prompt. This menu scheme works best on a hard disk system because of access to the menu definition files; however, these files are not large and could fit on application diskettes or a small RAM disk, if needed.

Each menu has two files associated with it. The first file contains the text of the menu (including the optional commands to clear the screen and draw a box on the screen, if you want to get fancy). The text file should include at least a list of the synonyms which are defined under this menu and a description of what each does. The second file is a CED command file containing all the synonym definitions is set up to have data from various companies in subdirectories each named after a specific company. This allows the company name to appear in the command line and be placed in the PC-GL command line via parameter substitution.

The definition of "misc" illustrates the general form of a submenu call. Its function is to call CED to load the synonyms for the submenu (including a redefinition of the synonym "menu"), and then to issue the command "menu". Since CED will always use the newest synonym definition, the submenu text file will be displayed, and not that of the parent menu.

The files for the submenu are shown in Figure 2. The text file, as before, simply contains a list of commands defined for that menu. The CED command file defining the synonyms includes lines for "menu," four utility commands (note the use of chaining, the pause statement and parameter substitution), and the additional command "return." All submenus include the "return" definition, which uses CED's Clear Synonym facility to delete all synonyms defined for the current menu, and issues another "menu" command. The "menu" command now accesses the previous definition (as the newest one has just been cleared) and displays the parent menu.

This submenu scheme can be continued as many levels deep as desired, and each menu can consist of further submenu calls or action-oriented commands. Note that in any submenu, all commands defined for all menus above it are available at the prompt (unless they are super-

ceded by a new definition). This will not cause a problem if attention is devoted to working directories and paths in the synonym definitions.

The final piece of the menu system is shown in Figure 3: the lines needed in the AUTOEXEC.BAT file to initiate the system. The first command installs CED and loads the main menu synonyms. This is the place to customize the buffer sizes, if you need to. The next commands are the commands needed to display the main menu. Ideally, a "menu" command would be used, but CED doesn't see input from batch files (it goes straight to MS-DOS) and so wouldn't be able to translate it.

Conclusions

CED is one of those programs that is easy to take for granted because it functions unobtrusively, providing functionality which should have been built into MS-DOS in the first place. It is a program with a specific function that doesn't try to be all things to all people. Given how long CED has been around and how nicely it works, it's hard to imagine that there is a PC system out there that doesn't have a copy installed. *

We'd Like to Send Your Computer a Christmas Card!



Single RS232 Serial Port Card	\$49
Dual Serial Port Card	\$85
Parallel Port Card	\$49
Game Port Card	\$49
1 Serial/1 Parallel Card	\$95
Serial/Parallel/Game Card	\$119
300/1200 Baud Modem Card	\$109
300/1200/2400 Baud Modem	\$149
Floppy Disk Controller Card	\$75
Hard Disk Controller Card	\$95
Floppy/Hard Disk Card (AT)	\$199
Floppy/Hard Disk Card (XT)	\$199
Mono Graph/Parallel Card	\$139
CGA Color Video Card	\$139
EGA Color Video Card	\$195
VGA Color Video Card	\$395
Add 3% shipping & handling on orders under \$500, 2% if total is over \$500.	

Call for Current Prices on RAM Cards!

Merry Christmas!

These boards have all been fully tested by our product evaluation staff and are guaranteed to work in Zenith Data Systems computers. All boards listed come with a 90-day replacement guarantee from First Capitol Computer *in addition to manufacturer's warranties!*

Prices are subject to change without notice, so please call to confirm current pricing when placing your order.

**First
Capitol
Computer**

Order Line: 1-800-TO-BUY-IT
Other Calls: 1-800-447-8697

How to Get the Most from a Zenith Laptop Computer

Joseph Katz
103 South Edisto Avenue
Columbia, SC 29205

*Part 6. If you can't make a case for maximum protection,
look through this range of products to find your bag.*

When you don't really need maximum protection for your Zenith laptop computer, think about carrying it in a bag instead. A bag should cost you less money to buy and less effort to carry than a case, especially if you follow the approach I suggested in Part 5 of this series. Now let's look at the application of that approach to commuting with a Zenith laptop computer. The goal is to reduce your load of computer paraphernalia to only the laptop computer itself.

First see how much of that equipment can be left wherever you use it. If you can postpone all printing jobs until you reach your workplace — whether it's your office or your home — there's no point in lugging a printer around with you. Then there's also no point in carrying printer-related stuff, such as fanfold paper, a spare ribbon, or even a printer cable. Not only have you reduced your burden greatly, but you've also done away with the need for a bag built to carry such things. You've saved yourself about \$50 that way: it's the extra cost you can expect to spend in today's market for a bag designed to snuggle a portable printer safely.

Next, see if you really can or will use your laptop computer during the traveling part of your trip. Even if you can and do, think about whether your computing time en route can be supported exclusively by

the power of a full charge on the internal battery in your Zenith laptop computer. If so, there's no point in carrying either the AC adapter or a spare battery pack. Invest in a second AC adapter if you need one at home and the other at work. Then be especially careful to follow a frequent deep discharge regimen for your NiCad battery pack to avoid the memory effect I often warn against in this series. Although most good bags for Zenith laptop computers have room for their AC adapters and even for a spare NiCad battery pack, such as is used in the Z-171, Z-181, and Z-183, you'll need to look more carefully for a bag with the capacity for the bulkier battery slice used in the SupersPort/286. But why tote these things if you aren't going to use them while you're actually in transit?

You might also think about whether you really need to dedicate a bag exclusively to your laptop computer. If so, you might want to consider either Traveling Software's Ultimate Laptop Attache, which provides good protection at a moderate price, or Beseler's Model LEC-2 in its LeMans series, which provides additional features at a higher price. If you don't need a dedicated bag — because you carry "stuff" that includes, but is not limited to, your Zenith laptop computer and because it is not exposed to great danger during your trip — you might want

to go in an entirely different direction and look at a Lands' End Attache instead. There are at least three to consider in a range of prices from \$39 through \$75.

Lands' End Attaches Lighthouse, Original, and Deluxe

Here's a case history, if you'll forgive an irresistible pun. My wife Janet is a graphics designer and consultant whose office is only a short drive from home. But her office is often only a base she leaves to consult with clients and suppliers elsewhere. She regularly carries heaps of paper and other media, and occasionally carries either a ZP-150 or a Z-183. What she needs is a stylish carryall that safely holds all her stuff without making her look or feel like a pack animal. It needs to look good because part of her business is image. It must offer good protection from the elements, because she's in and out of her car a lot in all kinds of weather. It has to be expandable, because she must carry various kinds and quantities of goods. It

Copyright © 1988, by Joseph Katz. All Rights Reserved. "How to Get the Most from a Zenith Laptop Computer" is a trademark of Joseph Katz. Address all correspondence to Dr. Joseph Katz, 103 South Edisto Avenue, Columbia SC 29205.

must have minimum tare, because she has no need to pay for great protection with the weight it imposes. In fact, protection against bumps and smashes is no great concern to her because Janet's environment is protected and her eye is watchful. Her choice of laptop computer carrier usually is the same bag she takes with her as a general portfolio — a soft-sided Attache from Lands' End Direct Merchants.

Just to show you that price is less important than functionality when it comes to choosing a carrier for your Zenith laptop computer, \$39 gets you Lands' End's Lighthouse Attache (Catalog 7705126) in nylon trimmed with canvas outside and reinforced with vinyl inside. It's a real lightweight — the lightest bag I've seen so far that is suitable for one of Zenith's laptop computers. Fifty cents more buys the Original Attache (Catalog 4402120), which is in 18 oz. cotton canvas throughout, and therefore, substantially heavier than the Lighthouse Attache. An 8-inch deep pocket extends the width of each bag on one side: you can stuff manuals and other papers in it. Sewn to it are three 6 X 8 inch pockets: put diskettes, cables, and a portable modem in them. Sewn to two of them are sets of three narrow pockets in which you can slip screwdrivers, writing instruments, and other tubular matter. There's also a keysnap on a strap at that side: it grips the ring on a ZP-150 computer for added security. Outside is another 8-inch deep pocket that extends along the bag's width. This one has a small velcro fastener to keep it closed. A nice distinction of these two bags is that they're among only three I've seen with loops outside for holding a furled umbrella <197>, a real plus to Janet and anyone else in her situation.

The only other bag I've seen with an umbrella loop is Lands' End's Deluxe Attache (Catalog 7294123), which costs \$79 and gives you leather trim and corner reinforcements, as well as washable nylon lining inside. You get the same umbrella holder, keysnap, and interior pockets as on the two less expensive bags, but the exterior pocket is just six inches wide because the handle on that side is longer for extra support and everything inside, except the keysnap is nylon. Of course you get your choice from a wide range of colors, and Lands' End even will monogram the Attache for you free if you ask when you order. And of course, there are canvas web handles and a canvas web shoulder strap on all the Lands' End attaches. The Deluxe Attache has a leather pad on the shoulder strap. You'll want the shoulder strap if you succumb to the temptation of cramming the Lands' End Attache chock full.

You may cram a Lands' End Attache chock full without worry. All three bags measure 17-1/2 inches wide, 13 inches

high, and 5 inches deep — enough to hold any Zenith laptop computer except the Z-171, and enough even to hold the battery slice for the SupersPort/286. But because the Lands' End Attaches are quintessential bags — completely flexible, with no padding or stiffeners of any kind — those measurements are useful mainly as a guide. The bag will flex to accommodate any reasonable stuffing instinct. It's constructed to be stuffed. All the seams are reinforced with cording, turned inside and bound. They shouldn't split and dump your computer unless you're really outrageous about packing stuff in. The zippers are heavy-duty brutes fastened inside the opening with double stitching. They shouldn't easily pull out. The loops and rings are metal instead of the plastic customarily found even on dedicated laptop computer bags. They won't suddenly give way, as has been known to happen on inferior bags, even some sold especially for carrying laptop computers.

Because these are not dedicated computer bags, but carryalls from a well-known source, and marked with its distinctive cachet, they offer the added advantage of an unusual kind of security. If you don't tell anyone or show anyone, no one will know you're carrying a few thousand dollars worth of laptop computer inside.

Traveling Software's Ultimate Attache

Traveling Software's Ultimate Attache is a good example of today's moderately-priced, dedicated laptop computer bag. It costs \$69.50 and its outershell is made of gray Cordura with a black nylon shoulder strap, webbing, lining, and trim. The strap has a generous 2-inch depth, so it's less likely to cut into your shoulder than a narrower strap, and it's removable so you can carry this outfit by its pair of nylon web handles, just as you would an attache case. There's a keeper stitched to one handle. That keeper wraps around the other handle and fastens to itself with a velcro strip, holding the handles together and serving to pad them from cutting into your hand.

Although it's sold for various laptop computers, the internal dimensions of the Ultimate Laptop Attache seem designed for a ZP-150, Z-180, Z-183, SupersPort, or SupersPort/286: 12-3/4 inches high, 15-1/2 inches wide, and 3-3/4 inches deep. For extra security, there's an internal safety belt that can fasten around the computer. It might not be much of a nuisance to unfasten should you decide to do some work with the Z-181 or while in transit. But you can't do that with one of Zenith's sPort models: although the computer itself fits inside this bag with room to spare, there's not enough room to hold the battery slice.

Inside, the Ultimate Laptop Attache

is filled with slim pockets. Two deep pockets extend the width of the bag on the top and bottom sides. They'll hold even legal-size file folders. Stitched to the outside of the top pocket are three 4-1/2 X 7 inch expanding pockets in which you can stash floppy diskettes, a portable modem, and even some non-computer items as well, such as a microcassette recorder. All the topside pockets are provided with Velcro fasteners, as is that internal safety belt. Just keep in mind that the Ultimate Laptop Attache won't allow you to transport both your computer and a heap of other stuff besides. Think slim. If you want to carry a slim file of papers in a more accessible place, there's an 11-inch deep pocket on the outside of the top side. That pocket has a zipper closure. So does the main part of the bag: it has one of those two-talon zippers that pull around towards the middle.

Of special importance to daily commuters is the canny distribution of padding. There's a layer of padding all around, and a stiffener in addition to the padding around the width. That's where you'll incur most bumps as you wend your way deftly through the mobs in airports and depots, and that's where your computer will inevitably bang against the armrest of almost every seat as you move through the vehicle.

What distinguishes this bag from other mid-priced laptop computer bags of a similar kind is its source. Traveling Software probably is the oldest company specializing in products for laptop computers.

Beseler's LeMans Model LEC-2

If you're a camera buff, you've probably encountered Beseler's name in connection with bags for camera equipment, accessories, and supplies. They're a major manufacturer. Now Beseler is moving into bags for laptop computers — a sure sign, by the way, that this kind of computer is no longer a passing fancy or odd phenomenon. Those are ways it used to be considered. Now it's a stable market, with its own characteristics that can be defined and predicted. That's interesting, I think, especially in its implications that you can expect a boom in laptop computer products and will need to evaluate them carefully. Beseler's initial entry into this market is so new that my bag still has markings in tailor's chalk, which suggest it's a sample. I understand that the price will be \$99.

My bag is the Model LEC-2 in the LeMans series of Beseler's Diamond brand. Its outershell is made of tan nylon coated with urethane for resistance to weather and soil. There is a lining of RipStop nylon and a layer of protective padding on the sides and top. Protection on the bottom comes from an insert that's really a two-inch-thick sandwich made from three different kinds of foam and a

Continued on Page 42

A Shot In The Arm

For The H/Z-100

Richard (Rich) L. Mueller, Ph.D.

11890-65th Avenue N.
Maple Grove, MN 55369

Is the H/Z-100 Machines Dying?

For the past year or so, especially in the last 6 months, I have noticed that the number of H/Z-100 machines (all types) for sale has been on the rise. "Want Ads" in Buss and H-Scoop, as well as vendor ads in REMark and Sextant for the H/Z-100 machines and related software show this to be true. The H/Z-100 machines are selling at an all-time low price and software is almost free. Vendors seem to be having final closeout sales on all H/Z-100 related equipment and software.

Does this mean that the H/Z-100 is a dead machine? Are all the H/Z-100 "Huggies" abandoning the "best machine" ever built? I, for one, am not abandoning the ship and I hope others are not as well. Oh, I know that one really needs to look at '286 and '386 class machines to stay current and that's OK, but why get rid of an excellent machine that, in my opinion, has a lot of life yet.

Back in late 1982, I felt that it was time that I got a microcomputer after working in the computer field at that time for over 20 years. I looked at and evaluated three machines, one which was the H/Z-100. After considering all the features that the H/Z-100 had (such as having two microprocessor chips, an 8085 for CP/M and an 8088 for MS-DOS and CP/M-86, support for both 5.25" and 8" floppy drives, and video that was better than the competition), as well as the add-on features available (such as more video memory, more RAM to 768K, additional serial and parallel ports, color support, etc.), I felt, and still do, that the H/Z-100 was the best machine.

Instead of getting a Z-100 in late 1982, I decided to get an H-100 in early 1983 when they became available and

I'm glad I waited. I have always enjoyed putting together Heath Company electronic kits, and the H-100 was no exception. Although I also have a Z-160, I still use my H-100 more. During the past couple of years, I have been upgrading my H-100 with new hardware, such as a new motherboard to support 768K, FBE ZCLK, V20 chip replacement for the 8088 chip, 8087 addition, have purchased the 8 MHz upgrade kit from First Capitol Computer, added two 8" drives, Z-204 Multiport Input/Output board, and planning to add a Z-205 Memory Board product from FBE to add a RAM Disk and hope to get a hard disk as well, etc. There is no doubt in my mind that I want to keep my H-100 for a long time to come.

At the same time that I was upgrading the hardware, I was adding to my collection of software for the H/Z-100 which includes MS-DOS, CP/M, and CP/M-86 software from whatever sources that I could find. HUG conferences are always a good source, the HUG software library is another source, as well as some of the Bulletin Boards, such as the CHUG BBS. Software is the reason that I decided to write this article. Before getting to the main part of my article, I would like to ask the Heath/Zenith User Community for leads in getting additional CP/M software for my H-100 especially CP/M-86 software, that is where I am a little thin. I would appreciate help in this area.

Public Domain Software

As I mentioned above, the HUG International Conferences have always been a good source for H/Z-100 software and the last one in August 1987 was no exception. On one of the tables outside of the vendor area at the last HUG confer-

ence (VI), I picked up a sheet from Gernware Enterprises of Utica, New York, offering some H/Z-100 Public Domain software.

The software comes from the INFO-HZ100 Public Domain Library. INFO-HZ100 is a forum on several of the large computer networks such as ARPAnet, Bitnet, Milnet, etc. I was not aware that such a forum existed, but because of it, we have a large number of software programs. The Library consists of over 750 files saved in ARChived compressed format. Because of the compression format, all the files fit on eleven (11) 360K floppy disks. Before I forget, I want to mention that many of the programs come with the source code, especially those written in C.

The software programs have been screened and are declared to be "public domain". They may be copied freely, but not sold. They are available as is. All the programs are designed to run with MS-DOS V2.0 or higher. BASIC programs are in ASCII format to allow the user to convert the programs to whatever BASIC is being used on his/her H/Z-100.

If all this software is available on a computer network, how does Gernware fit in? Well, they are providing a service to the Heath/Zenith User Community by offering all eleven disks for \$25 which is for the cost of the disks, their time in duplicating the disks, and shipping/handling. This is really a small price to pay for a large number of programs.

When I saw that, I jumped at the opportunity to get many programs at such a small price. To me, this was a "Shot-in-the-Arm" for the H/Z-100. It gave the H/Z-100 user a chance to get a lot of new software at a cheap price. Granted, some software may have a request for a dona-

tion if the software is put to use, but even with that, the Library is still a super deal. There is no way that anyone would ever put all the files/programs to use because of some duplication, lack of proper hardware, such as a color monitor or available RAM, some programs require a specific programming language, such as a specific C compiler or Pascal compiler, or some of the programs just don't fit in with the user's needs. However, even if just a few made sense and proved to be useful and productive for an individual, that sure would be well worth the \$25, plus any small donations if any were requested. Little-by-little I have been going through each of the files to see what I can and cannot use. There are some very interesting programs in this Library.

What I would like to do now is to highlight some of the files/programs that are in each of the eleven disks in the Library so that you can judge for yourself whether this Library is an excellent buy or not. Most of the disks contain more files/programs than what I mention below.

Disk 1:

Earlier, I mentioned that most of the files in the Library were in compressed format. Disk 1 contains PKARC3.5 and ARC500 to extract files/programs from the compressed formats. In fact, PKARC3.5 itself is the only program that is not in compressed format. It is needed to uncompress all the others. In addition to the file Archive/Extract programs, there is a VT-52 terminal emulator program, a modem program, and a number of small assembly language utility programs that do such tasks as change file attributes, set foreground/background colors, clear screen, update time information in a file, etc.

Disk 2:

This disk contains such programs as ZANSI (the ANSI Console Device Driver for the H/Z-100 which is similar to the ANSI Device Drive, ANSI.SYS, for the PC compatibles), another file compression/uncompression program, a command line editor (called CED100 that supports improved command line editing, recall of previously issued commands, command synonyms, filename completion, etc.), a program to manage memory resident programs, a full screen utility to display, copy, and delete files in a directory, a collection of font files that may be used as ALTCHAR.SYS or with the FONT command, and even a space war game.

Disk 3:

At the top of the list, this disk contains a full-screen interactive dis-assembler for the Intel 8086 through the 80386 chips, as well as for the NEC V20 and V30 chips. Next there is a set of disk cataloging tools similar to the CP/M NEWCAT pro-

gram that help the user add files to a catalog, delete files, add/change disk volume labels, label DOS V1 disks, etc. There also is a collection of GW-BASIC programs, such as mailing list and label printer program, typing tutor program, terminal emulator and file transfer program, translator for 8080 source code to 8086 code, etc.

Disk 4:

The main program on this disk is a full-screen editor called TVX. Both binaries and C source code are included. There also is a number of Turbo Pascal (software language from Borland International) graphic support programs for the H/Z-100.

Disk 5:

Contains another collection of GW-BASIC programs that are primarily graphic in nature. There also is a set of programs to plot on Tektronix terminals, and a couple of files that contain H/Z-100 newsletters and related information.

Disk 6:

This disk contains a set of BASIC programs for producing presentation displays on an H/Z-100 and a program to create slide shows using the displays. Also, there is a calculator program (actually two versions; one for the H/Z-100 and the other for a PC compatible) containing multiple registers, function buffers, and scientific functions.

Disk 7:

The largest program on this disk is a dungeon and dragons style game that includes another copy of ZANSI, the ANSI console device driver. There also is a collection of miscellaneous programs that display high-resolution images on a H/Z-100 (interlace mode), some scripts for HyperAccess (an excellent communications package from Hilgraeve, Inc.), etc.

Disk 8:

This disk contains quite a collection of programs. First, there is a set of GW-BASIC games; well over 30 of them. Some of the games are Solitaire, Pacman, Keno, Star Trek game, Othello, Hangman, and the list goes on and on.

Next, there is the desktop utility package called "Right-Hand Man" (RHM), which is similar in function to the popular PC utility Sidekick from Borland International and PERKS from Barry Watzman for both the H/Z-100, as well as PC compatibles. RHM provides such functions as an appointment calendar, notebook, names and address file, calculator, ASCII table, a file editor, card index, etc. Part of the utility remains memory resident, while the rest is disk resident.

This disk also contains a set of a few C language programs, such as a banner

program, a calculator, a string search utility similar to the DOS command, FIND, but with more options, a utility that will count lines, words, and characters in a file, etc...

Disk 9:

Just about the entire disk is taken up by JOVE, an interactive display-oriented editor with multiple window and multiple file editing capabilities which was written in C. What is on the disk is the source for over 40 C routines and other routines that make up JOVE. The user needs a hard disk system to compile and generate an executable version of JOVE for the H/Z-100. Also included on this disk is a set of Graph programs for both low and high resolution for graphing a tabulated function.

Disk 10:

This disk contains a text file with pre-1986 ZDS Bulletin Board information, an RBBS program written in BASIC for Hayes compatible modems, a COMMAND.COM replacement that implements Unix-like shell commands, such as: is, mv, fgrep, rm, chdir, chmod, etc., and a few GW-BASIC draw files.

Disk 11:

Contains a collection of miscellaneous programs, such as a calendar program written in Turbo Pascal, another Unix-like shell program, the game of Life, several files containing printer pictures, package for developing MS-DOS device drivers using Lattice C, MAKE which simplifies the building of programs by allowing the build sequence and dependencies to be specified in a file and only those modules that changed and those that depend on these to be rebuilt, etc.

Closing Comments

Back earlier, I said that if even just a few of these programs were found to be useful to you, it was well worth the \$25. Now that I listed and made a few comments on a number of the programs, maybe even one useful program is worth the \$25. Do you agree?

What is nice about this Library is that there is a variety of programs and languages. For example, there are BASIC, Turbo Pascal, C, and Assembly Language programs; a variety and large number of games; a variety of utilities; applications like Right-Hand Man, a mailing list program, and an interest amortization program; communication and terminal emulation programs; etc. What more can one ask for?

In my opinion, this certainly is a "shot-in-the-arm" for the H/Z-100 by providing a large number of MS-DOS software programs for almost nothing. Now that I added a "bunch" of new MS-DOS programs, I only hope that I can someday (soon) find a similar library of CP/M and CP/M-86 programs for the H/Z-100.

Again, if anyone has information on such a library or libraries, I sure would appreciate the information.

Long live the H/Z-100, the best machine ever built!!!!

Reference Information:

Gernware Enterprises
2803 Oneida Street
Utica, NY 13501

First Capitol Computer

16 Algana Drive
St. Peters, MO 63376-3930

Borland International
4585 Scotts Valley Drive
Scotts Valley, CA 95066

Capital Heath/Zenith Users' Group
(CHUG)
P.O. Box 6856
Alexandria, VA 22306

FBE Research Company, Inc.

P.O. Box 68234
Seattle, WA 98168

Hilgraeve, Inc.
P.O. Box 941
Monroe, MI 48161

Bary Watzman
(in process of moving at the
time of writing this article)

*Ed's Note: The entire Gernware Library is available on HUGPBBS (616) 982-3596. **

Continued from Page 8

One reason why not is that you're playing with fire and will be burned by diskette failures sooner or later. You're banking more than you might care to lose on the ability of the diskette to maintain its integrity under conditions of use it was not made or certified to handle. Assume for a moment that diskette manufacturers are playing some sort of game with the standards in order to maximize their profits, and therefore, that there's only one real difference between diskettes certified lower and those certified higher except for their prices. Assume it. Are you sure enough about your assumption to risk the future of your work on it? I'm not.

Another reason to avoid diddling around in this particular way is that you probably are producing diskettes that can be read only on your machines and those similar to it. When Janet brought home

such a diddled diskette containing some files she had scanned for my book, the 1.44MB 3.5" drive in my Z-386 just couldn't read the diskette. It was useless to anyone who didn't own the same kind of machine on which it was produced. No, the "trick" isn't limited to IBM's PS/2. I gather — although I have not yet checked for myself — that you can do the same thing on a Tandy 3000 micro-computer. The reason I believe you can do it is that I've heard of a few poor souls who are running around with diskettes that seem to have been created that way. Some of these folks have even had their own computers, floppy disk controllers, and diskette drives checked by service shops to see what could be wrong with them. Of course, there was nothing wrong. The problem is that they cannot read non-standard diskettes. And that — by definition — is exactly the product of

all those ingenious ways to pack more on a diskette than is called for in the standards they should meet. It makes sense, doesn't it: when you set out to create a non-standard diskette format and succeed, what you succeed in producing is a non-standard diskette. The result is that you have a diskette readable by only a few machines, and maybe only the one that produced the diskette. The very idea sends shivers down my spine.

Call these rules "obsessions" if you will, and I won't disagree. I'll probably be too busy working with data files that have come along with me through every micro-computer I've ever owned or used. What I've explained are a few rules that have helped me thrive and might be of some use to you if your goal is the same.

See you later.

*

Circle #136 on Reader Service Card

FORM FILL-R

NOW WITH FORM EDIT-R

FORM FILL-R: Fills in pre-printed or user-created forms up to 132 columns by 132 lines.

FORM EDIT-R: Creates forms; handles field placement and attributes; cut & paste rectangular regions of text; draws lines and boxes using line-drawing characters, and automatically joins lines together for simple form layout; many other features.

CREATE: Custom forms, charts, work sheets; your data entry screen can be either the printed form or a data entry screen which you design.

PRINT: Forms directly, or save in disk files.

SAVE: and Recall to change previously filled forms.

EXAMPLES: Several are included.

REQUIRES: IBM Compatible or Z-100 Computer; MS-DOS 2.0 or greater.

\$49.95 Postpaid

LINDLEY SYSTEMS 4257 Berwick Place
Woodbridge, VA 22192-5119 (703)590-8890

PowerStation



If you have PCs and VAXes, you NEED these products!

For connecting your PC to a VAX, the PowerStation provides the ultimate solution: a VT200 layout keyboard bundled with sophisticated ZSTEM terminal emulation software.

ZSTEM software includes DEC VT240, VT220, VT100, IBM 3101, TEK 4014 and DG D400 terminal emulation.

"Its performance is as perfect as an emulator can get . . ."

DIGITAL NEWS

For *true* terminal emulation call (800) 663-8702.

KEA Systems Ltd.

#412-2150 West Broadway
Vancouver, B.C. CANADA, V6K 4L9
Tel: 604-732-7411 Telex: 04-352848 VCR
FAX: 604-732-0715

Circle #179 on Reader Service Card



Easy Extra is easy WordStar

Ralph Seiler
377 E. Leslie Avenue
Salt Lake City, Utah 84115

The first word processor I ever used was WordStar 3.0 on a CP/M machine. I have changed computers several times since then, but I have never changed word processors. WordStar met all my needs and I purchased each major revision of it. Recently, my wife needed to learn to use a word processor and I began to teach her WordStar. She did not like it, the help was confusing and she would forget that “^” meant “ctrl” and not the character above the 6. After 2 hours she gave up. She wanted something easy to learn and use; she only uses the computer and word processor 2 hours a month and she didn’t want to invest a lot of time learning DOS, WordStar or any other word processor.

I went to the software stores in town looking for a word processor for a computer novice and I found Easy Extra. When I read on the box that the files were compatible with WordStar, I decided to give it a try.

Easy Extra is an easy-to-use word processor and mailing list program that is surprisingly powerful. The word processor is called Easy and the mailing list program is called EasyMail. The documentation is two small paper bound books. The first, Easy Guide, describes the word processor and the other, EasyMail Guide, describes the mailing list manager. Keyboard and menu maps for Easy and EasyMail are provided. Easy Extra comes on 6 non-copy protected disks.

Using Easy Extra

Easy Extra is operated by making choices from menus. You make a choice by using the arrow keys to move a highlight bar to your choice, and then pressing

return. Figure 1 shows the opening menu. From this menu, you select whether you will edit or print a file, work with a mailing list, configure Easy Extra or work with the files.

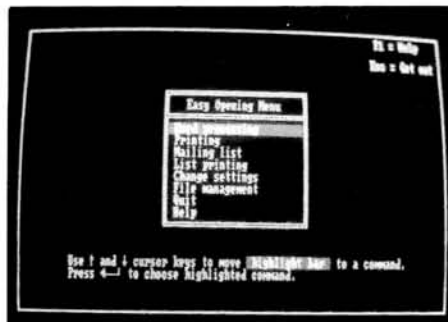


Figure 1
Easy Extra Opening Menu

There are only 3 keys to remember in Easy Extra and you don’t even really have to memorize them because that information is always on the screen somewhere. The three keys are Escape to ‘Get out’, and the function keys F1 for help, and F2 for the Editing or Data Entry Menu.

The Escape key is used to ‘Get out’ of whatever you were doing. Pressing Escape will cancel a command in progress, allow you to leave help, or return you to the previous menu. While you are editing, for example, pressing Escape returns you to the main menu (after asking if you want to save or abandon the file).

Pressing F1 brings help about what you are doing. For example, if you want help about a choice in a menu, press F1, and you will get detailed information describing the menu option that is highlighted. If you press F1 while typing, you will get general help about using menus,

the cursor movement keys, and special computer keys, such as the insert, delete, and control keys.

Pressing F2 while you are editing brings the Editing Menu. The Editing Menu contains the commands you will use most frequently while working on a document, for example, the commands for checking spelling and moving blocks of text. Pressing F2 while you are working with the mailing list brings the Data Entry Menu. The Data Entry Menu contains the commands you will use most frequently while working on the mailing list, for example, deleting records and searching for records.

Editing

Choosing ‘Word Processing’ from the opening menu brings a menu where you select what file to edit. The first option is ‘Edit Current File’; Easy remembers what the last file you edited was so if you want to edit that file just press return. If you want to edit another file, select the ‘Choose Another File’ option. This brings a new screen with a directory listing. You can select an existing file to edit by simply using the arrow keys to move to the file you want and pressing return, or you can type in the name of a new file you want to make.

Easy displays important information at the top of the screen while you are editing. The status line tells what page and line you are on, what print font you are using, and whether insert is on or off. The ruler line shows you what column the cursor is in, where the tabs are set, and where the right margin is. At the very top of the screen is a little reminder, F1=Help, F2=Menu and Esc=Get out.

On a color monitor, the normal text is green on a black background. There is no way to change the colors and use, for example, a blue background with white letters. Easy uses colors to show special print features on the monitor. Subscripts and superscripts are purple, boldface is yellow and underlined text is white. Dot commands are displayed in red on a white background.

Pressing F2 while editing brings the Editing Menu. There are 2 pages in the editing menu and F2 also flips back and forth between them. Figure 2 shows the first page of the editing menu. There are 20 choices in the editing menus, allowing you to move, copy, and delete text and restore deleted text. From the menu, you invoke the spell checker, change the line spacing, and do searches for a word or phrase. Figure 3 shows part of the second page of the Editing Menu and the help you can get by pressing F1 while in the menu.

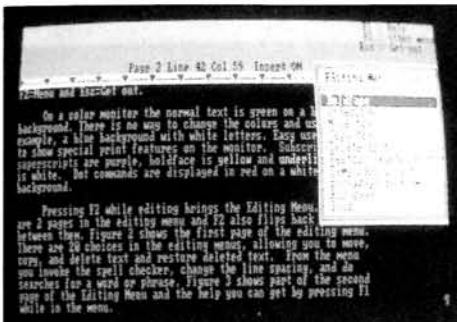


Figure 2
Easy Editing Menu

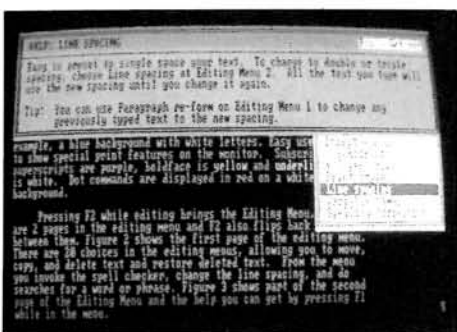


Figure 3
Easy Help

Moving, copying, and deleting blocks of text is simple and Easy always tells you what to do next if you forget. For example, if you want to move a paragraph, you press F2 to bring up the editing menu, and then select the option 'Move text'. Easy then tells you to move the cursor to the beginning of the block you want to move and press return. It then prompts you to move the cursor to the end of the block of text and press return again. At this point, the whole block of marked text is colored so you can see what will be moved. Easy then prompts you to move the cursor to where you want the text to

go and press return. It is as easy as that. Unfortunately, Easy has managed to make finding a phrase in the text confusing. I wanted to find the word "MicroPro" so I pressed F1 for the menu and then selected Find and Replace. I answered the question "Text to find?" and then it wanted me to answer the question:

Replace (+/-)?

Well that confused me so I remembered 'Press F1 for help'. Help told me to press '-' if I just wanted to just find a phrase and '+' if I wanted to find and replace. So I pressed '-' and it didn't do anything. It didn't write the '-' and it didn't do the search. Something is wrong I thought, so I pressed the '-' again. It still didn't do anything so I pressed the '+' and it drew a square root sign. I pressed the '-' again and the square root sign disappeared, but it didn't do the search.

After some experimenting, I finally figured it out. First, neither the documentation nor the Help tell you to press return after you press '-' or '+'. Second, the square root sign that Easy draws is really a checkmark! Third, there is what I think is a minor design problem in Easy. When you press the '-' it doesn't give you any feedback, it should write a minus sign (or whatever the opposite of a checkmark is!) However, if you press return after entering the minus sign, it will begin the search.

Once I figured out how to actually do a search, it was simple. Easy goes to the first place in the file it finds the phrase, highlights it, and waits. If you are where you want to be in the file, press escape, if not, press return and it will go to the next place the phrase occurs. Easy cannot do a reverse search.

If you want to do a find and replace, you press '+' and Easy draws a checkmark, then you press return. It then asks for the Replacement text and then asks "Automatic Replacement (+/-)?". If you press '-' and then return, it will ask you every time if you want to replace the phrase. If you press '+' and then return, the replacement will be made in the entire file without asking you.

Like everything else in Easy, you get to the spelling checker by making a selection from the menu. You must move the cursor to the beginning of the file if you want to check the spelling in the entire file, or if you want to check only one word, you must put the cursor on the beginning of the word. If Easy finds a word that isn't in the dictionary, it shows a suggestion and offers a menu of choices of what to do. If the word is correctly spelled, you can add it to your personal dictionary or tell Easy to go on to the next word. If the word is incorrectly spelled, you can fix the spelling manually or you can accept a suggested spelling and the computer will fix the spelling.

Mailing List

EasyMail allows you to make a data file of names and addresses so you can print personalized form letters. Neither my wife nor I use form letters, so my review of EasyMail is not as detailed as my review of the word processor. EasyMail is very much like the word processor, all choices are made through menus, Escape means Get out and F1 brings help.

Names, addresses, telephone numbers, and notes are entered into the data file by filling out a form on the screen. Pressing F2 brings the Data Entry Menu. From this menu, you can delete records from the data file, add new records or search for specific records. Figure 4 shows the Data Entry Form and the Data Entry Menu.

Combining information in the data file with text files to make personalized form letters is simple. First, you create the form letter using Easy. Then, using the Variable names option from the second page of the editing menu, you insert merge variables (blanks) into the form letter. When you print the file, EasyMail fills in the blanks with data from the mailing list file.

Form letters, mailing labels, rotary cards and envelopes are easy to print from the EasyMail Print Menu. When I tried to print some mailing labels, I ran into a problem. You have to tell EasyMail what kind of labels you want to print, so I selected 'Pick print format' from the EasyMail Print Menu. This brought a screen display that showed possible formats, including rotary cards, envelopes and mailing labels. The arrangements of mailing labels it allowed me to choose from were one, two, and four across the page. My mailing labels come three across the page and I could not use them with EasyMail.

Printers and printing

Selecting Printing from the opening menu brings the print menu, from which you can select which file to print, modify the print options and choose which printer to use. The print options menu allows you to choose the number of copies to print, the beginning and ending page numbers if you don't want to print the entire file, and whether to number the pages or not.

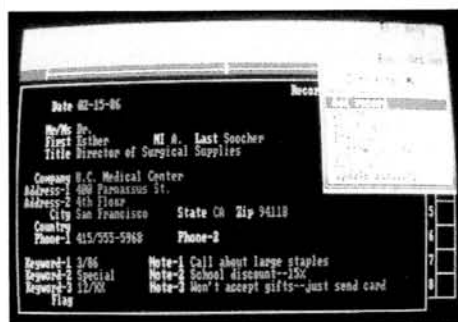


Figure 4
EasyMail Data Entry Form and Menu

You cannot edit one file and print another at the same time. While Easy is printing it displays a bar chart and shows you what percent of the file has been printed.

When you configure Easy, you select your printer(s) from a list with over 230 choices that covers everything from the Abati LQ-20 to the Xerox 2700 Laser. Easy can be configured to use 3 different printers. For example, you could connect a dot matrix printer and a color printer to different ports on the computer and select from the Print Menu which printer the file is sent to.

It is especially easy to take advantage of any fonts or special features that your printer may have. Easy knows what fonts are available to the printer(s) you told it are connected to the computer. In the customize printers menu option, you select the fonts and colors and 'print extras' you want to use for your printers. I customized Easy for my Epson MX-80 printer. Now by selecting font 2 from the editing menu, I can make text print compressed or by selecting font 3, I can make text print expanded. If I had a color printer, I could select the colors the text will print, and if I had a laser printer, such as the HP Laserjet, I could easily select print extras, such as math fonts and portrait and landscape modes.

Help

Help is just a keystroke away and the documentation isn't really needed. If the manual isn't handy, you can get detailed information from the 'Help Index'. The help index is selected from the Easy Main Menu or the EasyMail Main Menu. It is a 5 page list of over 60 topics in alphabetical order. Selecting a topic from the list will display information about that topic. For example, suppose you need to delete an old file from a floppy disk but you can't remember how. Selecting 'Delete File' from the Help Index will tell you that by going to the Opening Menu and choosing 'File Operations' you will get a menu that allows you to delete files.

Changing Settings

The Change Settings menu allows you to change the way some things work in Easy. From this menu, you can change the drive or directories where Easy looks for the dictionaries and text files, you can change page layouts, and you can setup and customize printers.

Changing the page layout is very simple. Easy draws a page on the screen and when it asks you what the top margin, for example, should be, it points on the page to what it is you are adjusting. Here is where you set the top and bottom margins, the right and left margins, the page length, and whether the pages are numbered or not.

You also can change some ways the

word processor works by using the Change Settings menu. Easy normally has a ragged right margin, however, you can make Easy have an even right margin. The find and replace in Easy is normally not case sensitive; that means if you search for 'cow', you will find cow, Cow and COW. You can make Easy case sensitive, so you won't find a COW if you search for a cow. Several other Easy features, such as the prompt level, the initial insert/overwrite setting, and whether Easy automatically reforms paragraphs are set in the Change Settings menu.

DOS operations

Easy makes it simple to do some DOS operations (copying, moving, renaming, and deleting files) without having to learn DOS. For example, my wife could not remember the format of the DOS command to change the name of a file. When she remembered the name of the command was Rename (and not Change-name), she forgot which came first, the name it is or the name you want it to be. She found it much easier to start Easy and use the File Operations menu whenever she had to work with DOS.

Tutor

The tutorial is excellent and is a well thought out program to teach a computer novice to use Easy. The tutorial only covers the word processor Easy and not the mailing list program EasyMail. The tutorial is divided into two parts; the first, 'Over-Easy', describes Easy in very general terms and takes about 10 minutes to complete. The second part, the 'Tutor', teaches the actual nuts and bolts of using Easy and takes about 45 minutes to complete.

I like tutorials for teaching novices programs; experts who try to teach novices to use programs have a characteristic fault — they become impatient. If a novice wants to think about what he is doing before pressing a key, guess what a human expert will do? He will reach across the keyboard, press the key and hurry along to the next step. The Easy tutorial does not do that, it is patient. If you make a mistake, it will correct you and tell you what you should do. Then, it will wait for you to do the step correctly and only then will it move along.

The first topic in the Tutor is a 'Keyboard Tour'. It assumes that you are basically familiar with a typewriter keyboard, but that you know nothing about a computer keyboard. It begins by explaining what the cursor is and then continues to show how all the other keys are used. The tutorial has you actually use all of the keys just as you would if you were in Easy. For example, you backspace to erase a word, you use the arrows to move the cursor around in text, you use the control key to change what the arrow keys do.

The second topic covers the 'Basics'.

The first part of the Basics has you type and print a letter, starting from DOS. The other parts show how to use the help files, how to delete blocks of text, and how to insert text.

I was very impressed with the lesson where you type and print a letter. Easy walks you through every step you must make to write and print a letter. It looks like you are using Easy; the actual menu screens you will see appear and you make the choices just as if you were using Easy.

You begin the lesson by learning how to start Easy from the A> prompt; the screen blanks and the A> prompt appears. A small window with an arrow pointing to the prompt tells you to type 'Easy' and then press return. The Opening Menu appears and a window tells you how to select word processing. Then you are walked through the Easy Main menu where you create a new file to edit. You are shown how to write a letter, save it and finally print it. After completing this part of the Tutor, a novice should have no problems typing a letter or report from start to finish.

The third topic covers the 'Fancier Features'. Easy considers fancy features such things as moving blocks of text, making text print in boldface, changing margins and reforming paragraphs. My only complaint with the tutorial is that it doesn't cover the complicated things. Using the dictionary for checking spelling is not covered nor does the tutorial cover using Find and Replace.

Installation

I installed Easy on 3 computers with different drive configurations, one computer had a hard disk, another had two 360 K drives and the third had a 360 K drive and a 720 K 3.5 inch drive. Installation was easy on the computers that had two 360 K drives or a hard disk. With floppies, you need 7 formatted 360 K disks if you want to use EasyMail. If you just want to use the word processor only, 4 disks are needed. Installing Easy and EasyMail on a hard disk requires about 1.3 megabytes.

To begin the installation, you put the distribution disk labeled 'Installation' in drive a: and type 'Easy'. The installation program explains how to make choices from a menu and lets you practice making choices by using the arrow keys on a practice menu. The installation begins by assuming you have a monochrome monitor and the first question you have to answer is whether you have color or not.

The program then asks whether you want a Standard, Custom or Network installation. I tried a Network installation to see what would happen; it told me "Network Installation not available with this release". When I installed Easy on the computers with the two 360 K drives and the hard drive, I used the standard installa-

tion. Easy was very easy to install on the floppies and the hard disk. I just followed simple instructions, like "Put the distribution disk labeled Printers in drive A: and press return" or "Remove the disk from drive B: and label it Dictionary".

With an unusual disk drive configuration, the installation is definitely not easy. The standard installation is for 360 K floppies, so on the computer with a 3.5 inch disk drive, I chose the custom installation. It asked what drive I was copying Easy to and I told it C:; it then asked what type of drive C: was. It said choose 720 K floppy to copy to 3-1/2 inch disks. When I selected the 720 K drive, Easy came back and told me that "720 K floppy installation not available with this release."

I called the MicroPro Technical Services Department (not an 800 number by the way) for help installing it. I got a machine telling me to wait for the next available Technical Representative. And while I waited, I got to listen to a yuppie radio station. Between the ads for video dating services and spa repairs, a voice would come on telling me to have my serial number ready for 'faster' service. After 10 minutes I hung up. The next day, I called again and waited 24 minutes before I hung up. Other people may have better luck calling MicroPro than I did. It could be the time of day I called (3:30 P.M. Pacific time) or it could be the time of year I called (3 days before Christmas!). MicroPro offers a premium service that offers priority service and a toll free number, however, a year's subscription costs nearly as much as Easy Extra does.

I wrote a letter to MicroPro describing my problem and asking for help. Almost exactly 3 months later, I received a reply. They sent me a new set of disks and a nice letter that contained detailed instructions on how to use the new disks to install Easy on a 720 K drive.

Compatibility with WordStar 4 and WordStar 3.3

Easy looks and works much like WordStar 4. Easy files are compatible with both WordStar 4.0 and 3.3; you can take a file written with Easy and can edit or print it using WordStar with no problems. Furthermore, files written using WordStar can be used by Easy even though the WordStar file contains commands that Easy doesn't support. For example, there is absolutely no way you make Easy print something in italics. Never-the-less, if the commands were entered into a file using WordStar, then Easy will mark the text on the screen as being in italics and will print the text in italics.

The documentation and Help cover only 7 dot commands. Easy uses dot commands to set headers and footers, set line height, force page breaks, and control page numbering. Easy seems to have no trouble with WordStar dot commands. I

entered the command '.PC 3' into an Easy file; this command forces the page number to be printed in column 3. Easy handled this command perfectly even though it is not mentioned in the Easy documentation.

If you are comfortable using WordStar, you will have no trouble using Easy. But you won't want to; if you know WordStar well, you will find the menus frustrating because they slow you down. However, my wife thinks the menus are great. To her, menus mean she can get on with writing. Having to learn WordStar would slow her down.

Use with ZPC

Easy will not run on the Z-100 under ZPC, even if the Hardware Support Board is installed.

Conclusions

I highly recommend that you have a hard disk if you want to use Easy. If you try to operate Easy with two 360 K floppies, be prepared to spend lots of time shuffling disks. You start Easy from the 'Easy System' disk in the A: drive. If you select Word Processing from the Opening Menu, you have to remove the 'Easy System' disk in drive a: and put in the 'Easy Word Processing' disk. If you want to check the spelling, you have to remove the 'Easy Word Processing' disk and put in the 'Dictionary' disk. If you try to quit after checking the spelling, you have to change disks 3 times to get back to DOS. Easy always prompts you on what disk to put in what drive and it is never confusing, but it is tedious.

In an otherwise excellent product, it is a shame to see that MicroPro has made the Find and Replace menu option confusing. At the very least, MicroPro should make the help file and Guide match the way Easy really works. There must be a better way for Easy to ask if you want to find a phrase or find and replace a phrase. I haven't seen a program in many years that uses +/- as a way to answer yes or no questions.

I found it difficult to get technical assistance from MicroPro over the telephone and their replies to written inquiries are slow. Unless you have a friend who could help, it would probably be best for a novice to buy Easy from a local dealer who can offer support. The only parts of Easy that I found that might confuse a novice were the installation on computers with high density disk drives and in the Find and Replace menu option. Installation with any other disk configuration than a hard disk or two 360 K floppy disks would probably be too difficult for a novice.

Easy is an excellent choice for people who want a capable word processor, but don't want to invest a lot of time learning complicated commands. Easy is powerful

and can do most of the things WordStar can do, but it is much easier to learn than WordStar. My wife is happy; the two hours a month she spends at the computer are productive because Easy really is easy.

Easy Extra \$99.00
MicroPro International
Attn: Customer Service
P.O. Box 7079
San Rafael, CA 94901-7079
(800) 227-5609 (Orders only) *

EXPLORE NEW WORLDS WITH HUG GAME SOFTWARE



VT340

Resolution and Color
on your
Zenith PCs
using
ZSTEM 240
Terminal Emulation

Call or write for details:
KEA SYSTEMS LTD.
#412 - 2150 West Broadway
Vancouver, B.C. CANADA V6K 4L9
Tel: 604-732-7411 Telex: 04-352848 VCR
FAX: 604-732-0715
Order Toll Free: (800) 663-8702



Circle # 111 on Reader Service Card

The following HUG Price List contains a list of all products in the HUG Software Catalog and Software Catalog Update #1. For a detailed abstract of these products, refer to the HUG Software Catalog, Software Catalog Update #1, or previous issues of REMark.

HUG Price List

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM		DESCRIPTION	PRICE
		H8 — H/Z-89/90			
ACCOUNTING SYSTEM	885-8047-37	CPM	BUSINESS		20.00
ACTION GAMES	885-1220-37	CPM	GAME		20.00
ADVENTURE	885-1010	HDOS	GAME		10.00
ASCIRITY	885-1238-37	CPM	AMATEUR RADIO		20.00
AUTOFILE (Z80 ONLY)	885-1110	HDOS	DBMS		30.00
BHBASIC SUPPORT PACKAGE	885-1119-37	HDOS	UTILITY		20.00
CASTLE	885-8032-37	HDOS	ENTERTAINMENT		20.00
CHEAPCALC	885-1131-37	HDOS	SPREADSHEET		20.00
CHECKOFF	885-8010	HDOS	CHECKBOOK SOFTWARE		25.00
DEVICE DRIVERS	885-1105	HDOS	UTILITY		20.00
DISK UTILITIES	885-1213-37	CPM	UTILITY		20.00
DUNGEONS & DRAGONS	885-1093-37	HDOS	GAME		20.00
FLOATING POINT PACKAGE	885-1063	HDOS	UTILITY		18.00
GALACTIC WARRIORS	885-8009-37	HDOS	GAME		20.00
GALACTIC WARRIORS	885-8008-37	CPM	GAME		20.00
GAMES 1	885-1029-37	HDOS	GAMES		18.00
HARD SECTOR SUPPORT PACKAGE	885-1121	HDOS	UTILITY		30.00
HDOS PROGRAMMERS HELPER	885-8017	HDOS	UTILITY		16.00
HOME FINANCE	885-1070	HDOS	BUSINESS		18.00
HUG DISK DUPLICATION UTILITIES	885-1217-37	CPM	UTILITY		20.00
HUG SOFTWARE CATALOG	885-4500	VARIOUS	PRODUCTS THRU 1985		9.75
HUGMAN & MOVIE ANIMATION	885-1124	HDOS	ENTERTAINMENT		20.00
INFO. SYSTEM AND TEL. & MAIL SYSTEM	885-1108-37	HDOS	DBMS		20.00
LOGBOOK	885-1107-37	HDOS	AMATEUR RADIO		30.00
MAGBASE	885-1249-37	CPM	GAZETTE PUBLISHING		25.00
MAPLE	885-8005	HDOS	COMMUNICATION		35.00
MAPLE	885-8012-37	CPM	COMMUNICATION		35.00
MICRONET CONNECTION	885-1122-37	HDOS	COMMUNICATION		20.00
MISCELLANEOUS UTILITIES	885-1089-37	HDOS	UTILITY		20.00
MORSE CODE TRANSCIVER	885-8016	HDOS	AMATEUR RADIO		20.00
MORSE CODE TRANSCIVER	885-8031-37	CPM	AMATEUR RADIO		20.00
PAGE EDITOR	885-1079-37	HDOS	UTILITY		25.00
PROGRAMS FOR PRINTERS	885-1088	HDOS	UTILITY		20.00
REMARK VOL 1 ISSUES 1-13	885-4631	N/A	1978 TO DECEMBER 1980		20.00
RUNOFF	885-1028	HDOS	TEXT PROCESSOR		35.00
SCICALC	885-8027	HDOS	UTILITY		20.00
SMALL BUSINESS PACKAGE	885-1071-37	HDOS	BUSINESS		75.00
SMALL-C COMPILER	885-1123	HDOS	LANGUAGE		30.00
SOFT SECTOR SUPPORT PACKAGE	885-1122-37	HDOS	UTILITY		20.00
STUDENT'S STATISTICS PACKAGE	885-8021	HDOS	EDUCATION		20.00
SUBMIT (Z80 ONLY)	885-8006	HDOS	UTILITY		20.00
TERM & HTOC	885-1207-37	CPM	COMMUNICATION & UTILITY		20.00
TINY BASIC COMPILER	885-1132-37	HDOS	LANGUAGE		25.00
TINY PASCAL	885-1086-37	HDOS	LANGUAGE		20.00
UDUMP	885-8004	HDOS	UTILITY		35.00
UTILITIES	885-1212-37	CPM	UTILITY		20.00
UTILITIES BY PS	885-1126	HDOS	UTILITY		20.00
VARIETY PACKAGE	885-1135-37	HDOS	UTILITY & GAMES		20.00
WATZMAN ROM SOURCE & DOC	885-1221-37	CPM	H19 FIRMWARE		30.00
WATZMAN ROM	885-4600	N/A	H19 FIRMWARE		45.00
WHEW UTILITIES	885-1120-37	HDOS	UTILITY		20.00
XMET ROBOT X ASSEMBLER	885-1229-37	CPM	UTILITY		20.00
Z80 ASSEMBLER	885-1078-37	HDOS	UTILITY		25.00
Z80 DEBUGGING TOOL (ALDT)	885-1116	HDOS	UTILITY		20.00

Make the no-hassle connection with your modem today! **HUGMCP** doesn't give you long menus to sift through like some modem packages do. With **HUGMCP**, **YOU'RE** always in control, not the software. Order **HUG P/N 885-3033-37** today, and see if it isn't the easiest-to-use modem software available. Joe Katz says it was so easy to use, he didn't even need to look at the manual. "It's the only modem software that I use, and I'm in charge of both HUG bulletin boards!" says Jim Buszkiewicz. **HUGMCP** runs on ANY Heath/Zenith computer that's capable of running MS-DOS!

HEPCAT is here! **HEPCAT** is here! **HEPCAT** is here! So what is **HEPCAT**, you may ask? Why it's just another Pat Swayne **SUPER-UTILITY**. **HEPCAT** is an acronym for **HUG Engineer's and Programmer's Calculation Tool**. Just what we don't need, another memory resident calculator, right? Wrong! With **HEPCAT**, you can throw away the rest and use the best. **HEPCAT** only uses two partial lines on your screen, and best of all, does **NOT** cause existing programs to stop executing! That means, while your computer is grinding numbers internally, you can be grinding them externally. Order **HUG P/N 885-3045-37**.

Can't remember how to use the MS-DOS 'COPY' command? Forget the exact command line format for 'ASGNPART'. Too far to go for the MS-DOS manuals on the shelf on the other side of the room? Why not just type 'HELP' on the keyboard? You say it comes back with "Bad command or file name"? It wouldn't if you had HUG's **HELP** program. With **HELP** installed on your hard disk, all you need to do is type 'HELP' for a complete list of MS-DOS commands and transients along with a brief explanation of how each command works, as well as the format for its use. **HELP**, **HUG P/N 885-8040-37**, works on ALL Heath/Zenith computers that run MS-DOS!

H8 — H/Z-89/90 — H/Z-100 (Not PC)

ADVENTURE	885-1222-37	CPM	GAME	10.00
BASIC-E	885-1215-37	CPM	LANGUAGE	20.00
CASSINO GAMES	885-1227-37	CPM	GAME	20.00
CHEAPCALC	885-1233-37	CPM	SPREADSHEET	20.00
CHECKOFF	885-8011-37	CPM	CHECKBOOK SOFTWARE	25.00
COPYDOS	885-1235-37	CPM	UTILITY	20.00
DISK DUMP & EDIT UTILITY	885-1225-37	CPM	UTILITY	30.00
DUNGEONS & DRAGONS	885-1209-37	CPM	GAMES	20.00
FAST ACTION GAMES	885-1228-37	CPM	GAME	20.00
FUN DISK I	885-1236-37	CPM	GAMES	20.00
FUN DISK II	885-1248-37	CPM	GAMES	35.00
GAMES DISK	885-1206-37	CPM	GAMES	20.00
GRADE	885-8036-37	CPM	GRADE BOOK	20.00
HRUN	885-1223-37	CPM	HDOS EMULATOR	40.00
HUG FILE MANAGER & UTILITIES	885-1246-37	CPM	UTILITY	20.00
HUG SOFTWARE CATALOG UPDATE #1	885-4501	VARIOUS	PRODUCTS 1983 THRU 1985	9.75
KEYMAP CPM-80	885-1230-37	CPM	UTILITY	20.00
MBASIC PAYROLL	885-1218-37	CPM	BUSINESS	60.00
MICRONET CONNECTION	885-1224-37	CPM	COMMUNICATION	16.00
NAVPROGSEVEN	885-1219-37	CPM	FLIGHT UTILITY	20.00
REMARK VOL 3 ISSUES 24-35	885-4003	N/A	1982	20.00
REMARK VOL 4 ISSUES 36-47	885-4004	N/A	1983	20.00
REMARK VOL 5 ISSUES 48-59	885-4005	N/A	1984	25.00
REMARK VOL 6 ISSUES 60-71	885-4006	N/A	1985	25.00
REMARK VOL 7 ISSUES 72-83	885-4007	N/A	1986	25.00

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
SEA BATTLE	885-1211-[37]	CPM	GAME	20.00
UTILITIES BY PS	885-1226-[37]	CPM	UTILITY	20.00
UTILITIES	885-1237-[37]	CPM	UTILITY	20.00
X-REFERENCE UTILITIES FOR MBASIC	885-1231-[37]	CPM	UTILITY	20.00
ZTERM	885-3003-[37]	CPM	COMMUNICATION	20.00

H/Z-100 (Not PC) Only

ACCOUNTING SYSTEM	885-8048-37	MSDOS	BUSINESS	20.00
CALC	885-8043-37	MSDOS	UTILITY	20.00
CARDCAT	885-3021-37	MSDOS	BUSINESS	20.00
CHEAPCALC	885-3006-37	MSDOS	SPREADSHEET	20.00
CHECKBOOK MANAGER	885-3013-37	MSDOS	BUSINESS	20.00
CP/EMULATOR	885-3007-37	MSDOS	CPM EMULATOR	20.00
DBZ	885-8034-37	MSDOS	DBMS	25.00
ETCHDUMP	885-3005-37	MSDOS	UTILITY	20.00
EZPLOT II	885-3049-37	MSDOS	PRINTER PLOTTING UTILITY	25.00
GAMES CONTEST PACKAGE	885-3017-37	MSDOS	GAMES	25.00
GAMES PACKAGE II	885-3044-37	MSDOS	GAMES	25.00
GRAPHICS	885-3031-37	MSDOS	ENTERTAINMENT	20.00
HELPSCREEN	885-3039-37	MSDOS	UTILITY	20.00
HUG BACKGROUND PRINT SPOOLER	885-1247-37	CPM	UTILITY	20.00
KEYMAP	885-3046-37	MSDOS	UTILITY	20.00
KEYMAP II	885-3010-37	MSDOS	UTILITY	20.00
KEYMAP CP	885-1245-37	CPM	UTILITY	20.00
MATH	885-8023-37	CPM	COMMUNICATION	35.00
MATH PAPER	885-8030-37	MSDOS	EDUCATION	20.00
ORBITS	885-8041-37	MSDOS	EDUCATION	25.00
POKER PART	885-8042-37	MSDOS	ENTERTAINMENT	20.00
SCICALC	885-8028-37	MSDOS	UTILITY	20.00
SKYVIEWS	885-3015-37	MSDOS	ASTRONOMY UTILITY	20.00
SMALL-C COMPILER	885-3026-37	MSDOS	LANGUAGE	30.00
SPELLS	885-3035-37	MSDOS	SPELLING CHECKER	20.00
SPREADSHEET CONTEST PACKAGE	885-3017-37	MSDOS	VARIOUS SPREADSHEETS	25.00
TREE-ID	885-3036-37	MSDOS	TREE IDENTIFIER	20.00
USEFUL PROGRAMS I	885-3022-37	MSDOS	UTILITIES	30.00
UTILITIES	885-3027-37	MSDOS	UTILITY	20.00
Z100 WORDSTAR CONNECTION	885-3029-37	MSDOS	UTILITY	20.00
ZBASIC DUNGEONS & DRAGONS	885-3009-37	MSDOS	GAME	20.00
ZBASIC GRAPHIC GAMES	885-3011-37	MSDOS	GAMES	20.00
ZBASIC GAMES	885-3012-37	MSDOS	GAMES	20.00
ZPC II	885-3037-37	MSDOS	PC EMULATOR	60.00
ZPC UPGRADE DISK	885-3038-37	MSDOS	UTILITY	20.00

H/Z-100 And PC Compatibles

ADVENTURE	885-3016	MSDOS	GAME	10.00
ASSEMBLY LANGUAGE UTILITIES	885-8046	MSDOS	UTILITY	20.00
BOTH SIDES PRINTER UTILITY	885-3048	MSDOS	UTILITY	20.00
CXREF	885-3051	MSDOS	UTILITY	17.00
DEBUG SUPPORT UTILITIES	885-3038	MSDOS	UTILITY	20.00
DPATH	885-8039	MSDOS	UTILITY	20.00
HADES	885-3040	MSDOS	UTILITY	40.00
HELP	885-8040	MSDOS	UTILITY	20.00
HEPCAT	885-3045	MSDOS	UTILITY	35.00
HUG BACKGROUND PRINT SPOOLER	885-3029	MSDOS	UTILITY	20.00
HUG EDITOR	885-3012	MSDOS	TEXT PROCESSOR	20.00
HUG MENU SYSTEM	885-3020	MSDOS	UTILITY	20.00
HUG SOFTWARE CATALOG UPDATE #1	885-4501	VARIOUS	PROD 1983 THROUGH 1985	9.75
HUGMCP	885-3033	MSDOS	COMMUNICATION	40.00
HUGPBBS SOURCE LISTING	885-3028	MSDOS	COMMUNICATION	60.00
HUGPBBS	885-3027	MSDOS	COMMUNICATION	40.00
ICT 8080 TO 8088 TRANSLATOR	885-3024	MSDOS	UTILITY	20.00
MAGBASE	885-3050	VARIOUS	MAGAZINE DATABASE	25.00
MATT	885-8045	MSDOS	MATRIX UTILITY	20.00
MISCELLANEOUS UTILITIES	885-3025	MSDOS	UTILITIES	20.00
PS's PC & Z100 UTILITIES	885-3052	MSDOS	UTILITY	20.00
REMARK VOL 5 ISSUES 48-59	885-4005	N/A	1984	25.00
REMARK VOL 6 ISSUES 60-71	885-4006	N/A	1985	25.00
REMARK VOL 7 ISSUES 72-83	885-4007	N/A	1986	25.00
REMARK VOL 8 ISSUES 84-95	885-4008	N/A	1987	25.00
SCREEN DUMP	885-3043	MSDOS	UTILITY	30.00
UTILITIES II	885-3014	MSDOS	UTILITY	20.00

PC Compatibles

ACCOUNTING SYSTEM	885-8049	MSDOS	BUSINESS	20.00
CARDCAT	885-6006	MSDOS	CATALOGING SYSTEM	20.00
CHEAPCALC	885-6004	MSDOS	SPREADSHEET	20.00
CP/EMULATOR II & ZEMULATOR	885-6002	MSDOS	CPM & Z100 EMULATORS	20.00
DUNGEONS & DRAGONS	885-6007	MSDOS	GAME	20.00
EZPLOT II	885-6013	MSDOS	PRINTER PLOTTING UTILITY	25.00
GRADE	885-8037	MSDOS	GRADE BOOK	20.00
HAM HELP	885-6010	MSDOS	AMATEUR RADIO	20.00
KEYMAP	885-6001	MSDOS	UTILITY	20.00
LASERWRITER CONNECTION	885-8050	MSDOS	PRINTER UTILITY	40.00
PS's PC UTILITIES	885-6011	MSDOS	UTILITIES	20.00
SCREEN SAVER PLUS	885-6009	MSDOS	UTILITIES	20.00
SKYVIEWS	885-6005	MSDOS	ASTRONOMY UTILITY	20.00
TCSPELL	885-8044	MSDOS	SPELLING CHECKER	20.00
ULTRA RTTY	885-6012	MSDOS	AMATEUR RADIO	20.00

Magazines everywhere, and no way to reference the wealth of information they hold? Not anymore! Now there's **MAGBASE**; a database designed specifically for referencing magazine articles. Don't let those one-hundred-and-some back issues of REMark, or C Users Journal, or Veterinary Medicine, (or any magazine) gather dust, use **MAGBASE**, and find that article you read two years ago! **MAGBASE** is available for **MSDOS HUG P/N 885-3050** or **CP/M (P/N 885-1249-[27])**.

885-3050 & 885-1249-[37] MAGBASE Update

An update for both the MS-DOS and CP/M versions of **MAGBASE** is now available. A small bug has been found, and will not appear until enough files have been entered in one category to fill the internal buffer. Doing search will find one buffer full of articles, but further searching on that category will fail. A new version is now available at no charge to original owners of **MAGBASE**. To receive this free update, send your original disk to: HUG, P.O. Box 217, Benton Harbor, MI 49022-0217.

LAPTOP OWNERS . . . don't feel left out! All of HUG's MSDOS software is available on 3-1/2" micro-floppies too! When ordering, just add a "-80" to the 7-digit HUG part number. For the standard 5-1/4" floppy, just add a "-37".

ORDERING INFORMATION

For VISA and MasterCard phone orders, telephone the Heath Users' Group directly at (616) 982-3463. Have the part number(s), descriptions, and quantity ready for quick processing. By mail, send your order, plus 10% postage and handling (\$1.00 minimum charge, up to a maximum of \$5.00) to: Heath Users' Group, P.O. Box 217, Benton Harbor, MI 49022-0217. VISA and MasterCard require minimum \$10.00 order. No C.O.D.s accepted.

Questions regarding your subscription? Call Margaret Bacon at (616) 982-3463.

***** Z-100 SERIES SOFTWARE *****

PART NUMBER	DESCRIPTION	LIST PRICE	SALE PRICE
MS-463-1	Z-Basic (16 bit)	\$175.00	\$12.00
MS-463-7	Multiplan	\$195.00	\$12.00
MS-253-1	Basic-80 (8-bit)	\$175.00	\$12.00
CD-463-2	Condor File Manager	\$299.00	\$12.00
LT-Z100	All 4 Listed Above	\$819.00	\$40.00

***** IBM COMPATIBLE SOFTWARE *****

PART NUMBER	DESCRIPTION	LIST PRICE	SALE PRICE
MS-5063-30	Microsoft Windows	\$ 99.00	\$ 24.00
NU-413	Norton Utilities Adv.	\$150.00	\$ 99.00
WP-528	WORDPERFECT 5.0	\$495.00	\$269.00
BO-290	QUATTRO	\$239.00	\$179.00
LO-311	LOTUS 123	499.00	\$359.00

***** ZENITH LAPTOP COMPUTERS *****

SUPERPORT 184-1	2 3 1/2" Floppy Drives, 640K RAM	\$1699.00
SUPERPORT 184-2	3 1/2" Floppy, 20 Meg Hard Disk, 640K RAM	\$2497.00
SUPERPORT 286-20	12/6 MHz, 80286 CPU, 3 1/2" Floppy, 20 MEG Hard Disk	\$3492.00

***** VIDEO MONITORS *****

ZCM-1490	ZENITH Color Flat Screen VGA	\$718.00
MA2565	SAMSUNG Amber TTL 720x350	\$89.00
MA2571	SAMSUNG White TTL 720x350	\$99.00
CW4644	SAMSUNG Color RGB 640x200	\$274.00
CM4531	SAMSUNG Color EGA 640x350	\$389.00
CN4551	SAMSUNG Multi-sync VGA 800x560	\$489.00

***** ZENITH PC COMPUTER UPGRADES *****

SmartWatch from FBE Research Installs in ROM Socket on CPU Board in Zenith computer series Z-100/138/148/150/160. This clock/calender contains a ten year battery and keeps your computer informed of both time and date at each boot-up. Instructions and software included. \$38.00

Z-150 Series Hard Disk Drive Kit Includes new generation High Speed (28 MS) Seagate Drive with Auto Park heads. Each kit is complete with controller card, cables, hardware and instructions to mount the Hard Disk under your two floppy drives in the Z-150 series computers. 32 MEG ST-138/150 Kit \$383.00

Z-148 Hard Disk Drive Kit Includes the Hard Disk Drive and controller in the kit above plus the PTZ-148 Expansion card described below. Each kit includes all cables, hardware and instructions required to replace one floppy drive with a high speed low power Hard Disk Drive. 32 MEG ST-138/PTZ-148 Kit \$459.00

INTERNAL MODEM Fully Hayes compatible (software included)
1200/300 baud \$94.00
2400/1200/300 baud \$159.00

EXTERNAL MODEM Fully Hayes compatible (software included)
1200/300 baud \$137.00
2400/1200/300 baud \$199.00

ZENITH KEYBOARDS These were used in a training class but function like new. 86 Key Keyboard for Z-200 Series only. A bargain at \$24.50

PTZ-148 Expansion Card adds 2 IBM expansion slots \$79.00
with SMARTWATCH clock/calender. \$114.00

VCE 150 Video Eliminator for Z-150
Allows use of EGA or any video card. Required memory chip included. \$54.00

V-20 Chips High Speed NEC V-20-8 8088 replacement. These run at up to 8 MEG and are said to increase CPU speed 10-30% \$14.75

***** Z-100 SERIES COMPUTER UPGRADES *****

High Density 1.2 Meg Drives. External floppy drive set-up complete with drive, power supply, case and cable. Ready to connect to your 8" floppy controller
Single Drive Unit 252.00 Dual Drive Unit \$369.00
Bare Drive and Cable for internal mount \$154.00

SmartWatch by FBE Research. If you don't have a clock for your Z-100, get this one. More details under Z-150 upgrade listings \$38.00

Gemini Emulator Board. Makes the Z-100 compatible with the IBM PC library of programs. \$432.00

UCI EASY PC. IBM PC Emulator. Makes your Z-100 IBM Software Compatible. Full 8 MEG operation, color graphics and audio compatible. \$477.00

UCI Easy87. Add an 8087 Numeric Coprocessor. \$69.00 for the board without an 8087 Chip. With 5 MEG 8087 \$188.00 or with 8 MEG 8087 installed . . . \$234.00

ZMF100A by FBE Research. A modification package which allows 256K chips to be used on the old-style motherboard to reach 768K. Simple assembly with no soldering or trace cutting. Compatible with Easy PC and Gemini Emulator. . . \$60.00
Requires 27 of the 256K RAM chips to complete the kit.

UCI Memory Upgrade Pal Chip Set For the Z-100's with the newer motherboard part number 181-4918 or greater. This chip set allows the installation of 256K RAM chips on the motherboard. With the addition of 27 256K RAM chips, a total memory of 768K is obtained. PAL Chip Set \$64.00

UCI Memory Upgrade Card We recommend this one highly. The board has sockets for up to 2 MEG of RAM. With no RAM installed \$288.00. Add \$35.00 for EasyDrive RAM Drive Software if desired. Either 64K or 256K RAM chips may be used to complete this kit.

UCI EASY-I/O S-100 board that provides IBM PC communications port compatibility with your EasyPC. Easy/I/O-1, One Serial Port \$91.00. Easy/I/O-2, Two Serial Ports, One Game Port, Clock-Calender \$127.00

UCI EasyWin Winchester Drive Systems at reasonable prices. Complete Hard Disk Systems for mounting inside your Z-100. Systems complete with Seagate Drives, 21 MEG \$598.00, 31 MEG \$634. System without Drive \$317.00

CDR Z-100 Speed Module Run your Z-100 Computer at 7.5 MHz. Installs easily with no soldering. Externally switchable between Speed and Normal mode. Payload \$44.00

***** FLOPPY DISK DRIVES *****

MITSUBISHI MF501	5.25" 48 TPI DS/DD 320K/360K	\$ 95.00
MITSUBISHI MF504	5.25" High Density 360K/1.2 MEG	\$124.00
MITSUBISHI M-353	3.5" in 5.25" frame 720K	\$103.00
MITSUBISHI M-355	3.5" in 5.25" frame 1.44 MEG	\$139.00
	M-355 Software Driver	\$ 19.00

M-355 runs on AT compatible or special controller only.

**** SEAGATE HARD DISK DRIVES *****

ST-125	21 MEG, 28 MS, Auto Park Heads With Controller & Cables	\$ 275.00 \$ 329.00
ST-138	31 MEG, 28 MS, Auto Park Heads With Controller & Cables	\$ 329.00 \$ 383.00
ST-238	31 MEG, 65 MS, RLL With RLL Controller & Cables	\$ 278.00 \$ 323.00
ST-251	42 MEG, 40 MS, Auto Park, Software With Controller & Cables	\$ 384.00 \$ 438.00
ST-251-1	42 MEG, 28 MS, Auto Park, Software With Controller & Cables	\$ 465.00 \$ 519.00
ST-4096	82 MEG, 28 MS, Auto Park, Software.	\$ 697.00

V-20 Chips High Speed NEC V-20-8 8088 replacement. These run at up to 8 MEG and are said to increase CPU speed 10-30% \$14.75

***** BUILD YOUR OWN CLONE *****

Kit includes DTK 8088 Motherboard, 150 watt power supply, one each, serial, parallel and game ports, Phoenix BIOS, clock/calender, 4.77/10 MHz CPU, 256K RAM expandable to one MEG, 101 key Keyboard, front panel mounted TURBO and reset buttons, one 360K floppy drive and detailed assembly instructions. \$479.00

A Video card is required. Purchase one to match your monitor.
TTL Monochrome Video card \$43.00
CGA Color Video card \$39.00
EGA Color Video card \$135.00
VGA Color Video card \$265.00

OPTIONS:		
ST-138 30 MEG Hard Disk / Controller and cables add	\$369.00
2nd Floppy Drive . . . 360K add	\$88.00
 add	\$95.00
720K		
Addition RAM	CALL
MS-DOS 3.3 add	\$85.00
2nd serial port add	\$18.50
Payload assemble and test add	\$48.00

**PAYLOAD
COMPUTER SERVICES**



15718 SYLVAN LAKE
HOUSTON, TEXAS 77062
PHONE (713) 486-0687



Your satisfaction is guaranteed. All hardware carries a 90 day Payload warranty. VISA and MASTERCARD orders welcome with no surcharges. Add \$5.00 to all prepaid orders for handling and shipping in Continental USA, we pay the balance. Actual shipping costs for foreign, overseas and net billing orders to approved accounts. We accept purchase orders from schools, government and approved accounts. Mail or Phone your order for prompt service. Texas residents please add 8.0% state sales tax.



**On
The
Leading
Edge**

William M. Adney
P.O. Box 531655
Grand Prairie, TX 75053-1655
Copyright © 1988 by William M. Adney

**Christmas, HUG Software,
Mace Utilities, Nvelope, Imager,
Gofer, Tornado, MS-DOS Course,
C Programming Course**

Even though it may be a little early, let me be the first to wish you and yours a Happy Holiday season. And, as has been my custom over the last few years, I will offer a few suggestions for gifts that you might want to consider for your favorite computer user or even treat yourself. As usual, I have found a number of items that are under \$100, but I have included one item that is so outstanding that I made an exception to that limit. If you need to drop a subtle hint to your Santa Claus, you can do so by circling the appropriate item on the list at the end of the article.

I will also take a look at some of the more interesting things that have occurred this year and try to place them in some kind of perspective. But before we get into that, let's take a look at some gift ideas.

HUG Software

I think that one of the best reasons for being a HUG member is access to the library of HUG software. There is a wide variety of software available for nearly all Heath computers and operating systems, but I have found a few that I use on a fairly regular basis. One is new, and the other three have been around for a while.

If you are interested in exploring the contents of any disk, then HADES — HUG's All-Purpose Disk Editing System — is an outstanding program. It was written by Jim Buszkiewicz, and it allows you to do all kinds of interesting things. For many users, HADES' most outstanding feature is that it will help you "unerase" files that have been accidentally deleted. You can also use HADES to display and change the file attributes of any file — hidden and system files cannot be hidden from this program. But HADES has a number of other features that can be quite valuable, especially if you are curious about what a disk really contains.

The disk editing system for HADES is precisely that. You can display or change a file in either hex or ASCII mode, and that allows you to directly change the contents of any file. Or, you can look at a disk in the sector mode to see what data are stored where. That is particularly useful if you want to see what the boot sector, FATs, directory, and data area on a disk really contains, as well as where those data are actually located. This kind of exploration requires some technical knowledge, including hex arithmetic, but you can get a feel for a disk's contents without all of that. Some caution is required because you can actually change the contents of anything on a disk, and incorrect changes can make a disk completely unusable. If you are interested in exploring, or if you only want a good program to unerase files, HADES is an excellent choice.

If you are looking for an excellent calculator-type program, then HEPCAT is the

best there is. HEPCAT — the HUG Engineer's and Programmer's Calculation Tool — written by Pat Swayne is the best calculator I have seen. It is more powerful than many spreadsheets because it provides capabilities they just don't have. It is compatible with nearly all software, except for the Samna word processor; but then Samna has compatibility problems with a lot of things, including IBM computers and most users.

HEPCAT is a super memory-resident popup calculator that provides calculating capabilities to perform engineering and programming calculations. For engineers and students, HEPCAT provides standard floating point calculations for sine, cosine, tangent, arc-sine, arc-cosine, arc-tangent, and natural and base 10 logarithms with anti-ln and anti-log functions. And, of course, HEPCAT supports scientific notation that you would expect in an engineering-type calculator.

For programmers, calculations and conversions can be done in a number of radices, including both hex and binary, not to mention decimal. HEPCAT even does logical operations in binary, such as AND and OR — especially useful in debugging programs where one thing does not seem to work exactly right. And HEPCAT has a built-in feature to convert from English to metric units, in case you don't quite feel comfortable with the metric system (I still don't either). All in all, HEPCAT is an extremely useful tool that I really like.

Perhaps you have been thinking about getting a modem to explore the interesting world of computer bulletin boards. If you do decide to get a modem, you will also need some kind of communications software. One of the best, and easiest to learn, is the HUG Modem Control Program or HUGMCP. It is quite powerful and is one of the best values around today. It is easy to set up for your system and is especially good for beginners. Even if you are an advanced user, HUGMCP is still one of the best communications programs that you will find at any price.

One of the newest utilities available from HUG is the Both Sides Printer Utility which is one of those things that, after you see it, you wonder why someone did not think of it before. As you can guess from its name, this program allows you to print a file on both sides of the paper which can obviously cut your paper usage in half. I have tried this program with both Microsoft Word and WordStar, and everything seems to work as you would expect. And I have used the program with both the tractors and sheet feeders for both my DTC StyleWriter and C. Itoh C-310 printers. I used the general defaults that already existed in various files that I tested, and I have to admit that I was surprised that I did not have any problem with the sheet feeders. I have had to spend con-

siderable time configuring my word processors to work with the sheet feeders on both printers because I have never found a driver that includes a sheet feeder for either one. An excellent utility program that provides a very useful function.

I have selected these four programs because I have found them particularly useful and also because they are available for both the Z-100 and PC compatible computers. I have also tried to pick programs from several categories that will be useful to beginners, as well as advanced users. I believe that you will find these programs as useful as I have. But there are a couple of other Heath "software" products that you may also find helpful.

Educational Software

As some of you may have noticed, I shamelessly and unabashedly have recommended some of my own works in this column. Most of you know about the FlipFast books I have written (more on that in a minute), but I completed my first project for the Heath Company Education Department this year: the new MS-DOS Course for version 3.

This course of 983 pages with a program disk includes just about everything you ever wanted to know about MS-DOS (and PC-DOS) and then some. If you are really interested in learning about DOS, there are 33 experiments that will help you do so. The program disk includes several programs used in the course, and my favorite is the LD (List Directory) program that displays an alphabetically sorted directory listing of ALL information, including file attributes, of ALL files on a disk. It also displays the cluster factor for any valid DOS disk, including a hard disk, so there is no question about how many sectors per cluster you have on a disk. This particular program was designed to be used long after you have completed the course. If you are interested in learning about the nuts and bolts of DOS, I think you will find this is an excellent course.

If you are interested in learning a popular programming language, you will find that Heath's new C Programming course is also an excellent value. This course was designed specifically for the novice programmer who wants to learn about a very powerful programming language. As such, it assumes that you have a little knowledge of a high-level language, such as BASIC, but that's about it. Included with this course is my favorite C compiler, the Ecosoft C88 C Compiler that also includes the CED editing/compiling environment.

In order to avoid the inevitable problems associated with specific computer hardware, this course deals strictly with programming functions available in the standard C language. That allows the course to concentrate on the function and syntax of C, instead of getting tangled

up in specific hardware-related discussions that are incredibly confusing to new programmers. One of the most important things to learn in any programming language is its fundamental syntax and usage, and I believe this course does an outstanding job of presenting those subjects. My compliments to author Randy Chafy of the Heath Education Department on this course.

Heath has a wide array of educational courses available for other topics, such as electronics, communications, robotics, lasers, and other current state-of-the-art subjects besides computers. I have often wished I had the time to learn about some of these other things, but you may find one of these courses will help complete your gift list for this holiday season.

All of these items are available from Heath Company and can be ordered by phone as listed at the end of this column. All of these products, with the exception of the CED editor in the C Programming course, will run on both the Z-100 and all PC compatible computers. The CED editor is limited to use on PC compatibles only. Now let's look at some other good products that are available from other vendors.

Paul Mace Software

Paul Mace Software has a couple of useful software packages that I have mentioned before. The Mace Utilities package includes a lot of programs that are especially useful if you have a hard disk. The UNFRAG program is one of the best disk defragmentation programs available today. In fact, a report in a recent issue of PC Magazine (October 11, 1988) says that the Mace UNFRAG program is a "solid, reliable performer" that "does the job thoroughly and safely." I agree. I have found that all of the Mace programs follow that same concept, and I have spent more than a little amount of time trying to get them to fail under some very unusual testing conditions. Perhaps the most important feature of the UNFRAG program is that you can speed up the performance of your system even if you have a "slow" hard disk.

Even if you don't have a hard disk, Mace Utilities includes an "undelete" program that recovers erased files. One way or another, EVERY computer user needs this kind of program once in a while to fix mistakes resulting from the dreaded DEL *.* command. With this package, you also get the DIAGNOSE program that checks ANY disk (floppy or hard) for a potential bad sector, marks them in the FATs, and moves any data in the cluster to a "good" part of the disk. Mace Utilities also includes the "db FIX" disk that has programs to help you recover corrupted dBase files. If you don't already have a utility package that includes these kinds of programs, I highly

recommend the Mace Utilities. Even if you do, you may find that Mace offers more safety and value than your current utilities.

Mace also offers another interesting and useful program called NVELOPE. This program was discussed in more detail in the July 1988 REMark, but it is a memory-resident (usually) program that includes an envelope addressing function with the capability to "grab" a name and address from a document created by most word processors. If you want to save that name and address for later use, you can save it to a special NVELOPE database. All saved names and addresses in your database can be listed with F3 and used again to print labels, envelopes, and a "directory" of that information. Also included in this package is a complete zip code "database" (actually separate files — one for each state) for the United States so that you can look up a zip code for just about any town or city in the country. This program can even print sideways, if necessary, to accommodate laser and other types of printers. If you do much mailing, you will find that NVELOPE is much easier than breaking out the old typewriter to address envelopes and mailing labels.

Going Places and Losing Things

Like any complicated piece of machinery, computers can exhibit all kinds of strange problems. The worst of these is when a hard disk fails, particularly a high capacity hard disk. For those of us that work in the computer field as a profession, we recognize the hazards of high-capacity storage, and I make no apologies for the fact that I continually mention backing up files on floppies and especially hard disks. Unfortunately, I find that most users seem to have a backup strategy that consists of NOT making backups of any files. It is rare that I go to a client's site when someone does not ask me: "How can I recover data from a bad hard disk?"

In most cases, virtually none of the data is recoverable, especially when there is a mechanical failure, such as a head crash or when the drive just stops spinning. Some kinds of media failure, such as a bad sector which suddenly develops, does not usually cause that kind of catastrophic data loss, but it always seems that the media develops a bad spot in the middle of a data file that you need immediately, if not sooner. The best way around that is to make regular backups of your critical files and program disks. If you are using a floppy disk system, that is easy to do with the DISKCOPY command. If you are using a hard disk system, there are a couple of ways that you can ease the backup chore.

DSBACKUP+

I have never really trusted the DOS

BACKUP and RESTORE commands since I found a Microsoft bug in one of the early releases of version 3. Besides that, the BACKUP and RESTORE commands are just plain slow, and a proper backup requires a lot of time and floppy disks.

Design Software's DSBACKUP+ is a fast, reliable, and cost-effective way to perform the backup chore. The latest version I have used on my '248 system is 1.10B, and it allows you to backup selected files, subdirectories or the entire partition. It works great with all standard 3.50" and 5.25" formats that I have tested. The best news is that it is fast — it takes about a minute to backup one megabyte to a 5.25" high-density floppy in my system. And you can configure it with special configuration files that set up the various backup parameters however you want. DSBACKUP+ is quite flexible, too — it can be run in the normal menu-driven mode or direct from the DOS command prompt, if you prefer. The program tells you how many floppies you will need to backup a hard disk partition, and it adjusts the number depending on the capacity of the floppy media you are using. And it will compress the files to some extent, too. If you are looking for a fast and reliable backup program for your hard disk, you will have a difficult time finding a better one than DSBACKUP+.

Even though it is an excellent program, DSBACKUP+ still relies on floppy disks as a backup media. While this is a reliable and inexpensive way to backup a hard disk, it still requires that you sit in front of your computer and change floppies as required. That can be a long and tedious process, especially if you are using a high-capacity hard disk with low-capacity floppies (i.e., 360 KB), and that seems to be the primary reason that many people do not backup a hard disk on a regular basis. Fortunately, there is another way if you don't want to mess with all those floppy disks.

The Imager

Tape backup systems are becoming more popular as the capacities of hard disks grow larger, but most of them have a singular disadvantage — they are relatively expensive to the tune of \$500 and up. And the cost of the special tapes that must be used with most of these units is on the order of \$20 each or so. Even so, many of these units cannot backup a hard disk partition that exceeds 60 MB, and some of them will only backup half that amount to a single tape. It is not too difficult to spend \$700 on an internal backup unit with the cost of tapes and such. If you need an external unit with a power supply, you need to plan on getting very little change back from \$1,000 by the time you buy tapes and everything. Tape backups are certainly much easier from a user perspective, and there is a way to

save a considerable amount of money on a tape backup system if you are a HUG member.

I expect that most of you already have part of the hardware required for a tape backup system: a standard Video Cassette Recorder (either VHS or Beta). Then all you need to perform an incredibly cost-effective tape backup is the Imager.

The Imager system consists of a full-size board that fits in an 8-bit slot on a PC compatible computer, a disk of software, connecting cables for the VCR, and of course a manual. Because the Imager fits in a standard 8-bit slot, it works with nearly all Zenith PC compatible computers (except laptops, of course) up to and including the Z-386. The only real requirement is that you must have space with an 8-bit slot for a full-size board which excludes the Z-148 and the eaZy PC systems. My test systems included both a '248 and a '386, and the Imager works fine with both. I am currently using Zenith MS-DOS 3.21 with BIOS level 3.34, although that should not make any difference as far as backups with the Imager are concerned.

The Imager is remarkably cost effective, especially if you already own a VCR. If you do, you can get the Imager for \$195 if you are a HUG member. If you are not a HUG member, the Imager is \$295. But that is not the end of the story as to how cost effective the Imager really is, because you don't have to spend over \$20 each for special backup tapes — all you need to do is buy high-quality tapes for your VCR which should cost, at the most, about half of that if you shop around for a good buy. Even so, you can still get an excellent value for your tape dollar based on the recording capacities of various kinds of tape, as shown in Figure 1.

Tape Time	Backup Capacity
30 minutes	26 MB
60 minutes	55 MB
90 minutes	83 MB
120 minutes	110 MB

Figure 1
Imager VHS Tape Backup Capacity

Any recording media (e.g., floppy disks or VCR tape) is important when you want to use it for a computer system, and it does not pay to cut corners on this. Remember, you are using this to backup your hard disk, and it doesn't pay to use low-quality (generally cheap) tape for this purpose. Even though the software supplied with the Imager backs up each block of data in four redundant blocks on tape to help prevent problems, low-quality tape might still cause problems at the worst possible time: during a restore of a critical file.

The Imager's software also provides a running value of the "Figure of Merit" based on 100 (100 is a perfect tape) so that you will have a general idea of how good your backup tape is. According to the documentation, a value of 80 or better is reasonable for a high-quality VHS tape, and I was able to achieve a 93 using a Maxell T-120 HGX tape using the recommended SP (Slow Play — the slowest possible speed on my VCR). In this mode, I was able to backup about one megabyte per minute which is a very reasonable time. When I used the /W switch with the Imager's backup command, I was able to achieve a 98 because that switch sets a "slow" backup mode to help overcome any problems in tape quality — a nice feature.

A couple of notes related to the Imager backup software, at least for version 2.8B that I tested. First, it does not backup hidden files, such as the BIOS and System Kernel. I don't consider that a problem unless you are using some kind of copy-protected software that uses hidden files as part of the copy protection. That is the reason that I don't use or review any copy-protected software in my system. Second, Imager does not backup any Read-Only files. Since current versions of DOS allow you to set the Read-Only attribute, I believe that Imager should backup those files, but that did not cause a problem for me. Third, the backup process does not reset the archive attribute for each file that is backed up — I believe it should. And finally, the backup process does not allow you to backup files based on the system date even though the restore command does. I suggest that a /D (Date) switch is also needed for the backup command so that you can perform an incremental backup based on a specified date.

I believe those software limitations are not particularly significant because the best and most efficient use of the Imager is to backup an entire hard disk partition anyway. The manual that I received with the Imager could use some improvement on the installation and software commands, and although it is generally adequate, I strongly recommend you read Joe Katz's report on the "Imager Backup" in the March 1988 issue of REMark (page 51) BEFORE you begin installation. Despite some shortcomings in the manual, I highly recommend the Imager as an excellent and very cost-effective way to backup your hard disk.

Losing More Things

Getting a computer to improve your productivity is one thing, but trying to manage all of the information that you can accumulate is another story. Nearly everyone who buys a computer also buys a word processor, and while that is a usual, and excellent, application for a system,

it also helps you create a lot of documents of one kind or another. After a while, you have so much data stored by your system, it is nearly impossible to find something. And if your desk is like mine (or the way it used to be), there are the ubiquitous scraps of paper that contain all kinds of valuable information. Trying to sort through all this information or trying to find a specific "something" in a disk file can be hazardous to your sanity. You know the story — I just saw that the other day! Information overload is the name of the game, and a couple of vendors have some neat programs to help solve these kinds of problems.

GOfer

Have you ever written something — in a letter or other document — and tried to find it several months later? All you can remember is a specific phrase or a couple of key words, but you cannot recall what specific file they were located in. Even if you keep floppy disks organized by subject, a specific phrase or a few key words can be difficult to find. If you have a hard disk, it may take a long time to search the files with a word processor to find those magic words you are looking for. And as you use your word processor to write more and more documents, the problem becomes larger and larger. What can you do? Maybe you need a GOfer.

Everyone knows what a GOfer does: he goes for this and he goes for that. You can use the GOfer program to search various files, subdirectories or an entire disk for a specified text string or multiple strings. GOfer also has enough smarts to be able to use boolean logic that includes the usual and/or/not operators, as well as "nearby". For example, let's say I wanted to find which of my REMark articles talked about the Z-100 and a hard disk — that's exactly what I would tell GOfer. The search strings would be "Z-100" AND "hard disk". Then, I would tell GOfer which drive, subdirectory, and/or files to check for those "keywords". Begin the search, and GOfer will present the results in several possible ways. My preference is to "View each find on screen", which results in a display showing the text being searched for, along with the file name. I have found this is usually the quickest way to check for the text I am looking for in the context I remember. Once I have found that, I can even copy text from the "found" file into the current text document that I am editing with my word processor.

GOfer is usually run as a memory-resident program, but it can be run in a "stand-alone" mode, too. Unlike some similar programs, you don't have to create a special "index" for use with GOfer to perform the search functions — you can use your word processing files exactly as they are. And GOfer can be used with most popular word processors, such as

Microsoft Word, WordStar, and Word Perfect, as well as other programs like dBase III, Framework, Symphony, and SideKick. GOfier is a very flexible and useful program that can help you keep from losing too many things, at least in your computer files.

Tornado

One of the hazards of the so-called "Information Age" is getting inundated with lots of important data. Some things need to be "retained" on a more or less temporary basis, and other data need to be recorded more permanently. For example, the mileage and/or date of the last oil change on your car is one thing, but trying to find out which area code is associated with what general city is another, not to mention keeping up with a running "To Do" list with important due dates.

There seems to be a lot of these unrelated "information tidbits" that are easy to forget because you don't use them much. And if you try to record some of this kind of information on various scraps of paper, they always seem to get lost or you can never find the critical one you are looking for at the moment. Losing seldom-used, but important, information is just one of the problems that most of us face today. Unfortunately, most of this information is unrelated and unstructured so it is hard to store it in a defined database. Even worse, you may only remember a "piece" of it, such as "Fred somebody" who called about a Super DataWhacker 5000 computer system. Not to mention the other 1001 little data items like: "When did I last paint the house?"

Fortunately, there is a program that can help you cope with all of these need-to-know tidbits of information: TORNADO. Instead of having all those scraps of paper with important information spread all over the house, you can record information with TORNADO in window-shaped notes that are organized in "piles" (i.e., files). More importantly, you can FIND that information easily by doing a "search" for anything on the note that you DO remember or you can simply "look through" a group of these notes.

If you consider that the notes in a file are similar to those on scraps of paper, you can do anything with them that you can do with paper. You can mark a note as "important" or you can "bury" it in the bottom of the pile. Or, you can "throw away" a note just like its paper counterpart. How you organize and manage your notes is strictly up to you. And as you might expect, TORNADO includes an editor with basic word processing functions. The editor includes enough of the standard word processing functions so that you can even use it to write short documents which is, of course, all the "notes" in your database really are. You can also use TOR-

NADO to "grab" information from a foreground program (e.g., a word processor or spreadsheet) to store as a note or you can "put" (i.e., copy) information from a note to a foreground program.

TORNADO has so many interesting and useful features that it is difficult to select the best ones to comment on. Since TORNADO is a memory-resident program, it is important to know that it only takes about 20 K of memory, but even that can be adjusted, if you wish. Capacity? Well, you can have up to 50 piles or files of notes that contain 500 notes each for a total of 25,000 notes. I can't imagine exceeding that limit, but I suppose it's possible.

This software comes in two versions: the "standard" version and TORNADO with Library. The version with the Library costs \$50 more than the standard version and includes four special files: AREA (codes for various cities and time zones, MISC (miscellaneous forms, graphics, and other goodies), FACTS (various conversions and conversion factors, formulas, ASCII codes), and YEAR (basically a calendar that you can use as a diary). Although someone clearly spent a lot of time compiling the useful information in these files, I am not convinced that the information is worth the additional \$50 for the Library version. The standard version contains more than enough useful examples in the sample files to get you started, and I doubt that most people would use the Library files much. All in all, TORNADO is an excellent program that can help you manage and find all the miscellaneous kinds of information you need. Highly recommended.

Looking Back at 1988

Even though the year is not over as I write this, there are some notable news items that have occurred this year. IBM has announced they have sold more than one million PS/2 computers with the Micro Channel Architecture (MCA), and the new OS/2 operating system created by Microsoft has generally become available from Zenith, IBM, and other vendors. Apple has sued Microsoft for copyright infringement based on the revised Windows interface. And it looks like VGA is on its way to becoming the display adapter standard which fits nicely with the dynamic Zenith FTM (Flat Tension Mask) ZCM-1490 monitor that has received rave reviews. But I think the more important news is a little more recent. And it will, in one way or another, shape the type of hardware that is available in the future.

You may have heard that a large number of well-known hardware computer manufacturers have joined together (including Zenith and Compaq) to develop the Extended Industry Standard Architecture, commonly called EISA. This group was formed to develop a computer hard-

ware standard that can hopefully be used in 32-bit computers (e.g., 80386 CPUs) and beyond. To most industry observers, this appears to be a direct response to IBM's exceptionally proprietary views on the use and licensing of the MCA bus.

As you may know, IBM has publicly stated that the MCA bus will be vigorously defended against ALL potential copyright infringements, so it is unlikely that many vendors will even try to "clone" that kind of hardware. In other words, IBM is telling all hardware manufacturers that they had better not try to develop an MCA equivalent or they will be up to their corporate eyeballs in lawsuits. By taking this stance, it seems to me that IBM has managed to shoot a number of large holes in their corporate feet. Even though IBM publicly maintains that sales of PS/2 computers are meeting or exceeding expectations, a look at the sales figures for number of units sold and revenues tends to indicate a different story. These numbers have decreased since the IBM AT was discontinued. I am currently doing some mainframe computer consulting for a couple of very large companies, and my own observations tend to confirm this — I see very few IBM PS/2 computers in these companies. When I asked about this, the most common response is that "it's too much trouble to convert our 5.25" floppy data to the 3.50" disks." In addition, I was also told that it was difficult and expensive to obtain add-on boards, such as memory, for the PS/2 systems. Standard PC compatible boards are NOT compatible with the PS/2, and in fact, cannot even be inserted into the connectors on that system's bus.

When Zenith released the Z-386 with a proprietary 32-bit bus, I was beginning to wonder if we were going back to the "old days" (i.e., before the IBM PC) when every hardware manufacturer had a different kind of bus, and one manufacturer's computer was not at all compatible with another's. Even disk formats were different, and it was sometimes difficult to obtain popular software in one of the Heath formats, particularly for the H-89. The only common link among all of these various brands of computers was that they used the old 8-bit CP/M operating system. Then, IBM developed the bus for the 16-bit PC, and this eventually spawned the real beginning of the personal computer revolution.

Even though a number of hardware vendors are supporting the EISA bus, it will be a while before we see any systems with it, but I think it is a good idea.

Then, of course, we come to the OS/2 operating system. You may hear some people say that "DOS is dead", but don't believe it for an instant. In order to really use the multi-tasking features of OS/2, you must buy new software that was writ-

ten specifically for that operating system. There are already some indications that OS/2-specific applications will be even more expensive than their DOS equivalents, not to mention that OS/2 itself already costs at least twice as much to buy as DOS. It is expensive to "convert" to OS/2, and I doubt that many people will find that expense worthwhile. Assuming that nothing better comes along in the meantime, I think it will be a long time before OS/2 becomes THE standard operating system. There are already some real contenders on the horizon, such as Xenix, and I think it will take at least a couple of years before anything replaces DOS as a standard.

One other item of note: MicroPro has released WordStar version 5.0, and I really like it. WordStar now does "windows", and it has a number of other really nice features, including "pull-down" menus in addition to the "classic" ones. More on that in a future article.

Powering Down

I have received several letters about my request for help in locating various DOS shells that can be used to manage disk files. We'll look at some of the responses next time, although I don't believe that I will be able to get copies of all suggested software by that time. If I can't, I will share some of the comments about it that were included in the letters.

If you have any questions about any thing in this column, or about Heath/Zenith systems, in general, be sure to include a self-addressed, stamped envelope (business size preferred) if you would like a personal reply to your question, suggestion or comment.

Items Discussed

HUG Software

Both Sides Printer Utility
(885-3048) \$20.00
HADES (885-3040) 40.00
HEPCAT (885-3045) 35.00
HUGMCP (885-3033) 40.00
Heath/Zenith Users' Group
P.O. Box 217
Benton Harbor, MI 49022-0217
(616) 982-3838 (HUG Software only)
All HUG Software is 30% off through the month of December -ED

Hardware

Imager \$295.00
(\$195 for HUG Members Only)
The Light Pen Company
Box 45255
Los Angeles, CA 90045-0255
(800) 634-1967

Software

MS-DOS Course (EC-1121-A) \$59.95
C Programming Course

(EC-1124) 99.95
(includes compiler)
Heath/Zenith Computer Centers
Heath Company Parts Department
Hilltop Road
St. Joseph, MI 49085
(800) 253-0570
(Heath Catalog orders only)

NVELOPE \$49.00
Mace Utilities 99.00
Paul Mace Software, Inc.
400 Williamson Way
Ashland, OR 97520
(800) 523-0258 (Orders only)

DSBACKUP+ \$79.95
Design Software
1275 W. Roosevelt Road
Chicago, IL 60185
(800) 231-3088 (except Illinois)
(312) 231-2225 (Illinois only)

GOfer \$79.95
Microlytics, Inc.
One Tobey Village Office Park
Suite #547
Pittsford, NY 14534
(800) 828-6293 (except NY)
(716) 248-9150 (NY only)

Tornado \$99.95
Tornado with Library 149.95
Micro Logic Corp.
100 2nd Street
P.O. Box 174
Hackensack, NJ 07602
(800) 342-5930 (except NJ)
(201) 342-8101 (NJ only) *

Continued from Page 26

plywood stiffener. You can remove the insert for extra depth, but you shouldn't and you won't need to with the ZP-150, Z-181, Z-183, SupersPort, and SupersPort/286. When the insert is in place, those computers fit snugly and right. As added protection on the bottom, for the bag as well as its contents, are a pair of molded rubber platforms that serve as shock absorbers and bumpers against abrasion. If you own a ZP-150, you'll benefit from a padded insert that can subdivide the bag inside into two compartments: use one for the computer and the other for papers or anything else that has relatively low bulk. If your laptop is one of the others from Zenith, you'll need to remove the insert: because it's held in by Velcro fasteners; all you need do is peel it out. That removable insert is a sign that this bag is an "all-purpose" model designed for adaptability to many different brands of laptop computers.

The LeMans is a toploader: the top is a lid fastened by a two-talon nylon zipper and secured by two web belts that terminate in quick-release nylon closures. Stitched to the lid is a web belt with a contoured, padded vinyl handle that is

unusually comfortable. So is the contoured pad stitched to the removable web shoulder sling: when the convex side is turned towards my neck and the sueded underside is on my shoulder, the bag feels stable, snug, and easy to carry. The only problem is that it creaks loudly as I walk with it fully loaded: that annoying sound seems to come from the nylon snap fasteners holding the shoulder strap to the bag. My hunch is that Beseler will remedy quickly this one deficiency in an otherwise splendid bag.

The bag is 13.6" wide, 13.6" high, and 4.75" deep. Sewn to the outside ends are two large compartments, fastened with nylon zippers. The AC adapter for any of Zenith's laptop computers can go into either pocket, leaving the other free for a portable modem or other essentials. These pockets are a nominal 4.75" wide, 10.6" high, and 2" deep: they give a little. Sewn to the front of the bag is a three-quarter-zippered compartment — 12" wide, 10" high, and 2" deep — with four pockets to hold 3.5" floppy diskettes and three slender pockets just right for pens, pencils, or a small screwdriver. I'd like to see one pocket more: to hold the 5.25" diskette needed for software to transfer files between a laptop and many desktop computers. The compartment has enough space for it, and for a manual, notebook, or other papers.

Beseler's LeMans LEC-2 really is a splendid bag for Zenith's laptop computers. It's an auspicious beginning for a new line by a major manufacturer. The bag carries a limited lifetime warranty against defects in material and workmanship.

Products

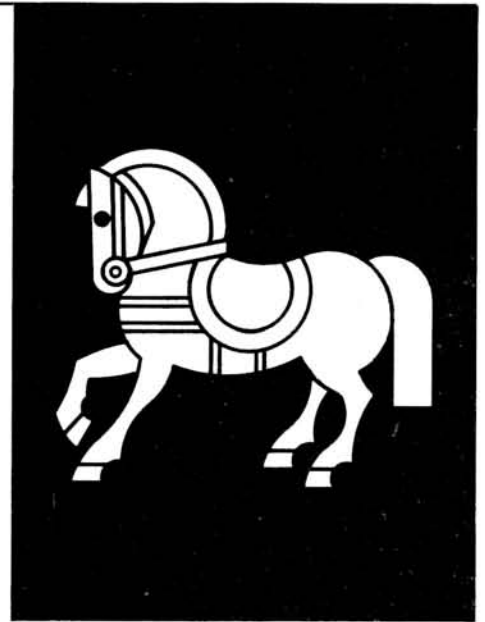
Lighthouse Attache \$39.00
Original Attache \$39.50
Deluxe Attache \$79.00
Lands' End Direct Merchants
1 Lands' End Lane
Dodgeville, WI 53595
(800) 356-4444
(608) 935-9341

Ultimate Laptop Attache \$69.50
Traveling Software
19310 North Creek Parkway
Bothell, WA 98011
(800) 343-8080
(206) 483-8088

Diamond LeMans LEC-2 \$99.00
Beseler
1600 Lower Road
Linden, NJ 07036
(800) BESELER
(201) 862-7999 *

Sabotage

Frank Starr
2730 Pase Drive
Omaha, NE 68123-1720



Viruses and Trojan Horses

I think that we need to start by being straightforward about what creators of Trojan Horse and Virus programs are. They aren't harmless pranksters — often, their pranks are far from harmless, damaging both people and property. I think it is more accurate to refer to them as saboteurs. The dictionary defines a saboteur as: "a person who commits or practices any underhand interference with production, work, etc." I think that is an honest description of these sort of people.

It's hard to see into the mind of someone you haven't met. Maybe this sort of programmer thinks that they are getting even with the military industrial complex. Perhaps they despise progress they perceive as displacing workers, as did the first saboteurs, Dutch weavers who fought automatic knitting machines. Or, they may perceive the entire thing as a harmless practical joke. This sort of mentality excuses things like pouring sand into a gas tank, tying firecrackers onto the tail of a dog, or setting a thumbtack on someone's chair. Allowed to progress unchecked, this kind of thought can ultimately allow things like racial prejudice, and the Holocaust suffered by Jews during World War II.

Some might wonder if I am overreacting. But, if you begin by rationalizing property damage as good clean fun, where do you draw the line? If you say that disabling a large industry or government agency does no harm because "they" have so much money, you ignore

the fact that this money comes from your pocket. This sort of sabotage, like shoplifting or taking office stationery supplies home, increases company expenses, which gets passed along in higher prices, or higher taxes. With the growth of service jobs and industries, it's increasingly more likely that the industry sabotaged by trojan horse programmers will recoup its losses through higher prices, which will come back to said saboteur in higher prices for food, clothing, insurance, and much, much more. Whether these thoughts will halt even one sabotage programmer, it is hard to say. One can dare to hope, but I won't hold my breath.

In the meantime, here is a compilation of information to help impede this sort of vandal. Until these people change their thinking, their criminal acts will remain with us. And, you need to keep informed on your best defenses. Admittedly, unless you can hire a professional consulting firm, your micros will always be at risk. Home computer owners, and small businesses, will have to do their best with small scale precautions. As with car thieves, you ultimately won't be able to stop a professional. But, just as there are common sense precautions to inhibit auto thieves, like not taping a spare key under your hood, so there are similar precautions for protecting your small computer.

One safeguard you can implement is a software write-protect for your DOS system files. Files can have up to four

switches, called attributes, set. Your users' guide manuals can give you details on them all. To write-protect your file, you only have to deal with the read-only attribute. DOS 3.x has a command called ATTRIB. Making the following key-in will protect files you wish to set to read-only:

```
ATTRIB +R COMMAND.COM
```

There are numerous shareware programs which will also let you set file attributes. The same thing can be done with commercial DOS shell programs, such as XTREE and PCTOOLS. There are also shareware DOS shells, such as COMMANDO and DOSAMATIC. You can protect your COMMAND.COM file, as well as some hidden system files, such as MSDOS.SYS and IO.SYS. These last two are ordinarily already protected, however. Depending on how far you feel that you want to go, you may want to write-protect your application program files, as well. Unfortunately, many programs have to have uninhibited access to their main files. You'll just have to experiment around, to see how much software write protection is worth your while. I know from personal experience that setting COMMAND.COM to "read only" will not inhibit your operations.

A precaution which can protect you from "keyboard brats", as well as a hidden time bomb in a batch file, would be to rename your reformat command. A simple defense, this will cause anyone who keys in REFORMAT to be greeted with the message "bad command or

filename". An example of renaming FORMAT:

```
REN FORMAT.COM TRASH.COM
```

To channel someone who might want to use the FORMAT command legitimately, you can make a batch file so that keying in FORMAT will only act on your floppy drive. Once you have renamed FORMAT.COM, you can create a FORMAT.BAT file similar to this:

```
FORMAT.BAT
ECHO OFF
ECHO !!!!!!!!!!! DANGER !!!!!!!!!!!
ECHO You have started the program which will
ECHO prepare any floppy disk inserted in DRIVE A:
ECHO This will also erase any data on the floppy.
ECHO If you want to abort this program, depress
ECHO the CTRL and "C" keys. Otherwise
PAUSE
TRASH A:
```

You could substitute REM for ECHO. I've discovered that REM is shown on every line, while, if you have ECHO OFF in your BATCH file, ECHO will only show the information you want your BATCH file to type out, without typing the word ECHO. A bit more sophisticated in appearance, when you want to go that way. And, of course, the PAUSE command prints out "strike any key to continue", while halting things for a decision to be made.

You can do something similar for the DELETE and ERASE commands. These commands are not .COM or .EXE files, so you can't just rename them. However, you can go into COMMAND.COM where they reside. You will need to use DEBUG, or a similar program editor which will let you make modifications to a program file. In general, I have found that the DEL and ERASE in COMMAND.COM can be located towards the last 1/4 of the file. In user-friendly terms, you might alter these commands into something easy to remember, by changing the first vowels. DEL could be changed to DUL, and ERASE to URASE. Using DEBUG, Microsoft's version 3.2 of COMMAND.COM could be altered as follows:

```
DEBUG COMMAND.COM
-d 4EC0
-e 4EC4 55
-e 4ECE 55
-d 4EC0
-w
-q
```

Notice that the first thing I did in DEBUG was to use the "d" command to insure that I was in the right area. I suggest that you do this, too, rather than take the addresses provided above as gospel. If you don't find DEL and ERASE in this area, continue searching. If you have never used DEBUG before, please become familiar with it before attempting this. It isn't that hard to learn, and it's safer to do this if you know what you're doing. Before I stored off my altered COMMAND.COM, I confirmed that my changes were correct. I also made certain that I retained my re-

tail copy of COMMAND.COM. Not only was this an additional precaution in case my work with DEBUG didn't work. But, if you do want to return to using DEL and ERASE, it's easier to upload your old COMMAND.COM than to have to do more work with DEBUG.

To make sure that you have a safety net in case of errors, it would be an excellent idea to perform this program editing

on a copy of COMMAND.COM. For best results, copy COMMAND.COM onto a blank floppy, or maybe onto your RAM disk. To test your changes, call this version of COMMAND.COM into residence by typing in COMMAND. You may have to set your path equal to your current drive, so you don't call on your normal COMMAND.COM. Or, if your altered COMMAND.COM is on a floppy, you could boot up with that disk. Have some files you can safely delete. Delete one by typing DUL TEST1.DOC, or whatever your file may be that you want deleted. To test your reprogrammed erase command, type in URASE TEST2.DOC. Then, do a DIR to find out if the files were deleted. If your input commands got the phrase "bad command or filename", first check to see if you used the right name for the file you wanted to delete. Then, confirm your work with DEBUG.

You can also save yourself from malicious or accidental deletions by using a program which confirms each file you command it to delete, making you answer with a "y" or "n" for each filename it displays. Several bulletin boards and user's groups have copies of VDL and VDEL, verifying deletion programs which work in this manner. For best safety, obtain these programs from someone you trust. Besides protection from intentional sabotage, the last two steps can be especially helpful to you if you have an operation where most of the people have at most basic experience.

I had an occasion to briefly supervise a shop where my co-workers had been given only initial training with micros. They had DOS-compatible machines with 20 megabyte hard disks, but not even enough experience to use directories. One man used the old DEL *.* command, and erased everything on one machine. I spent 40 minutes using a file restoration program. Immediately after, I altered the DEL and ERASE commands, and required everyone to use VDEL. I only explained to

my successor what I had done, and left him to judge when to trust his workers to use DEL.

A final set of safety measures against sabotage programmers can be obtained by staying informed. Like thieves and criminals of any sort, sabotage programmers will keep changing their methods. Some of them will become aware of the methods outlined in this article, and will find ways around them. Others, like a car thief interviewed in a major automotive magazine a few years back, will count on most people disregarding or remaining ignorant of these precautions. For those who choose to stay informed and aware, magazines like this one will help.

Another resource is a good users' group. For those who may have positively never heard of users' groups, a users' group, or Special Interest Group (SIG), is essentially a computer club. In spite of bad publicity by some who think that this sort of club serves only to foster sabotage programmers and stealing of copyrighted programs, the primary purpose of most of these groups is to allow people interested in micros to get together, in order to share experience and information. Some groups are based around only one brand or type of computer. Others have people with several kinds of micros.

A few years ago, when I was just learning about small computers, I got to a point where I needed to increase my knowledge, and to have contact with as many people as possible who were familiar with micros. A co-worker suggested that I look into a users' group, one dedicated to the type of micro I was working with. The Air Force had just begun buying Z-100s, so he suggested that I try a Zenith group. He had heard of one in our area, but didn't know any members. So, he suggested that I try the logical approach: contact the dealer. I did, and he had the phone number of the group's president and vice-president. In short, I became a member, got elected as the program chairman (recruiter of guest speakers, not keeper of software), and tapped into a most valuable mother lode of information and guidance.

More information about users' groups would take a separate article. As far as protection against sabotage programs goes, users' groups can provide help in keeping track of bad programs by name. If you have a requirement for a good free utility you've heard of, a users' group is likely to have a safe, tested copy. While it is true that sabotaged programs can be passed on through users' groups, you're generally no more at risk with programs from a users' group than from commercial programs. Recent trade publications have chronicled incidents of commercial programs, doctored by temporary workers, which have damaged the hardware of paying customers.

Continued on Page 50

A Really Fast Word Processor For the Zenith 171

Souping Up PC-Write, Versions 2.6 and 2.71

Daveed Dov Shachar

P.O. Box 835
Dimona, Israel

Introduction

As many may remember from my last article, I own a Zenith Z-171, and I'm proud of it! I use it instead of a full-sized PC, as it runs all PC programs and takes up relatively little space in my crowded office. I know that there are many other Z-171 lovers out there. I also know that many of you, having made a decision to buy a slightly off-the-beaten-track computer, probably own some off-the-beaten-track software. I refer specifically to a word processor called PC-Write which, since it is Shareware, is not available in the computer stores. A complete copy of version 2.71 can be purchased from Micro-Com Systems for about \$6. If you like it, you can register with the publishers, Quicksoft. Versions 2.6 and 2.71 are both available from Quicksoft. One major advantage of registration, aside from telephone support and updates, is a copy of the full instruction manual, which is better than any book I have ever seen on PC-Write.

The publishers of PC-Write did not have the slightly eccentric Z-171 keyboard or screen in mind when they wrote version 2.6. This article will explain how to use the word processor with this keyboard and screen. The publishers never intended 2.71 to be used with a RAM disk, but I have more or less solved that problem, as well. Aside from the normal advantages of using a RAM disk, the built-in dictionary in 2.71 works very, very quickly.

The Z-171 screen is a problem which exists with any version of PC-Write. I understand that Zenith now has a supertwist screen for the Z-171. I also understand that it is quite expensive. There is a fairly easy way to get a brighter image on the screen while you are editing normal text.

Simply edit the PC-Write file `pr.def` so that the text will appear brighter on the screen. Make sure that the following line appears in your `ed.def` file:

```
!pr.def
```

Make sure that the following line does not appear in your `pr.def` file, or PC-Write will enter a closed, black-hole type loop:

```
!ed.def.
```

Now, using PC-Write, edit `pr.def` and add the following line:

```
&1:15.15.71
```

This will give you a much brighter screen. It will make it impossible to differentiate between marked and unmarked text, but you can solve that with the following line:

```
&5:7.7.23
```

This will make marked text pale, and regular text dark, reversing the existing situation, and making normal text much easier to read.

Version 2.6

Let's start with version 2.6. This version fits completely into a RAM disk, making it very, very fast. You can transfer between the print and edit programs extremely quickly, and you can shell to DOS instantaneously. I am assuming that your Z-171 is equipped, as mine is, with two floppies and 640k of RAM memory. First, prepare a PC-Write boot disk from the master disk. This should include the DOS hidden files, `command.com`, and the following PC-Write files: `pr.def`, `ed.hlp`, `pr.exe`, `ed.exe`, `ed.def`. The first file, `pr.def`, contains the printer driver for your printer. You choose this from the printer menu offered to you when you run `workdisk.exe`, which helps you set up your boot disk. The next file, `ed.hlp`, contains all the help menus, `pr.exe` contains the print program, `ed.exe` is the edit program, and `ed.def`

contains some important information for the editor.

Next, copy `vdisk.sys` from your MS-DOS diskette onto your PC-Write boot diskette. This will create the RAM disk for you. I am assuming that you are patriotically using Zenith's MSDOS, which will create a RAM disk called C. Some DOSes create a RAM disk called G. If your DOS does this, simply replace any line below containing C:, with G:. Now you will need a file called `config.sys`. DOS reads this even before it reads `autoexec.bat`.

PC-Write creates pure ASCII files, unless you add dot lines and other enhancements. Therefore, you can use PC-Write to create `autoexec.bat` or `config.sys` files. With the PC-Write boot disk in drive A, simply type `ed config.sys` at the DOS prompt, and you can type in the lines which should be in this file:

```
FILES = 12  
BUFFERS = 10  
DEVICE = VDISK.SYS 320 256 64
```

The first two commands are needed by PC-Write to handle some of its print commands. The third one creates a RAM disk of 320k. My `config.sys` file has a fourth line, `COUNTRY = 972`, which tells the computer that I want the date the way we write it here, Day/Month/Year.

The next file to create is the `autoexec.bat` file. This will call up a very important utility, `keybuf`, which provides 128 characters of typeahead buffer space, instead of the 12 characters provided by DOS. It is available on Utilities 1, also from Micro-Com systems. Make sure that a copy of it resides on your boot disk. Readers already familiar with PC-Write know that when you try to send the cursor more than a couple of spaces at a time on a Z-171, it freezes and the keyboard starts beeping at you. `Keybuf` solves this problem com-

pletely. Although the cursor will no longer stop on a dime, especially when you are using the up and down arrows to scan through text, the annoying beeps disappear and you have far more control over the cursor. The other advantage of keybuf is that you can type in several DOS commands at one time, without waiting for the computer to finish processing one before typing in the next. Assuming that you do not want to change the DOS prompt, the autoexec.bat file should read as follows:

```
keybuf
copy command.com c:
copy pr.exe c:
copy ed.exe c:
copy ed.hlp c:
copy ed.def c:
copy pr.def c:
c: set comspec = c:\command.com
ed
```

The second to last line means that when you hit F1 F4 (or Esc 1 4, as I will explain shortly) to shell out of PC-Write into DOS, the program will look for command.com on the C drive, speeding up things even more, and saving wear and tear on your disk drives.

This is probably the time to mention something obvious, but nevertheless very, very important. PC-Write will now try to save everything you write, on the C disk. Unfortunately, as soon as you turn the computer off, whatever was on the C disk is wiped out forever. There are ways of getting around this within PC-Write, but they are a little complicated. Therefore, remember: When you give PC-Write the name of the file you are writing, always preface the name with B:, unless you are editing something on the boot disk itself, in which case preface the name with A:. In other words, you will be editing "a:autoexec.bat", and "b:mythesis", but never merely "gonefore.ver".

The last line, ed, calls up PC-Write and lets you start editing.

So far, you have created a boot disk which creates a RAM disk, calls up PC-Write, and makes your cursor user-friendly. You have also made the screen easier to read. Now you are ready to edit PC-Write itself, to make it more Zenith-user friendly.

The first problem is searching for text with the so-called grey+ and grey- keys. On the Z-171, these are only accessible by going into number lock (and they're blue, instead of grey). Unfortunately, if you go into number lock and then try to use the blue plus and minus (located on the question mark and semicolon, respectively), you will be surprised at just how fast PC-Write will freeze up on you. This can be solved by redefining the accent key and tilde to act as the grey+ and grey- keys. Edit the a:ed.def file. Add the following lines to it:

```
096:334
126:330
```

A complete list of these keycodes and how to redefine any key to be any other key is in the full PC-Write instruction book.

You will now be able to use the accent key and the tilde to do almost anything you normally used to use the blue + and - keys for, such as accessing the next or previous ruler line. Unfortunately, this means giving up the use of the accent and tilde, but I will lay out a possible solution for that later on in this article, in the section on Using PC-Write for Foreign Languages.

In order to see if this is working properly, you will have to exit PC-Write, copy a:pr.def to c:, and then re-enter PCWrite. Do not attempt this from a shell (F1 F4). The reason for this is that any change made to either a pr.def or ed.def file only takes affect the next time you use the program. I learned this after many frustrating hours of wondering why changes I was making weren't having any affect on the program.

One thing you will not be able to do with the accent/tilde key, is use it to move to the end or the beginning of an entire file. PC-Write has a built-in solution, for this problem, however. While holding down the Alt, type either equals (=) or minus (-) at the top of the screen. This will jump you to the beginning or end of the file you are working on.

One irritating shortcoming of the Z-171 is the F keys. I try to use them as little as possible. PC-Write offers several solutions to this problem. First of all, almost all commands calling for F keys can be given with either the number keys at the top of the keyboard, or by using the letter key with the first letter of the command you wish to give. Additionally, you can access all the help menus via the Esc key, instead of with F1. For instance, suppose I want to save what I am writing right now. I can do this in any of four ways, the first of which is not recommended for the Z-171.

```
1: F1 F3
2: Esc 1 3
3: Esc s 3
3: Esc Return 3
```

The first method utilizes the F keys, which are not fun to use on this machine. The second utilizes the fact that you can usually replace the F keys with top-row number keys (except for one or two print commands). The third way utilizes the fact that many commands can be accessed with the letter keys, the "s" stands for "System/help", which I see on the top menu line after hitting "Esc". The fourth way utilizes the fact that many commands can be cursor to, and then hit Return to implement them. The advantage of this last system is that a single line at the bottom of the menu reminds you of what each command is for, but only if the command has been accessed by the cursor. For instance, the following will put a ruler

at the cursor:

```
Esc downarrow Return
```

This has the advantage that after you hit the down arrow, a line appears at the top of the screen to remind you what the ruler is used for. This line says:

```
RULER: split screen, edit margins, menu:
insert, default, to/from file, etc.
```

Personally, I find Esc 1 3 for saving, or Esc 1 2 for quitting, or Esc 1 6 for changing files, etc., extremely fast and easy to remember.

If you want to print an entire text without skipping any pages or stopping in the middle, Peter Klein of Quicksoft has a solution. I am quoting the Autumn 1987 edition of "Quick Notes, News about the PC-Write Word Processor by Quicksoft", which is sent free to registered users: "To print the file we're editing, we must press five keys. You can simplify the process by adding this line to ed.def:

```
356:315,321,335,32,47,101,264
```

"This makes Ctl F7 (key 356) act as if you had pressed F1, F7, and End (to get to the end of the current file name), added '/e' and hit Enter. You use '/e' instead of two F10 codes because function key codes can't be passed between the edit and print programs."

This works very fast with a RAM disk, and is reminiscent of other word processors, such as Microsoft Word, where a simple two-keystroke combination is sufficient to print a text.

Version 2.71

This section is not to be read instead of the section on version 2.6, but as a supplement to it. The only thing about 2.6 that isn't true about 2.71 is how to set up the RAM disk. First of all, if you happen to have a copy of 2.7, throw it out. It's full of nasty bugs, and not worth the effort. As version 2.71 can be copied from a friend or purchased for \$6 from MicroCom, it's just not worth the effort dealing with it. It was the first version to attempt the use of a speller, and the bugs were discovered and corrected almost as soon as it came out.

I like version 2.6. It's my word processor of choice. Normally, when I'm typing a letter or an article, I simply watch the screen. In my work, however, as a Hebrew/English translator, I have to look constantly at the text I'm translating from. This means that I can't catch my typos, and that's where 2.71 with its speller comes in. Version 2.71 is a huge program, and comes on two distribution disks. After literally weeks of experimentation, I managed to get the edit program to work, with the dictionary and all the help screens, on a RAM disk. The program takes up so much memory, however, that I had to make the RAM disk quite small, and it can't hold the print program. It also can't hold command.com, meaning that shelling is slower, as well. Therefore, I use

2.71 for translating, since the speller beeps at all my typos, but 2.6 for everything else.

First of all, let's see what the speller does. When it is on, it beeps at every mistake. If it beeps at a real word that you want to use, but that it doesn't happen to know, you can add that word to a file called words.use. Remember, however, that that file is going to be on the C (RAM) drive, and will disappear when you turn off the computer. Therefore, we will create the autoexec.bat file in such a way as to have the computer copy words.use to a real disk automatically when you exit PC-Write. Another problem with the speller is that you have to keep hitting F2 if you want to see other possibilities for the correct spelling of a word you have misspelled. I've set up my copy so that hitting "Ctl p g (for guess) will accomplish the same thing. This saves a lot of wear and tear on the not-so-useful F keys on the Z-171 keyboard.

Another problem in version 2.71 is that help is only read into memory the first time you call it up. Therefore, I've edited ed.def so that help comes up even before you start editing, and then immediately goes off the screen, ready at your beck and call.

I have redefined ed.def to the following: Redefine the grey (blue) + and - keys, print with Ctl F7, turn on the speller, call up help, and make the speller much easier to use. Here are my new ed.def commands, with a complete explanation of each one:

```
096:334
126:330
```

These two lines redefine the accent/tilde key, as I did for version 2.6.

```
P:555, Speller'';
556,099,361,321;
556,103,317
```

These three lines redefine Ctl F2 F7 to Ctl p c. This means that I can turn the speller on and off with Ctl p c, saving the use of two F keys. They also allow me to guess at a word by typing Alt F2, and then Ctl p g (for guess), each time I want to see another guess. This is instead of having to punch Ctl F3 for each guess, thereby saving the use of yet another F key.

You can use the accent/tilde key to scan a file for the next word unrecognized by the speller, instead of having to use grey (blue) +/-.

```
400:361,321,315,315,260
```

This line tells PC-Write what to do each time I begin editing (writing). It turns on the speller, loads help, and goes out of help into the edit mode. Since help is one of the files that wouldn't fit onto my RAM disk, this means that I have immediate access to the help screens while I'm editing, without having to wait for a floppy disk read.

The autoexec.bat file contains a major surprise. In spite of the fact that there

is no hard disk, a path= command is necessary. Without this command, PC-Write will not be able to find words.mas (the list of words used by the speller) and words.use (the user-created word list) on the RAM disk. Also, it will not be able to find pr.exe on the A disk. No matter what I have tried to do after I print, the program reads ed.exe from the A disk rather than from the C (RAM) disk. Evidently, PC-Write always looks for ed.exe in whatever directory it was in when it was printing. If any HUGgies out there can solve this, I would like to hear about it, as it would really speed things up, making version 2.71 closer in speed to 2.6.

Finally, in order to find room for everything, words.mas and words.use must stay on the b disk, which is the data disk. This cuts down the amount of data you can store on each disk by a third, but that's one of life's many little compromises.

And here, in all its glory, is my autoexec.bat:

```
path=c:\a:
keybuf
prompt $p --
copy ed.exe c:
copy ed.def c:
copy pr.def c:
copy b:words.mas c:
copy b:words.use c:
c:
ed
copy words.use b:
```

The config.sys file is different than the one for 2.6, as the RAM disk is considerably smaller:

```
files = 12
buffers = 10
device = vdisk.sys 211 256 64
```

From experimentation (much experimentation!), this is the largest RAM disk I can have if I want all the help screens to be accessible, as well. Unfortunately, since it won't hold the pr.exe program or command.com, both printing and shelling are much slower than with 2.6.

Using PC-Write for Foreign Languages

There are two conventional ways of accessing the foreign language characters from PC-Write (ASCII 128-168), other than using the KEYBxx command in DOS to change your keyboard. These are 1) Using number lock and then Alt and the keypad number keys to type in the ASCII number or 2) using the accent key, plus a regular key. The first method is both slow and clumsy, and it entails looking up the letters on an ASCII chart every time you want to use them. The second method is unavailable, as I put the accent key out of commission because of the grey + grey - compatibility problem. It should, therefore, come as no surprise to learn that I have chosen a non-conventional method of accessing the foreign language characters.

I recommend a separate PC-Write boot disk for each foreign language you

want to use. In my case, the only foreign language I use which is presently supported by PC-Write is German (the Hebrew version is evidently almost ready), and since German obviously doesn't work with the speller, I only had to modify version 2.6. This modification causes Ctl + any letter to be the German accented modification of that letter. For instance, Ctl a prints ä, Ctl A prints Ä, Ctl s prints ß, etc. For German, I needed umlauted vowels, small and capitalized, and the double ess. The commands which have to be added to ed.def in order to accomplish this are:

```
001:132
472:142
015:148
486:153
021:129
492:154
019:225
```

Although I mentioned this before, I cannot stress it too strongly. After making any changes to pr.def or ed.def, exit PC-Write completely and re-enter it in order for the changes to take effect.

PC-Write is an extremely versatile word processor, and can be edited to act like almost any other word processor. It is already set up to emulate most WordStar commands, if the user feels more comfortable with them, but it can be easily used with its own set of more simplified commands. The interesting variations available with PC-Write are, unfortunately, beyond the scope of this article, which was, as stated, to offer a method of turning PC-Write into a superfast word processor on the Zenith Z-171, given its slightly unusual screen and keyboard.

By the way, if any readers are interested, PC-Write now supports DCA-RfT conversion, available either directly from Quicksoft or on certain bulletin boards.

Companies mentioned by Daveed Shachar:

Quicksoft
219 First North #224
Seattle, WA 98109
(206) 282-0452

MicroCom Systems
P.O. Box 51657
Palo Alto, CA 94303
(415) 325-6500



**Are you reading
a borrowed copy of REMark?
Subscribe now!**

Turbo Pascal

Part 2

File I/O and More Commands

*Matt Elwood
1670 North Laddie Court
Beavercreek, Ohio 45432*

Welcome back for part two in a series that deals with programming in Turbo Pascal 4.0. First, we'll talk about file I/O, and then about some standard procedures and functions included with Turbo 4.0.

Our first subject will be file I/O. Turbo accesses files through a variable that you must define. The file can be one of three types: typed, text, and untyped.

The first file type, typed files, uses database-type records and are used for database-type applications. For example:

```
type InvRec = record
  Idnum : word;
  Desc : string[60];
  Unitprc : integer;
  Instock : word;
end;
```

```
Var InvFile: file of InvRec;
    InvItem: InvRec;
```

This would define a file variable using the record shown above. A record is a group of fields grouped together under one common name. Notice, the record name is defined in type, where you can define custom variables and records. These types can't be accessed as variables. Other variables must be defined using the record type. To access the individual fields in a variable defined as a record, you would type the field name after the variable defined as the record's name (in the above example "InvItem") separated by a period. For example:

```
InvItem.Desc:='Turbo Pascal 4.0';
InvItem.UnitPrC:=72.95;
```

An inventory file adding program is shown below. This can be run with the R command or compiled with the C command from the top line menu:

```
program InvAdd;
type InvRec = record
  Idnum : word;
  Desc : string[60];
  Unitprc : integer;
  Instock : word;
end;
Var InvFile:file of InvRec;
    InvItem:InvRec;
begin
Assign(InvFile,'INVENT.DAT');
Rewrite;
Write('What is the inventory ID number
(1-65535)?');
ReadLn(InvItem.Idnum);
WriteLn('Describe the item. ');
ReadLn(InvItem.Desc);
Write('What is the unit price? ');
ReadLn(InvItem.Unitprc);
Write('How many items are in stock? ');
ReadLn(InvItem.Instock);
Write(InvFile,InvItem);
end.
```

**Listing 1
Inventory Adding Program**

This program deletes the old file every time and the file will only have one record in it. It is very basic, and will only allow integer (without decimal points) prices. As you see here, the file variable is defined as a file of the record name, which is here "InvItem". All of the I/O functions shown above are discussed later.

Our second file type, text, consists of lines of characters (variable type char) with each line terminated with the MS-DOS end-of-line character. Text file variables are defined like this:

```
Var txtfile : text;
```

The text type may be assigned and opened like any other file after being defined. For input and output, "ReadLn" and "WriteLn" are used. Both skip to the next line of the file, with "WriteLn" writing an end-of-line marker to the file. Remember, if the file was opened with "Reset", you may only READ from the file, and if it was opened with "Rewrite" or "Append", you may only WRITE to the file. Explanations of these terms are covered later in this article.

Our third and last file type, untyped, is a "raw" file I/O used just to read and write characters, so it may be used on binary files. This uses "BlockRead" and "BlockWrite", which do the same as the normal write and read, but read or write blocks of 128 characters.

Before the file is opened, you must assign the variable to an external file through the Assign command. The variable is defined like any other variable, with the Var command. Here is an example defining a file variable "TxtFile" as a file of text type then Assigning it:

```
Var txtfile : text;
Assign(txtfile,'C:\MECC\README.DOC');
```

After it is assigned, the file must be opened with one of three commands. The first command is "Reset", which is used on an already existing file that you do NOT want to erase. This resets the file pointer to the first byte in the file. The second command, "Rewrite", destroys the file and prepares it for new input. Rewrite also creates a new file that doesn't exist. The last command, "Append",

moves the pointer to the end of an already existing file and prepares the file for additional input. Remember, "Reset" will only allow you to read and the other two, "Rewrite" and "Append", will only allow you to write.

When inputting and outputting to files, just use the normal "Read" and "Write" syntax, but include the file name before the data to be read or written. For example, if you want to write the variable "TxtLin" to the file "TxtFile", use:

```
WriteLn(TxtFile, TxtLin);
```

Remember, you must use BlockRead and BlockWrite for untyped files. The syntax is the same. Also, you only need to use Write to write the record when using typed files because the whole record is written.

You may do many other things to files that have been assigned, and the first is "Erase". To erase a file, it must be assigned a file variable using "Assign", but NOT opened. This example deletes the file "INVENT.DAT":

```
Var f:file;
begin
Assign(f, 'INVENT.DAT');
Erase(f);
end.
```

The file may first be opened with one of the three standard opening commands, then must be closed with the "Close" command. The syntax for the Close command follows:

```
Close(f);
```

The "f" variable is any file variable. "Close" writes all the records in the buffer to the disk file. All files, when closed, must be reassigned if they are to be reaccessed.

The standard I/O functions also include support for MS-DOS directories with the "Chdir", "Mkdir", "Getdir" and "Rmdir". The syntax for "Chdir", "Mkdir", and "Rmdir" are as follows:

```
Chdir('c:\');
Mkdir('c:\turbo\mc');
Rmdir('c:\turbo\mc');
```

The parameter may be a string variable containing the name of the directory. A special Turbo addition, GetDir, gets the name of the current directory for the drive specified in a number with 0 being the current drive, 1 being A, 2 being B, 3 being C, etc. Another parameter which must be a string variable contains the output of the command. Here's an example:

```
Var s: string[80];
begin
GetDir(0,s);
WriteLn('The current directory is ',s);
end;
```

In this example, s, which is a string variable, would hold the output. Constants would not be allowed here, and would produce an error statement.

Turbo has a file error trapping like BASIC, which uses the IOResult variable and a special command called a directive, which is a special statement which turns

off and on a feature in Turbo. The directive is the {\$I+} and {\$I-} directive. To use them, just put them in front of a command that might cause an I/O error. For example, this program checks to see if a file exists, and if it does, it doesn't overwrite it; but if it doesn't, it will create it:

```
Program IOChk;
var f:text;
begin
Assign(f, 'IOCHK.TST');
{$I-} Append(f); {$I+}
if IOResult<>0 then begin
Rewrite(f); end;
WriteLn(f, 'This is written to IOCHK.TST');
Close(f);
end.
```

When you run this program, it checks for the file IOCHK.TST with the "Append" command. If the file exists, it places the file into the append mode, which allows the program to write the line into the file. If it doesn't exist, instead of exiting and saying "File not Found", it produces an error code in IOResult. The "if" command compares IOResult to 0 (which means no errors) and creates a new file with "Rewrite". The IOResult codes are listed below:

- 0 - Successfully completed operation
- 100 - Disk read error
- 101 - Disk write error
- 102 - File not assigned
- 103 - File not open
- 104 - File not open for input
- 105 - File not open for output
- 106 - Invalid numeric format

Now I'll go through the error codes one by one. The first error code, number 100, is reported when you attempt to read past the end of the file. For example, it would be reported by "Read", when you tried to read 6 lines when you only have a 5 line file.

The second code, number 101, is reported when the disk becomes full. The third code, number 102, is reported when the file is not assigned. To fix this, just use the "Assign" call to assign the file a variable.

The fourth, fifth, and sixth codes have to do with the file not being opened. The fourth code is used when the file just isn't opened. The fifth is when the file is opened with Append or Rewrite, so the file can not be read. The last code is used when the file is opened with Reset and can't be written to. These are only reported when a statement tries to do the actual command.

The last function, number 106, is reported when you try to read a numeric variable from a text file and there is not a numeric variable of that type there.

That does it for the file commands, and as you see, they are simple to use, especially when creating databases. Now we move on to procedures and functions included with Turbo Pascal.

The first two procedures exit either from the entire program or the current

program block. "Halt" stops the program's execution and returns to DOS or the Turbo menus, depending if you compiled it or just ran it from the menus. This is used to exit the program in case of a fatal error, where the program can't go on. This is especially useful since there can only be one "end." in the program and this is the only way to quickly end the program. An example using "Halt" is:

```
Var f:file;
begin
Assign(f, 'PGM.OVR');
{$I-} Reset(f); {$I+}
if IOResult<>0 then begin
WriteLn('Fatal Error..');
Halt;
end;
Close(f);
end.
```

This would look for the file "PGM.OVR", and if the program doesn't find the file, it will write an error message and end. Notice you still need the "end;" at the end of the "if" program block. Our other exiting command "Exit", exits from the program block, whether a procedure or function. An example of this is:

```
var i:integer;
procedure TstOne;
begin
repeat
WriteLn('Enter a number -100 to 100:');
ReadLn(i);
if i<-50 then Exit;
until i>50;
end;
begin
TstOne;
end.
```

The above program will keep on asking for numbers until the number is greater than 50 or is less than negative 50. Notice the Exit is used with if, but it would exit the "procedure" program block, not the "if" program block.

The next group of functions and procedures are the Transfer functions. These functions change a variable of one type to another type. The first one of these is the function "Chr". This function, just like the function "CHR\$" in BASIC, converts a number from 0-255 to the character having that decimal number. For example:

```
var a,b,c: char;
begin
a:=Chr(97);
b:=Chr(98);
c:=Chr(99);
WriteLn(a,b,c);
end.
```

This example would write "abc" to the screen. We assigned character variables with the 97, 98, and 99 values which are equal to a, b, and c in the ASCII table. This can be useful when programming printers with hidden characters.

The next function in the Transfer group is "Round", which rounds a real number to a number of type "longint". All that round actually does is rounds a decimal number to the nearest whole number.

The last function is "Trunc", which again converts a real number to a "long-int" type number. "Trunc" is the same as "Round", except it always rounds down.

The next function group is all mathematical functions. I will both list them off for you and tell you how to use the 80x87 numeric coprocessor with them. The 80x87 is not included with the basic computer and needs to be bought separately if you do a lot of math, like trigonometry. Remember when we talked about the compiler directive "{ \$I }"? To enable the coprocessor, just use the directive "{ \$N+ }" and to disable it, just use "{ \$N- }". Here's a list of the functions:

Abs	Returns absolute value x:=Abs(-6); -x would equal 6
ArcTan	Returns the arctangent
Cos	Returns the cosine
Exp	Returns the exponential
Frac	Returns the fractional part
Int	Returns the natural logarithm
Ln	Returns the natural logarithm
Pi	Returns the value of pi
Sin	Returns the sine
Sqr	Returns the square
Sqrt	Returns the square root

All of them have the same syntax as the "Abs" example shown above. These are normally not used except in a mathematical program, so I just wanted you to know that they existed.

Our next group of procedures and functions is the string group. There are many useful tools to manipulate strings in this group, so all will be explained in detail.

Our first procedure, "Delete" deletes part of a string from a string. An example of "Delete" follows:

```
Var s:string[60];
begin
```

```
s:='This is a string';
Delete(s,6,5);
WriteLn(s);
end;
```

In this example, Delete would delete "is a" and print out "This string". Delete started deleting at the 6th character here and deleted 5 characters.

Our next procedure does the opposite of "Delete". It inserts a string into another string. An example of this procedure is:

```
Var s : string[60];
begin
s:='Turbo 4.0';
Insert('Pascal ',s,6);
end.
```

The resulting string would be "Turbo Pascal 4.0". Insert would insert "Pascal" at the 7th position into s. The number must be the character you want to insert the string before. Our last two procedures, "Str" and "Val" convert numeric to string and string to numeric variables. "Str" converts numeric values to strings, and "Val" converts strings to numeric values. The syntax is about the same for both commands, except the variables are reversed, and "Val" contains the "code" variable at the end, which contains the index number of the error. Short examples are shown below:

To convert the integer x to the string

```
s:
  Str(x,s);
```

To convert the string s to the integer

```
x:
  Val(x,s,cod);
```

With "Var", the variable "code" contains the index (string offset) number if there is an error in processing the string, such as if an alphabetic character is in the string.

Now here are the four string functions. Our first function is "Concat". This function adds all the strings together into one string. For example, this line concatenates the sequence of strings "Turbo", "Pascal", "4.0" into one big variable, "s".

```
s:=Concat('Turbo ', Pascal ', '4.0');
```

Our next function, Copy, works like MID\$ in BASIC. It copies part of a string into another string. This can be controlled with two numeric variables, index and count. Index specifies the beginning character of the substring and count specifies how many characters would be copied.

```
Var s,t : string;
begin
s := 'Turbo Pascal 4.0';
t := Copy(s,7,6);
WriteLn(t);
end.
```

This would print out "Pascal". Here, Copy would get the part of 's' beginning with character index number 7 and being 6 characters long. Our second to last function, "Length", returns the length of the string. An example would be:

```
Var s:string;
begin
s:='Turbo Pascal 4.0';
WriteLn(Length(s)); {Wouldpbep16}
end;
```

Our final function, "Pos", searches for a substring you specify in a larger string. "Pos" returns the index number (the offset) that the substring you're looking for begins in. One final example:

```
Var s:string;
begin
s:='Turbo Pascal 4.0';
WriteLn(Pos('Pascal',s));
  {Would print out 7}
```

That's it for the second article. The next article we'll start getting into the neat features of Turbo, beginning with units and looking at some of the units included with Turbo Pascal 4.0. *

Continued from Page 16

```
while (((cont=getch())=='Y') || ((cont=getch())=='y'))
```

While you wouldn't get any error messages if you compiled or ran this, it wouldn't work as planned. The two getch() functions require two separate inputs from the keyboard, not one.

Okay. That's really a lot for your first time with Turbo C. So let's exit Turbo and get back into the DOS. If you are in the edit window, press F10. Then press F Q to quit the editor. If you changed your program since saving it you'll be prompted to save it at this point.

On your own, try writing some programs. Make them short and simple at first, just using a sequence of lines. As your confidence grows so does your programming abilities.

LISTING 8

```
main()
{
  float f,c;
  char cont;

  puts("Do you want to convert a temperature, Y or N?");

  while ((cont=getch())=='Y')
  {
    getch();
    puts("Enter a fahrenheit temperature");
    scanf("%f",&f);
    c = (f-32.0)/(5.0/9.0);
    printf("\n%.2f degrees fahrenheit is %.2f centigrade\n",
f,c);
    puts("Do you want to convert a temperature, Y or N?");
  }
}
```

*

ENABLE

Part 12

A Tutorial Spreadsheet

George P. Elwood
1670 N. Laddie Court
Beavercreek, OH 45432

In this, the 12th part in a series on ENABLE, an integrated software package developed by the Software Group, another part of the spreadsheet will be discussed. This article is the follow-on to part nine and covers the basic graphics capability of ENABLE which is available in both the Z-100 and the PC versions. The advanced graphics capability, Perspective, is only available in the PC version.

These series have resulted in several letters from fellow HUGgers. Last month, I received letters from Alaska and the Federal Republic of Germany. REMark is an International magazine as are Zenith computer systems. I enjoy hearing from HUGgers and will respond to questions on ENABLE. I can also work my way through Z-100 questions as that is my favorite computer. Other questions I will direct to my son, Matt, as he is the other "expert" in the house.

Enable version 3.0 should be out by the time you read this. It should have many new features, some of which are available on version 2.15. One of the features of version 2.15 is password protection of spreadsheets. When you develop a spreadsheet, this option is made available. You must remember the password as there is no way to decode the file.

In part nine of this series, I started the discussion of ENABLE's graphics capability. In this article, I will complete that

discussion by covering line, XY and Hi-Lo graphics. Again, I will use data from a business plan I developed last year while trying to buy a railroad. Did you ever try to find \$3 million dollars? I was able to raise \$2 million with the plan.

The line graph is probably the first graph type that you made. This is the type of graph that is made in grade or high school. It basically plots data points along the x and y axis with a line between the points.

Entering the ENABLE spreadsheet, we will create all of the data that is needed for the graph. This graph will plot the projected income and expenses during year two of the operation of the railroad. You must include the X-axis labels if you want them displayed on the graph. After the data is input, press F10 or the "slash", which is the same as LOTUS, to display the spreadsheet top line menu. From this menu, select "Graph" by either highlighting the word using the cursor keys and <RETURN> or by pressing "G". ENABLE will permit this type of selection from menus throughout the program. You select "Create" from the displayed menu options. Once selected, you are prompted for a name of the graph. I typed in "INCEXP" as the name. Remember, you can have up to eight graphs and their settings in ENABLE for one spreadsheet.

After typing in the name, press <RE-

TURN> twice to get to the "Select Options" menu. Press "1" or highlight the "1" using the cursor keys. After this is displayed, press <RETURN> to accept the "Data" options. You can now move the highlighted cell over the cell that contains the January income data. Then press "." to lock the cell down as the first cell and then press the down arrow key to highlight the entire column of income data, pressing the <RETURN> in the December cell. Next, move to the "Legend" option and type in "INCOME".

Press {ESC} to return to the basic options menu, and then select "2" to enter the expense data. Using the same procedure, select the cells that contain the expenses data and type in "EXPENSE" as the legend. Press {ESC} to return to the basic menu and {ESC} again to move up to the "Select" menu and press (D)isplay to show the chart. It will be displayed as a flat bar chart which is the default setting in ENABLE. The legend will be displayed, but no heading or axis information is shown. Press {ESC} and move down the menus until you get to the "Select INCEXP Options Global" menu and press <RETURN> to accept the TYPE option. Here select "Line". Press {ESC} to return to the menu and move to "Axis" option and select "Y-axis-format", "\$" and "0" for the decimal places. From the same menu select "1-X-axis-data". You will be

returned to the spreadsheet where you will select the columns that contain the month data using the procedures outlined above. Next add the Heading, Sub-heading, x-axis and y-axis legends to the graph. Select Fonts if you desire, I normally use Roman for the heading.

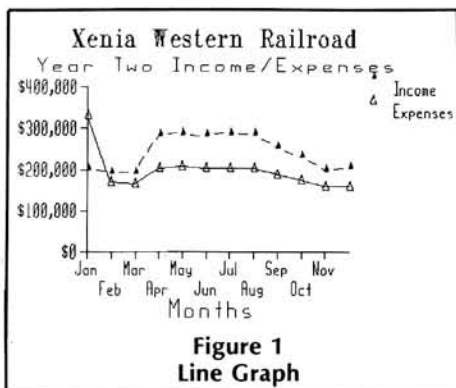


Figure 1
Line Graph

Once you have all of the information for the graph input, go back out to the "Select" menu using the {ESC} key. Select (D)isplay and look at your graph. Note that the graph has displayed the legend on the right side of the screen with each line on the chart a different type. The data points are shown as changes in the direction of the line. No symbols are displayed with the default line graph.

If you wish to add symbols to further define the data points, return to the "Global" options menu and "XYL-format". When this is selected, you can select "Lines", "Symbols", or "Both." If "Symbols" is selected, there are eight different symbols available, one for each data group. You can not change the type of symbol. If "Symbols" is selected, only symbols are displayed. If "Both" is selected, then both lines and symbols are displayed.

If you have a color monitor, the graph can be displayed in color. These colors can be selected by the user from the "Create Options" menu for each data point.

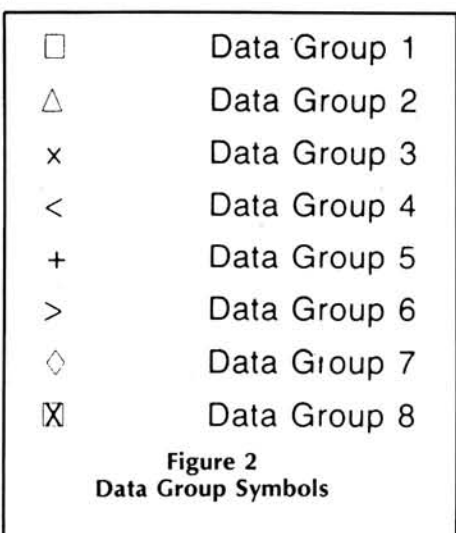


Figure 2
Data Group Symbols

The next of the basic charts in ENABLE is the High-Low-Close chart. This chart is the one used to display stock market information, usually the daily market trends. For our example, a monthly report will be completed showing the overall performance of an example stock. The graph will display the HIGH quotation for the month, the LOW quotation, the OPENING bid for the stock and the CLOSING bid.

To create this graph a spreadsheet must be completed with the data aligned in columns by month. The order does not matter, as you can select them when making the graph. After inputting the data, press F10 to display the Top Line menu and select (G)raph. Create a new graph of your choice. Move down to the Create OPTIONS and select the data. The number one data element will be HIGH for the chart. After selecting the data, select LEGEND and type in HIGH pressing {ESC} to back out one level. The next group will be "2" and will consist of the LOW data elements. Type in LOW as the legend. The third data element will be close data for the stock. Type in CLOSE as the legend. The fourth and last element is the OPEN quotation. Again, type in OPEN as the legend. You can select these data elements in any order as long as "1" is the HIGH, "2" is the LOW, "3" is the CLOSE and "4" is the OPEN quotations.

After selecting the data elements, press {ESC} to back out so that you can move down to the global type selection. From this menu, select "HLC" as the type graph. You can then move down the Global menu to input the heading, X and Y axis labels and select the font of your choice. Press {ESC} to return to the OPTIONS menu where you can select DISPLAY. ENABLE will display the graph with short lines displayed between the HIGH and LOW points. The HIGH and LOW point will be displayed as a dash (-) at the correct value location. The CLOSE value will be displayed as an "X", while the

LOW value a "<". ENABLE will scale the y-axis to provide the most usable information with the x-axis being the lowest LOW if it is an even number, or the next lowest whole number below the lowest LOW value. If you wish to plot additional information, like last year's values, these can be added. These added values need to be data elements five through eight. They will be displayed as symbols on the graph.

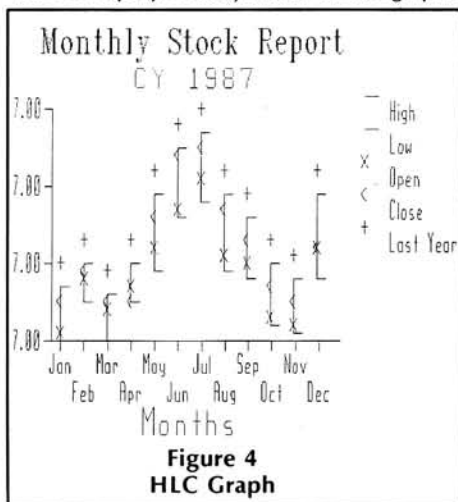


Figure 4
HLC Graph

The last of the basic charts is the XY graph. Unlike the Line graph, the XY graph is a two dimensional graph. Each point on the graph corresponds to two different cells. The x-data's values are input in "X-AXIS-DATA" on the Global menu. The y-axis data is selected from the normal OPTIONS "1" menu. The x-data values should be sorted so that they are in ascending order. In the example, the number of passengers per trip per month is shown. This information would be used to determine a break even point and show how you are doing. I have included the break even point for the railroad as determined from another graph. The graph shows that the number of trips and passengers carried will be not meet the break even point during one month of the year. The number of trips is displayed on

Monthly Stock Report						
	High	Low	Close	Open	Last Year	High
Jan	34	27	28	32	37	
Feb	37	32	35	36	40	
Mar	33	27	31	32	36	
Apr	37	32	34	32	40	
May	46	36	39	43	49	
Jun	52	43	44	51	55	
Jul	54	45	48	52	57	
Aug	46	36	38	44	49	
Sep	43	35	37	40	46	
Oct	37	29	30	34	40	
Nov	35	28	29	32	38	
Dec	46	35	39	39	49	

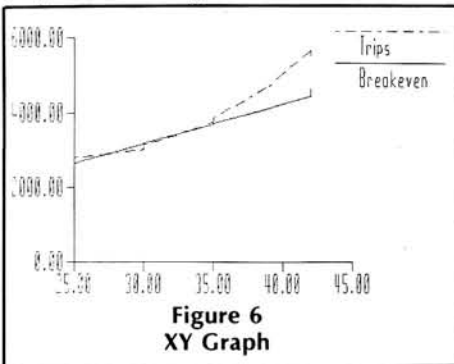
Figure 3
HLC Spreadsheet

the bottom, while the number of passengers is shown on the y-axis. I added a break even line based on projected need as a reference point.

Dinner Train Loadings			
	Trips	Passengers	
Nov	25	2628	2650
Jan	25	2700	2650
Dec	25	2808	2650
Mar	30	3024	3180
Feb	30	3154	3210
Oct	35	3780	3710
Apr	35	3881	3725
May	39	4717	4134
Sep	40	5069	4240
Jul	42	5685	4445
Jun	42	5564	4452
Aug	42	5504	4654

Figure 5
XY Graph Data

ENABLE will normally select automatic scaling based on the values of the y-axis so that the data will fit. However, ENABLE permits the user to select y-axis scaling, if desired. If you select the scaling options, you must specify the lowest and highest number and the increment between these numbers. You would use this option to display numbers that will display correctly. An example would be very large numbers where ENABLE will show all values from 0 to the highest value required. If the numbers are large and the difference is not great, when displayed, they will look the same. By using the ENABLE scaling feature, you can set the lower and upper limits so that the information displayed is meaningful.



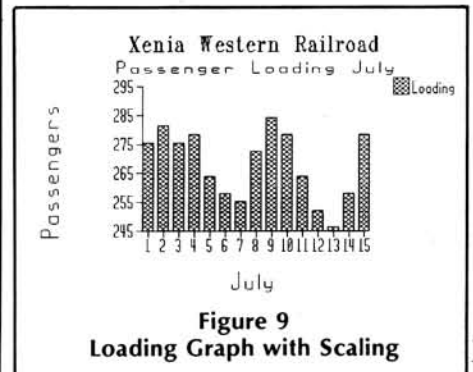
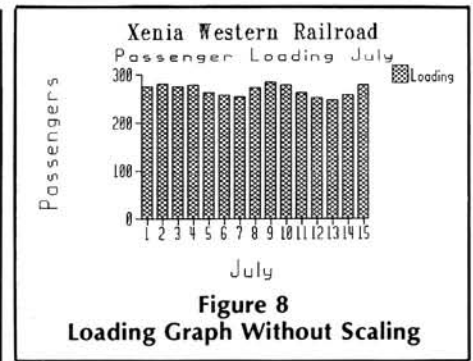
To use the scaling feature, first create your data and enter the graph function of ENABLE by pressing "F10 (G)raph". Create a new graph and select the data elements as previously discussed. The data for this example will be the dinner train passenger loadings for the first half of July. Note in Figure XX that the bar shows very little vertical difference in the normal graph. To select the scaling option, move to the "Create LOADING Options Global Scale" options in the graph. When selected, ENABLE will display "Automatic" as the selected option. Select "User Specified" and press <RETURN>. ENABLE will prompt to "ENTER THE LOWEST Y VALUE TO BE GRAPHED". In our example, 245

was selected. ENABLE will then prompt to "ENTER THE HIGHEST Y VALUE TO BE GRAPHED". In our example, type in 295. The final scaling prompt is to "ENTER THE INCREMENT FOR Y-AXIS SCALE MARKS". In our example, 10 was input. If you use too small of an increment, they will be displayed on top of each other and become hard to read. Return to the display option and display the graph and compare it to the first example. Scaling can be used with standard and 3-D bars, XY and Line graphs. This capability will permit you to display information in a manner that will enhance the image and thought that you are trying to convey to the viewer.

Dinner Train Loadings by Day	
July 1988	
1	276
2	281
3	276
4	278
5	264
6	258
7	255
8	273
9	284
10	278
11	264
12	252
13	247
14	258
15	278

Figure 7
Loading Spreadsheet

Another feature of the ENABLE basic graphing capability is proportional scaling. Unlike the scaling discussed above, this type permits the display of wide numerical values in a meaningful graph. Proportional scaling permits you, the user, to select a scale for a data element that is out of line with the rest of the data elements. Without this capability, the one high value will overpower the rest of the data elements and make them non-meaningful. In our example, the first two values are multiplied by 10 to increase their value so that they are in the same range. To show this capability, create the data that will be used. Select (G)raph and create the graph, "PROP" in our example. The data we will be using is the freight loadings for a four month period. During this period, the grain elevators on the line start shipping unit trains of covered hoppers. When displayed normally, the first two months without the shipments are fairly low. The next two months show increases of 10 fold. To make the bars near equal, select "Create PROP Options Group (S)cale" for the first two months and type in "10" as the multiplying factor. To make this fact known to the viewer of the chart, type in "10X" in the LEGEND for both data elements. When you display the chart the



first two months with the small numbers will have bars that are similar in height to the second set of two months. Note that the legend with "10X" is displayed on the right side of the chart.

Using this procedure you can increase or decrease the values of the selected data element by your selection of scale. The value can be a number above one (i.e., 10) to increase the value or below one (i.e., .1) to decrease the value. Another way to use proportional values is to select the Global Scale option. If you have a very large number (i.e., above a million), and you try to display this on the y-axis, it will be displayed as asterisks due to lack of space. By selecting Global Scale and inserting .0001 as the multiplying factor, the number will be displayed as 100. This scaling factor could be explained using the y-axis heading.

ENABLE's expert commands are available in the spreadsheet. These commands are listed on the plastic commands overlay that comes with the software. ENABLE expert commands are like the rest of ENABLE, once you learn a command, it is carried forward to the other modules. The function keys are used to call these commands. F1 is the HELP key and can be called from any module. It provides a screen of helpful information on the subject that you are in. F0 F1 (ALT/F1 for the PC) will cause the function key chart to be displayed at the top of the screen. F2 will prompt for a cell to go to. F0 F2 (ALT/F2 for the PC version) will print the spreadsheet using the default settings. F3 will insert a row above the cursor position. If you press this key, ENABLE will prompt for the number of rows to be added. F3 is

```

select SCALE Options Global Scale user specified
Enter the LOWEST Y value to be graphed : 245

```

	C	D	E	F	G	H	I	J
21					290			
22					Dinner Train Loadings by Day			
23	Passengers				July 1988			
24								
25	2628	2650	0.95	1	276			
26	2700	2650	0.97	2	281			
27	2808	2650	0.95	3	276			
28	3024	3180	0.96	4	278			
29	3154	3210	0.91	5	264			
30	3780	3710	0.89	6	258			
31	3881	3725	0.88	7	255			
32	4717	4134	0.94	8	273			
33	5069	4240	0.98	9	284			
34	5685	4445	0.96	10	278			
35	5564	4452	0.91	11	264			
36	5504	4654	0.87	12	252			
37			0.85	13	247			
38			0.89	14	258			
39			0.96	15	278			
40								

#1 A:\INCEXP.SSF Cap K56

Figure 10
Scale Selection Options from Menu

Freight Car Loading

Jan	300
Feb	31
Mar	31
Apr	41
May	37
Jun	27
Jul	24
Aug	38
Sep	47
Oct	500
Nov	750
Dec	450

Figure 11
Spreadsheet Data

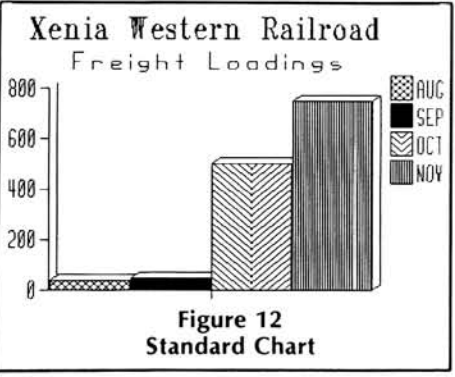


Figure 12
Standard Chart

also used to insert rows in the word processor, and as you see, this carries through. F0 F3 (ALT/F3 for the PC) will delete a row. F4 will permit you to edit a cell. F0 F4 will center the contents in the cell. F5 will force a recalculation of the spreadsheet. F0 F5 will permit copying between windows, which we will talk about later. F6 will permit you to move between the two halves of a split screen. F0 F6 will cause ENABLE to skip over protected cells. F7 makes relative cell references ab-

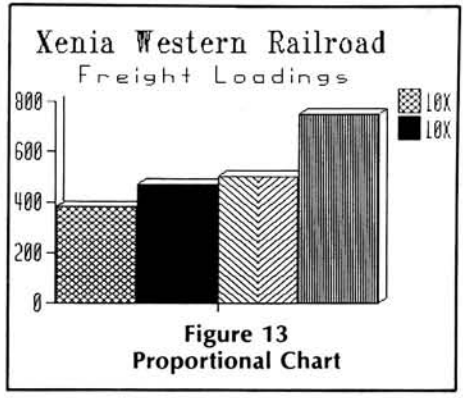


Figure 13
Proportional Chart

solute while pointing to formulas for copying. F0 F7 will move you directly from the spreadsheet to the graphics menu without having to go to the top line menu. F8 will copy cells directly rather than having to go to the Top Line spreadsheet menu and select Worksheet Copy. F0 F8 will move the contents of the selected cell to a new location. F9 is the expert key and is used with other keys to perform certain functions without having to go to the Top Line menu. F0 F9 will run macros selected with the command. F10 or "/" will bring up the Top Line menu. F0 F10 will save the spreadsheet.

- Spreadsheet Help Message Index
- For information on any of the topics below, press F10 and a letter from the list below. To view a complete list of expert commands and cursor control commands, see Enable's Quick Reference Guide. (Press ESC to return to your work.)
- a. Alignment of Data
 - b. Combining Spreadsheets
 - c. Copying and Moving Cell Entries
 - d. Global Commands
 - e. Graphics Help Index
 - f. Macros
 - g. MCM (Master Control Module)
 - h. Naming a File
 - i. Number Formats
 - j. Pointing Out a Range or Formula
 - k. Printing a Spreadsheet File
 - l. Quitting Your Work
 - m. Range Commands
 - n. Saving Your Work
 - o. Splitting a Spreadsheet Window
 - p. Using DBQ Data in Spreadsheet
 - q. Worksheet Commands

Figure 14
Spreadsheet HELP Screen

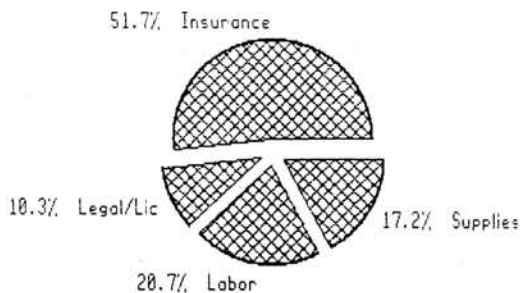
While working on the business plan that has been used in other examples, I needed to develop a very short summary of the plan that could lay out the main points. I felt that graphics within the text

was the best way to present the data as a picture is worth a thousand words. ENABLE showed its copy capabilities to the fullest during this exercise. By creating the document and then deciding where the graphs were to go and then moving them, it was easily completed. The data for the graphics was completed by copying data between the spreadsheets to one working sheet. The graphs were completed. Using the F0 Up arrow (ALT/Up arrow on the PC version) to move between the windows, a move from the graph to the word processing document is made. Place the cursor where you want the graph to appear and press F0 F5 (ALT/F5 for the PC) and the interwindow copy menu is displayed. Select the graph from the windows available and press <RETURN>. The graph will be displayed with the information that Copy is pending. By pressing F0 F5 again, the graph will be moved to the word processing document. On the Z-100, or the PC not in graphs mode, it will be displayed as a series of blocks in the screen. By pressing F10 MCM (3) Screen and (4) Graphics, the graph will be displayed. This graph can be printed at the same time as the word processing document. I added several graphs in the middle of the document which saved a lot of cut and paste. You can use the same procedure to copy several graphs onto the same page. See the example. Using the same procedure, selected cells from the spreadsheet can also be copied into a document. First position the cursor where you want the spreadsheet cells to be placed. Then press F0 F5 which will display the Interwindow Copy Options menu, select the spreadsheet and press <RETURN>. The ENABLE default option is to copy all of the spreadsheet, but you can move the cursor to the first cell you wish to copy and press the "." to lock the cell and then highlight all cells you want to copy. When you press <RETURN> all of the highlighted cells will be copied to the word processing document where you had the cursor positioned. You can now modify the data by moving, deleting or using any other word processing function.

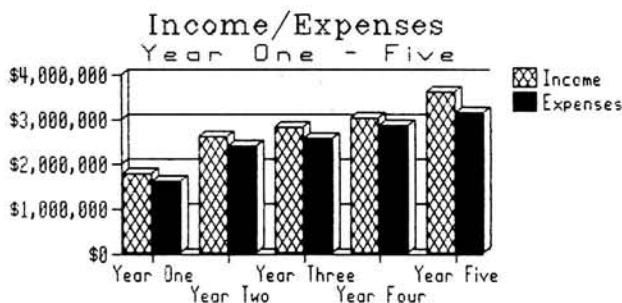
You can also use the F0 F5 copy command to move data between spreadsheets. When you are in the spreadsheet into which you want to copy data, press F0 F5 to open the Interwindow Copy Option. Select the other spreadsheet and press <RETURN>. Like the word processing copy, the default option is to copy it all. Once you have selected the data to copy, press <RETURN>. ENABLE will prompt for the starting cell for the imported data. Once selected, it will move it over by pressing <RETURN> again.

With ENABLE version 2.15 for the PC, which is a special version made for the DoD, it is possible to password protect

Non-Capital Expenses



Based on the five year financial plan and not counting startup costs, the line is projected to make a small profit the first year of operation. The projections show the profit ratio increasing every year with startup cost recovered by year five. Yearly expenses include a dividend payment to all stock holders in the company.



The loan requested is a fixed rate 10 percent 15 year amortized loan. The loan will be secured by the value of the line and associated equipment. The loan from commercial banks will be for \$1,000,000. A low interest state backed loan for \$600,000 to \$750,000 will be sought. The remainder of the financing will be sought from investors, the dinner train operator and freight shippers in the form of a private stock offering for \$1,600,000.

Paperwork for incorporation is being prepared and will be submitted as soon as possible. The company will consist of the Xenia Western Company, the basic structure, and the operating companies, the Xenia Western Railroad Company and the Phoenix Dinner Train. The company will be headed

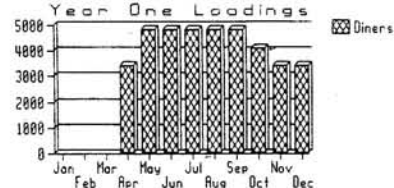
Figure 15
Sample Page with Text and Graph

your spreadsheet. Version 2.15 has just been released and version 3.0 is coming soon, so I would expect this feature to be carried forward into the new version. To use this feature, you must complete the spreadsheet and any other work. You can go back and work on it, but the basic file must exist. After saving the file and returning to the main menu, press (M)CM, (F)ile, and (S)preadsheet. All spreadsheet files on the data disk as defined in the setup will be displayed. Using the cursor keys, highlight the spreadsheet file you wish to protect and press (P) Assign Password. ENABLE will prompt for the password at that time. After the password is assigned, everytime you request the file, ENABLE will prompt for the password before it will display the file.

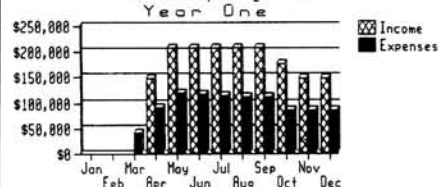
Like LOTUS 1-2-3, ENABLE supports macros to assist its performing routine repeating tasks. Unlike LOTUS, ENABLE can be taught the macro procedure rather than

having to type in the complete procedure in code in one cell. ENABLE will also permit macro development in the same manner as LOTUS. Using the LOTUS method, the macro must be placed inside a cell or range. ENABLE will process this macro only when the spreadsheet containing the macro is in the active window. To start this procedure, select the Top Line menu by pressing F10 or pressing the Slash key (/). Select (W)orksheet, (R)ange, (N)ame, and (C)reate. Each macro must have a number letter or number name in order to use it. The name consists of the Backslash (\) and the selected single character selected as the name. You must also include the range that will contain the macro. The macro must be typed into cells below one another. ENABLE uses commands similar to LOTUS, but has additional ones to use the increased capabilities of the program. Once you have created the macro in the cell(s), you can run it by typ-

Phoenix Dinner Train



DT Income/Expenses



Phoenix DT Income



Figure 15A
Sample Page with Several Graphs

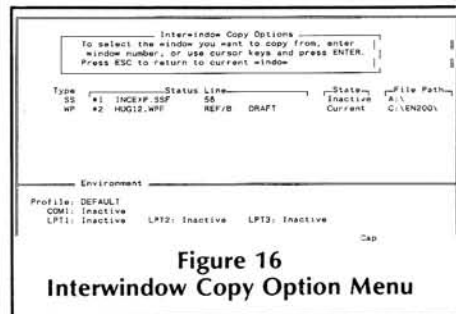


Figure 16
Interwindow Copy Option Menu

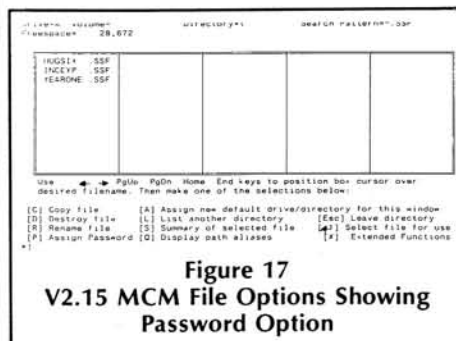


Figure 17
V2.15 MCM File Options Showing Password Option

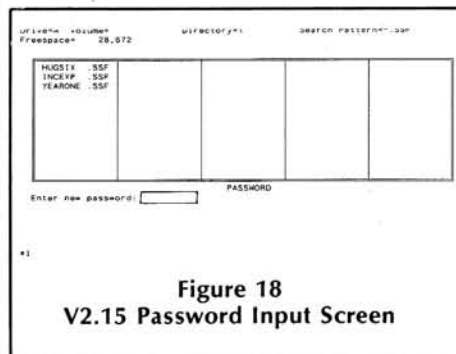


Figure 18
V2.15 Password Input Screen

ing SHIFT/F9 and the letter that you selected. In the example below, ENABLE will go to cell A1 {HOME}, open the Top Line menu {F10}, select (g)raphics, (s)elect the second chart in the list and then display it.

**Figure 19
Basic Macro**

```
{home}{f10}gs{right}~d
```

The available macro commands available in ENABLE are listed below. Note that they are similar to the same codes in LOTUS 1-2-3.

**Figure 20
Codes for Macro Keys**

Key	Code
F1	{F1}
F2	{F2}
F3-F10	{F3}-{F10}
ESCAPE	{ESC}
TAB	{Tab}
HOME	{HOME}
Up arrow	{UP}
Down arrow	{Down}
Right arrow	{Right}
Left arrow	{Left}
PgUp	{PgUp}
PgDn	{PgDn}
END	{End}
INS	{Ins}
DEL	{Del}
Backspace	{BS}
ENTER or RETURN	~
CTRL/ENTER	{LF}
{	{}
}	}
~	~
ALT/x	{&x}
CTRL/x	{^x}
SHIFT/x	{!x}
ALT/Up arrow	{PrevW}
ALT/Down arrow	{NextW}
ALT/END	{CloseW}
ALT/Home	{OpenW}

In addition to the above listed macro codes, ENABLE has the following commands that can be used with the spreadsheet. The "/X" precedes the special functions that you wish to use and must be the first entry in the cell. Remember that ENABLE will process macro commands from top to bottom in the defined

range. Because the "/" is the first character in the cell, ENABLE will try to call up its menu, so you must first place an alignment character (^, < or >) in the cell.

From the above entries, the capabilities of ENABLE's macros are shown. The pause and wait for inputs allow you to build a macro to run the program and stop and wait for the input. By including a menu, the person could make a decision based on the input data and ENABLE would continue to process the spreadsheet based on reply.

Not only will ENABLE use its own macros, it can also use the macros developed in LOTUS. ENABLE will translate LOTUS macros that it supports so that they can be run. ENABLE will highlight cells containing invalid LOTUS macro commands or commands that require cells that are outside of the ENABLE spreadsheet dimensions. ENABLE will also highlight a cell that contains a macro with a syntax error it discovers during execution. You must correct these error statements before the macro can be run. LOTUS commands requiring graphing or printing will not work. I discovered this while trying to run a spreadsheet created in LOTUS that required printing. There are several other commands that will not work or require modification before they will work. These are listed on page 104 of the ENABLE spreadsheet manual.

This completes the 12th article in this series. The next article will cover the database and will complete the DoD computer order program that was started in earlier articles. The next spreadsheet article will cover the PC graphics program, the 3D Perspective package. Both of the versions released with ENABLE 2.0 and Perspective II, the full color version of the package, which is available as an add on program from the Software Group, will be covered. Until later . . . *

**Are you reading
a borrowed copy of REMark?
Subscribe now!**

Command	Function
/Xlcondition~action	If-then decision
/XGcell	Go to cell x for further macro keystrokes.
/XCcell	Execute a subroutine until encountering an XR.
/XR	Return from a subroutine
/XLmessage~cell~	Pause, display message in indicated cell and wait for label entry.
/XNmessage~cell~	Pause and wait for number input from keyboard.
/XMcell~	Pause and process a user defined menu.
/XQ	Quit macro execution.

Prime H/Z Enhancements!

Clock Uses no Slot

FBE SmartWatch: On-line date/time. Installs under BIOS/Monitor ROM. Ten year battery. Software included. Works with all Heath/Zenith MSDOS computers. For PC's \$35; Z-100 \$36.50. Module \$27.50

H/Z-148 Expansions

ZEX-148: Adds 1-1/2 card slots. \$79.95. ZEX-148 + SmartWatch \$109.95
ZP-148: PAL chip expands existing 640K memory to 704K. \$19.95

H/Z-150 Stuff (Not for '157, '158 or '159)

VCE-150: Eliminate video card. Install EGA/VGA card. All plug in. Includes VEM-150, RM-150. Requires SRAM chip. VCE-150 \$39.95, SRAM Chip \$15
VEM-150: Card combines existing two BIOS ROM's into one socket. \$34.95
RM-150: Decoder PROM used in removing video card. With detailed instructions. \$9.95

ZP640 PLUS: Expand to 640K/704K by adding 2 banks of 256K RAM chips (not included). ZP640 PLUS \$19.95 (first one); \$9.50 thereafter.

LIM 150: 640K RAM plus 512K of simulated Lotus/Intel/Microsoft EMS v3.2 expanded memory. Installs on H/Z-150/160 memory card. No soldering. Requires forty-five 256K RAM chips (not included). LIM150 \$39.95

Mega RAM-150: Get 640K/704K main memory plus 512K RAM disk on H/Z-150/160 memory card. No soldering. Without RAM chips \$39.95

COM3: Change existing COM2 port address. Internal MODEM at COM2. Don't lose serial port. COM3 \$29.95

Maximize Your Z-100

ZMF100A: Put 256K RAM chips on "old" motherboard (p/n 181-4917 or less). Expand to 768K. No soldering. Without RAM chips. \$65.00

ZRAM-205: Put 256K RAM chips on Z-205 board. Get 256K memory plus 768K RAM disk. Contact us for data sheet before ordering. Without RAM chips. \$49.00

Z-171 Memory Expansion

MegaRAM-171: Put 256K RAM chips on memory card. Get 640K memory plus 384K RAM disk. \$59.95

H/Z-89 Corner

H89PIP: Two port parallel printer interface card. With software.

H89PIP \$50.00; Cable \$24.00

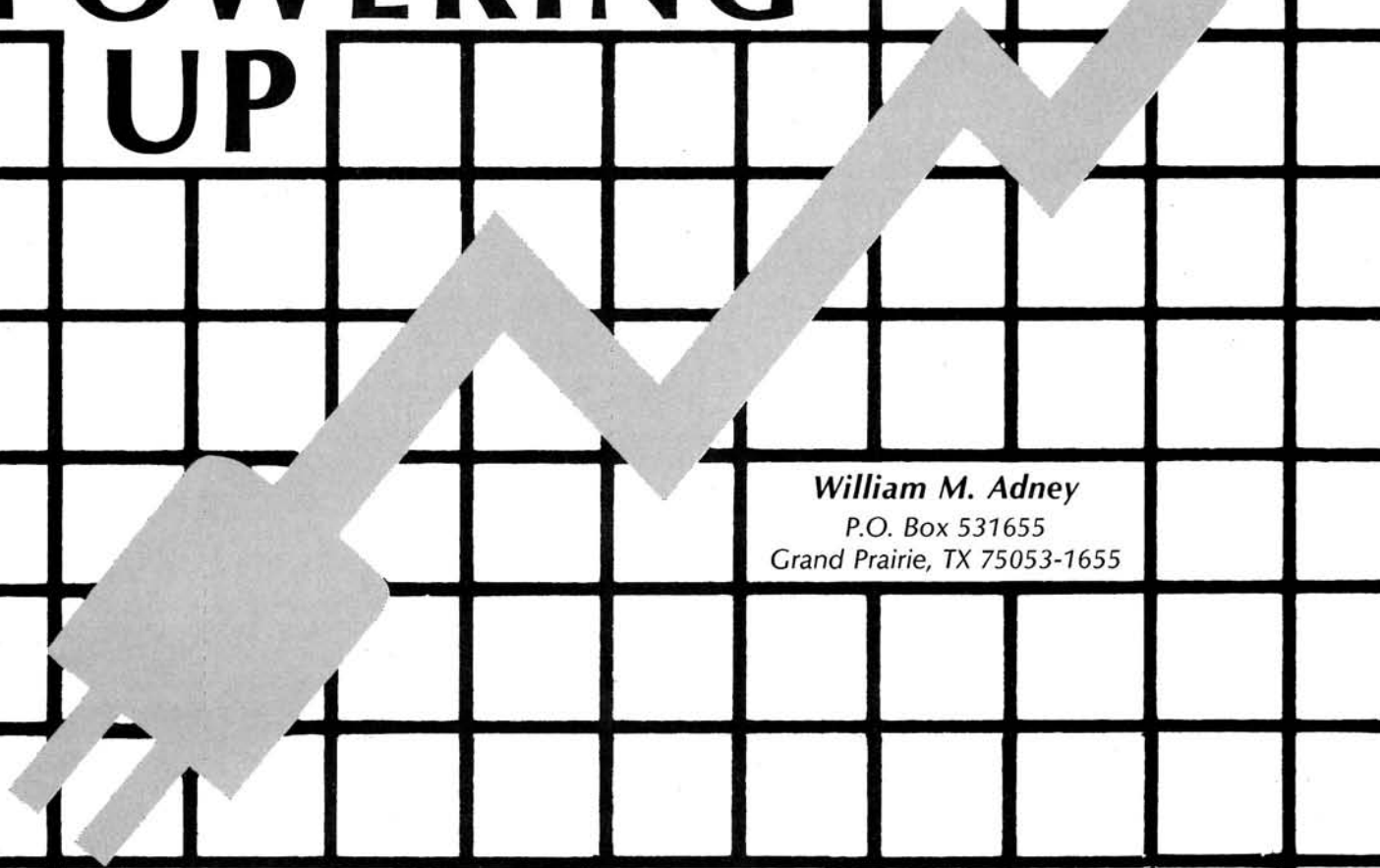
SPOOLDISK 89 and SLOT 4: Cards still available. Contact us for information.

Order by mail, phone or see a Heath/Zenith Dealer. UPS/APO/FPO shipping included. VISA or MC. WA residents add 8.1% tax. Hours: M-F 9-5 PST. We return all calls left on answering machine!

FBE

FBE Research Co., Inc.
P.O. Box 68234, Seattle, WA 98168
206-246-9815

POWERING UP



William M. Adney
P.O. Box 531655
Grand Prairie, TX 75053-1655

Using Batch Files And CONFIG.SYS

If you have ever spent some time looking at your system disk, you have probably noticed that there are two files that seem to be on every disk: AUTOEXEC.BAT and CONFIG.SYS. In this article, we will look at the general subject of batch files, including the AUTOEXEC.BAT file, and the configuration file, CONFIG.SYS.

Both the AUTOEXEC.BAT and CONFIG.SYS files were copied to your system disk when you installed MS-DOS, if you used the Zenith SETUP program that is supplied with all current MS-DOS versions. These files are important to the operation and functionality of your system, and you can use various parameters in those files to better control numerous functions.

Both of these special files also have one characteristic in common: they must be ASCII files. Remember that an ASCII

file must not have any special formatting characters in it, such as bolding, and you can create an ASCII file using the COPY command that was discussed last time. Or, you can use your word processor to edit and save the file in an "unformatted" or WordStar non-document mode.

For many user functions, the batch file capability represents one of the most important features of your computer system, so let's see what a batch file is and how it can help you.

What Is a Batch File?

As usual, we will begin with a definition of a batch file. A BATCH FILE is an ASCII file that contains various commands which are executed in the same order as they are listed in the file. Each line in the batch file contains a command that would normally be typed on the com-

mand line. For example, you could have the following commands in a batch file:

```
DATE  
TIME  
PATH A:\  
PROMPT $P $Q$Q$G
```

This particular batch file may be useful during system startup, and we will return to this example in a minute. It is also important to note that you can have a batch file that only contains a single line which can make it much easier to execute a long and complicated command, such as the BACKUP command.

```
A:\>==>BACKUP C:*.* A:/S/E:*.*.COM+*.EXE
```

The BACKUP command shown here can be used to backup all files on hard disk partition C to floppy drive A, include subdirectories in the backup (/S), and exclude COM and EXE files (/E). Entering a short command, such as BACKUP*, is

much easier than typing that entire command line.

Aside from the general requirement that batch files must be ASCII files, the file name must also be in the form of *filename*.BAT. The BAT file type or extension indicates a batch file, and the *filename* is what you actually type to execute the command.

Batch files are the last file type that DOS searches for in the standard order of internal, COM, EXE, and BAT files. Like all of the external commands, DOS will search the current directory on the current drive first, then other directories and/or drives will be searched if the PATH command has been entered. When you work with batch files, it is especially important to keep this search order in mind when you are naming a batch file.

To understand why this is critical, consider a batch file with the file name of DIR.BAT that contains a command like "DIR B:/P." Although that may appear to be a "good" file name, DOS will NEVER execute a batch file with that name. Why? Well, DIR is an internal command, and as such, it will be found first when DOS begins the command search for internal, COM, EXE, and BAT commands. And when DOS performs this command search, it stops "looking" at the very first match, in this case, an internal command. For similar reasons, you would probably not want to name a batch file as FORMAT.BAT because DOS would likely find FORMAT.COM first depending on your current drive and/or subdirectory and any path specifications entered with the PATH command.

There are occasions when you want to stop or terminate batch file execution, particularly when you are testing a new batch process. You can interrupt batch processing at any time by pressing CTRL-C or CTRL-BREAK which will display the following:

```
Terminate batch job (Y/N)?
```

If you type a Y, the job will be terminated, and you will be returned to the DOS command prompt. If you type an N, processing will continue with the NEXT command in the batch file.

One disadvantage of batch file processing is that it does not generally provide a way to enter a response to a program prompt — this must be done manually. For example, the FORMAT program contains a prompt like: "Format another disk (Y/N)?" You will have to enter responses to these prompts from the keyboard.

As you might expect, all batch file processing is performed by the Command Interpreter, COMMAND.COM. Each command is typed on a separate line in exactly the same way you would enter it at the command prompt. Each command is checked just as it would be on the command line. Batch file processing also in-

cludes some special commands that are helpful when "programming" a batch file, and we'll look at some of those in a minute.

The First Batch File

Batch files can have just about any valid file name with one exception — the AUTOEXEC.BAT file. The file name of AUTOEXEC is reserved for exclusive use by DOS because it searches for that file name every time the system is booted and AUTOMATICALLY EXECUTES the commands contained in that file. In fact, the Command Interpreter, COMMAND.COM, controls all command processing, in general, and batch processing, in particular, including execution of the AUTOEXEC.BAT file. You can always tell when DOS has been completely loaded in your system's memory when the system is booted because the Command Interpreter will then look for an AUTOEXEC.BAT file in the root directory on the bootable disk. If found, commands in the file are executed.

The AUTOEXEC.BAT file is quite useful for setting up your system, and you will probably want to include it on all of your system disks. For example, you will probably want to execute the PATH and the PROMPT commands every time you boot your system. You could have a batch file called START.BAT, but you could include those commands in an AUTOEXEC.BAT file and avoid typing the extra five keystrokes (i.e., START).

If you decide to use the AUTOEXEC.BAT feature, there is one interesting anomaly that you should know about. If you boot your system *without* an AUTOEXEC.BAT file, most DOS versions prompt you to set the current date and time. But, if you use an AUTOEXEC.BAT file, you will need to enter the DATE and TIME commands in it to prompt for those values unless you have a system that has a real-time hardware clock like the Z-200 or Z-386 computers. So, let's assume your computer does not have a real-time clock, and you would probably want an AUTOEXEC.BAT file that contains the following lines:

```
DATE
TIME
PATH A:\
PROMPT $P $Q$Q$Q
```

During the execution of this AUTOEXEC.BAT file, you will be prompted to enter a new system date, and a new system time. These prompts will appear exactly as they would if you had entered the DATE and TIME commands on the command line. The PATH command will set an expanded search path for COM, EXE and BAT files; and the PROMPT command will display a prompt that looks like:

```
A:\ ==>
```

Perhaps the best feature of batch file

processing is that it helps you reduce or eliminate repetitive and complicated commands to a single batch file command. In the special case of the AUTOEXEC.BAT file, it is executed every time your system is booted, but it MUST ALWAYS be in the ROOT DIRECTORY of the bootable disk so that DOS can find it.

One other special note about the AUTOEXEC batch file: many people have found, usually by accident, that you can "restart" the AUTOEXEC file by just entering the file name, like any other batch file. That is occasionally useful, but it is bad practice since many people include memory resident commands, like Borland's SideKick, in the AUTOEXEC file. If you attempt to restart this file, you will actually load a second copy (or third or fourth, etc.), and that takes up an unnecessary amount of memory. It is much better simply to reboot the computer to implement a new AUTOEXEC batch file.

Naming and Using Batch Files

Developing a name for batch files is easy, and perhaps the most important thing is to identify it with a name that is meaningful to YOU. A previous example suggested that BACKUPC.BAT would be an appropriate name for the batch file that contained the following line:

```
BACKUP C:*. * A:/S/E:*.COM*. EXE
```

The exact file name for the batch file doesn't really matter too much so long as it is not AUTOEXEC and does not conflict with the file name for an existing command. Instead of choosing BACKUPC for the file name, you could also use BACKC, BC or just B. When choosing a file name for a batch file, it is generally better to pick a descriptive name that is not likely to conflict with any file name for an existing COM or EXE file. And although you can include any allowable characters in the file name, such as a hyphen (-), it is generally easier to limit these file names to alphabetic characters for ease of typing the command. If you run into a situation where you want to have two similar types of batch files for some reason, you can always add a number to differentiate it. For example, you could have file names like FORMAT3.BAT and FORMAT5.BAT that contain the appropriate FORMAT switches for formatting a 3.5" and a 5.25" disk, respectively. Now let's take a look at some useful batch files and commands.

After the system is booted, there are several kinds of applications that you may want to use — perhaps word processing or running a spreadsheet. Word processing is the most common use of a personal computer, so let's take a look at some examples for starting a word processor. Let's start with a simple example using WordStar as a word processor with a file name of STARTWS.BAT as shown in Figure 1.

```

(1) REM          Start WordStar 4 on current drive/directory
(2) ECHO Command syntax is:  STARTWS filename
(3) ECHO Hit CTRL-C to abort or...
(4) PAUSE
(5) REM          Execute WordStar from subdirectory for filename
(6) C:\WS4\WS %1
(7) REM          Set up backup for drive A and copy file to it
(8) ECHO Insert appropriate backup disk in drive A and...
(9) PAUSE
(10) COPY %1 A:/V

```

Figure 1
STARTWS.BAT

This batch file illustrates the use of the three most important batch file commands: REM, ECHO, and PAUSE. It also demonstrates how a batch file can be used to make an automatic backup by using the COPY command to copy the edited file from the hard disk to a floppy disk in drive A. The numbers in parentheses at the beginning of each line shown the line number that will be used for explanation purposes, and they are NOT included a batch file.

All batch file processing commands, such as REM, ECHO, and PAUSE shown in Figure 1, are internal DOS commands. You can run a batch file on any drive in any subdirectory without having to worry about whether DOS can find them or not. From a previous article, you will remember that all internal commands are not preceded by a drive letter or a subdirectory name. That makes it much easier to use these commands in any batch file, including the AUTOEXEC.BAT file. These three commands form the basis for nearly all kinds of batch file processing, and we will examine each in turn.

The REM Command

The REM command is used to include a "comment" or reminder (and is short for REMark — an interesting name) of some kind in a batch file. Its syntax is shown in Figure 2.

```
REM [comment]
```

Figure 2
REM Command Syntax

The REM command is useful for including all kinds of comments in a batch file. For example, line (1) in Figure 1 describes the purpose of this batch file, and it is good practice to include a description of each as one of the first lines in the file. Although you may know what a specific batch file is for today, you may not remember three months from now. Good programmers add comments to their programs, and you are programming a batch file, so it is wise to add whatever comments are necessary for a clear description of what is going on. You can also add multiple lines of comments by simply preceding each line with the REM com-

mand if you need more than one line of comments.

REM — like all batch processing commands or any command for that matter — must be the FIRST command on each line. If it isn't, you will see the infamous "Invalid command or file name" error message because DOS will not "find" it.

You can also use the REM command by itself without the optional comment to "separate" various lines in a batch file. In the context of this example, ALL lines preceded by the REM command will display exactly as entered in the batch file in Figure 1. The ECHO command performs a similar function.

The ECHO Command

The ECHO command is quite similar to the REM command, and its syntax is shown in Figure 3.

```
ECHO [comment]
ECHO/OFF
ECHO/ON
```

Figure 3
ECHO Command Syntax

Like the REM command, ECHO also allows you to enter an optional comment as shown in line (2) of Figure 1. But ECHO really has two functions, and that is only the first.

The second function is to turn off the CRT display (i.e., "echo") of batch file commands as they are executed by using the "ECHO OFF" command. The normal DOS default (i.e., ECHO ON) is to display each command in the batch file as it is executed so you can see what the current command is. Regardless of the echo setting, however, all prompts and other information normally displayed by a command (e.g., "Format another disk (Y/N)?") will still be displayed — only the command line in the batch file is NOT displayed when the "ECHO OFF" command is entered.

Although the "ECHO OFF/ON" command is typically used in a batch file, it is one of the few batch processing commands that can reasonably be used at the DOS command prompt. You could, for example, include the "ECHO OFF" command in the AUTOEXEC.BAT file, and all

batch files from then on would NOT display the command lines. I don't recommend that, but it is an alternative. We will take a more detailed look at what really happens when you use the ECHO command in a minute, but now it's time to look at the PAUSE command.

The PAUSE Command

The PAUSE command also has a syntax similar to REM and ECHO, as shown in Figure 4.

```
PAUSE [comment]
```

Figure 4
PAUSE Command Syntax

The significant difference in the PAUSE command is that it also displays an optional comment followed by a message like "Strike a key when ready...". Depending on what you are doing, the PAUSE command is normally used to prompt you to take some specific action before proceeding with the batch file processing. In Figure 1, notice that PAUSE has been used by itself without the optional comment.

Another nice feature in the PAUSE command is that it can be used to provide a convenient way to completely stop batch file processing. When batch file processing has "paused", you can press a CTRL-C or CTRL-BREAK to stop the batch file at that point, or you can "Strike any key to continue". When you are programming a complicated batch file, it can be helpful to include a PAUSE command every other line or so to help you troubleshoot any problems. You can also use the PAUSE command to stop processing because you need to change a disk in a drive or for any other reason. Let's return to the discussion of Figure 1 to see what happens when this batch file is executed.

Using a Batch File

Now that you know about the three common batch file commands, it is important to see how this file could be used. If I wanted to use this batch file to start WordStar to edit this Powering-Up article, which has a file name of POW88-12, I would enter the command line as follows:

```
D:\ \ ==>STARTWS POW88-12
```

And I would actually see the results displayed on the CRT, as shown in Figure 5.

The basic command "STARTWS" followed by the file name is shown in bold as you would actually type it on the command line. Some comments about the processing are shown in parentheses. The only real problem here is that the display looks messy because there are a number of lines presented that we really don't need. The REM lines, for example, were intended to be information about the file and really do not need to be displayed

```

D:\REMARK ==>STARTWS POW88-12      (Command to start processing)
D:\REMARK ==>REM                    Start WordStar 4 on current drive/directory
D:\REMARK ==>ECHO Command syntax is: STARTWS filename
Command syntax is:  STARTWS filename
D:\REMARK ==>ECHO Hit CTRL-C to abort or...
Hit CTRL-C to abort or...
D:\REMARK ==>PAUSE
Strike a key when ready . . .
D:\REMARK ==>REM Execute WordStar from subdirectory for filename
D:\REMARK ==>C:\WS4\WS POW88-12 (Note file name substitution)
D:\REMARK ==>REM Set up backup for drive A and copy file to it
D:\REMARK ==>ECHO Insert appropriate backup disk in drive A and...
Insert appropriate backup disk in drive A and...
D:\REMARK ==>PAUSE
Strike a key when ready . . .
D:\REMARK ==>COPY POW88-12 A:/V
1 File(s) copied
D:\REMARK ==> (Return to command prompt)

```

Figure 5
STARTWS Batch File Command Display

when the file is executed. The easy way to clean up this display is to use the ECHO command at the beginning and end of the file, as shown in Figure 6.

As before, you type the command "STARTWS POW88-12", and parentheses are used for comments about the processing. Notice that the addition of the

```

ECHO OFF
(1) REM          Start WordStar 4 on current drive/directory
(2) ECHO Command syntax is:          STARTWS filename
(3) ECHO Hit CTRL-C to abort or...
(4) PAUSE
(5) REM          Execute WordStar from subdirectory for filename
(6) C:\WS4\WS %1
(7) REM          Set up backup for drive A and copy file to it
(8) ECHO Insert appropriate backup disk in drive A and...
(9) PAUSE
(10) COPY %1 A:/V
ECHO ON

```

Figure 6
STARTWS.BAT with ECHO OFF

There are a couple of changes to note here. The "ECHO OFF" command was added as the first line so that all of the command lines would NOT be displayed when the batch file was run. Then, the "ECHO ON" command was added as the last line in the file to return to the normal DOS default that displays all command lines in a batch file. This is good practice since it allows you to suppress the display of the command lines after you have thoroughly tested and debugged the batch file. Now let's see what that does to the displayed results, as shown in Figure 7.

"ECHO OFF" command suppresses all of the command lines, including every line that is not preceded by the ECHO command and a comment. Also notice that the PAUSE command's "Strike a key when ready" prompt is displayed, as is the information message from the COPY command showing "1 File(s) copied".

Batch files are incredibly useful for repetitive tasks, and every once in a while, you will find that you want to make a TEMPORARY change to an existing batch file. Perhaps you want to create the file in its entirety, as shown in Figure 6 with the

```

D:\REMARK ==>STARTWS POW88-12      (Command to start processing)
D:\REMARK ==>ECHO OFF
Command syntax is:  STARTWS filename
Hit CTRL-C to abort or...
Strike a key when ready . . .

                (WordStar is started so you can edit the file)

Insert appropriate backup disk in drive A and...
Strike a key when ready . . .
                1 File(s) copied
ECHO ON
D:\REMARK ==>                        (Return to command prompt)

```

Figure 7
STARTWS Batch File with ECHO OFF

ECHO commands. But there is a problem because the batch file is not working as it should, so you want to actually see all of the command lines as they are processed. One solution is to simply delete the line that contains the "ECHO OFF" command, but there is an easier way.

This technique is usually called "commenting out" a line by programmers who can use a similar idea when writing a program. The idea is to preserve an existing line of "code" or command line in a batch file without actually deleting it from the file. Although this may sound like some kind of magic, all you have to do is precede any line by the REM command in a batch file, and batch file processing simply treats the rest of that line as a comment since REM is the first command on the line. To "comment out" the ECHO OFF command, all you need to do is begin the line with the REM command so that the line now contains:

```
REM ECHO OFF
```

To return the file to its "normal" state, just delete the REM command from the beginning of the line. While this trick is not particularly necessary in this file, it does make it obvious what you have done to change the file. Then you don't have to remember what command (or where it was) you deleted for your "temporary" file — everything is still in the file. At this point, you know most of the batch file fundamentals, except for one thing that has been ignored in the example until now. The general subject is called Replaceable Parameters.

Replaceable Parameters in a Batch File

If you carefully examine the batch STARTWS file in Figure 6, you will find that line (6) contains the following:

```
C:\WS4\WS %1
```

The beginning part of that command says that we want DOS to look on drive C in the \WS4 subdirectory to execute the WS command. No problem with that because you have seen that syntax before. But what is that "%1" parameter on the command line? The "%1" is called a Replaceable Parameter. REPLACEABLE PARAMETERS are replaced by the corresponding value that is actually typed on the command line. In this example, the batch file was started by the command line:

```
D:\REMARK ==>STARTWS'POW88-12
```

Therefore, the first replaceable parameter on the command line was the filename of "POW88-12", and the first replaceable parameter of %1 was replaced with that filename so that batch file processing actually executed the command line:

```
C:\WS4\WS POW88-12
```

In programming discussions, a replaceable parameter is usually called a VARIABLE because a symbol, like %1, represents a variable value (like the filename

in this example) that is "stored" under that name. For batch processing, we use the percent sign symbol to identify a replaceable parameter, and these parameters are numbered from 1 to 9; that is, you can have up to nine replaceable parameters on a command line ranging from %1 to %9 so that a command line could look like:

```
C:\ ==>filename x1 x2 x3 x4 x5
                                x6 x7 x8 x9
```

The "x" values represent the various locations of the replaceable parameters %1 to %9. Although a space is used to separate each parameter in this example for clarity, you will find that any valid DOS command line delimiter — such as a colon (:) — may also be used to separate such parameters as a drive letter and a path or file name. To illustrate the use of multiple replaceable parameters in a batch file, consider the simple LOGON.BAT file shown in Figure 8.

```
ECHO OFF
REM      Logon onto new drive and/or directory
REM      Note that root directory is default if no path name
ECHO Command syntax is: LOGON d directory (no colon after d)
CD %1:\%2
%1:
ECHO ON
```

Figure 8
LOGON.BAT

Assuming that you have a search path set with the PATH command so that DOS can always find the LOGON.BAT file, you could use this batch file to change your current drive and/or subdirectory. For example, you could enter the command line:

```
C:\ ==>LOGON'D'REMARK
```

In this case, the %1 parameter is replaced by a D, and the %2 parameter is replaced by REMARK so that the two effective command lines become:

```
CD D:\REMARK
D:
```

Notice that in both cases where %1 appears in the batch file, it is replaced by the D because it is the first parameter on the original command line. And of course, the %2 parameter is replaced by the REMARK (second) value which is a subdirectory name in this example. So far, so good, but what happens if you want to (or forget to) enter a directory name? In this example, the %2 is effectively a "blank" value so you can enter the command:

```
C:\ ==>LOGON D
```

When batch processing replaces the parameters, the two command lines become:

```
CD D:\
D:
```

Since there was no directory name entered on the command line for the batch file, the %2 becomes a blank, and the default is to logon to the root directory on the specified drive — D in this example. But there is one additional point

that needs to be made here.

Suppose you had forgotten to include the backslash (\) in the batch file line containing the "CD %1:\%2". What would have happened? In this example, the command would have simply been "CD D:REMARK", and you would have probably seen some kind of error message from this Change Directory command syntax.

This particular example illustrates the importance of being quite precise when you are developing a batch file. Remember, you are programming in a very real sense, and everything must be EXACTLY correct for each and every command line in your batch file. The simple omission of the backslash in this example results in an error message. When the backslash is included as shown, the worst that can happen is that you will be in the root directory of the specified drive — assuming that

you have typed a valid drive letter, of course.

Once you have spent several hours trying to troubleshoot this kind of problem, it is much easier to appreciate the difficulties involved in programming powerful and complex application software, such as a word processor or spreadsheet. Most batch file (and application program) bugs are usually the result of a failure to consider ALL potential kinds of problems that "might" occur. But if you include your intended command line syntax as part of the batch file, and consider ALL possibilities (e.g., what happens if a directory name is omitted from the command line?), you will be able to write effective and useful batch files.

Batch File Processing

Batch files offer an easy way to automate repetitive tasks and can save you a lot of time and keystrokes. It is usually much easier to enter a series of commands on your system, write down what you entered as you do it, and then create a batch file from your written list. Be sure to include lots of REM statements with comments about what is going on so you can quickly review it in the future. And you can always use the REM command to comment out various command lines in a batch file to create a temporary version.

Since this has been an introduction to batch file processing, we have not looked at the more complicated commands: FOR, GOTO (and labels), IF, and

SHIFT. You can use these commands to create menu-type systems among other things, but there is an easier way because there are a number of software packages available, such as DSMANAGER, that are much better in that respect. Still, the use of batch files to perform your daily computer tasks will help reduce or eliminate many keystrokes and mistakes.

One other note about using batch processing — If you have a hard disk, you may find it useful to create a \BATCH subdirectory that contains all of your batch files. And if you also include that subdirectory in the PATH command in the AUTOEXEC.BAT file, you can execute any batch file regardless of your currently logged drive and/or subdirectory. Now let's take a look at the CONFIG.SYS file.

The CONFIG.SYS File

The Configuration File — CONFIG.SYS — is another special file name that DOS searches the ROOT DIRECTORY for each time the system is booted. If you also have an AUTOEXEC.BAT file in the root directory, you will notice that commands contained in the CONFIG.SYS file are performed before execution of the AUTOEXEC file begins. The most common commands shown in a CONFIG.SYS file are shown in Figure 9.

The CONFIG.SYS file shown in Figure 10 is an example of one I use on my Z-248 system using Zenith MS-DOS version 3.21. In order to avoid burdensome technical detail, suffice it to say that the BUFFERS= command is simply a way to reserve some memory for faster system performance for disk reads and writes. This reserved memory helps speed up disk reads and writes. For most systems, a value between 25 and 40 seems to work the best, but you may have to experiment to get the best performance for a specific software package.

The FILES= command is a way to tell DOS how many files may be used at any one time. Here again, this command reserves some memory for use by DOS, and the amount of memory used depends on the number of files entered in the command line. For most purposes, 25 seems to be a good choice, but you may have to increase that if you use a database system like dBASE.

Two important points about the BUFFERS= and FILES= commands. First, increasing the number for each may or may not improve the performance of your system, and you will need to experiment to determine the best values for your particular system. In some cases, increasing these numbers beyond the recommended values may actually slow down your system because these commands use system memory (i.e., RAM), and your particular programs may run better with more available RAM.

The second important point is that

```

BUFFERS=30
FILES=25
DEVICE=C:\DRIVERS\VDISK.SYS SIZE=150 SECTSIZE=512 RDIRS=64 /E

```

Figure 9
Common CONFIG.SYS File Commands

most of today's programs tell you the values required for these commands, so you probably won't have to worry much about it. The values suggested here will work with nearly all common application programs, but you should check your documentation to be sure. If these CONFIG.SYS commands are not mentioned, then it is generally safe to assume the values suggested here will work fine. But regardless of what the program documentation says, if your actual values for the FILES= and BUFFERS= commands in your file are larger than recommended for a specific program, do not change them.

If you find a situation where the program documentation recommends larger values than suggested here, always use the larger values because they are probably required by the program for some reason. In short, always use the largest value that is at least equal to or greater than the recommendations in your program documentation. The only real penalty for using a larger value is that these commands will require a modest amount of extra system memory. And if you want to experiment to see how much memory is actually required, you can always change the value, reboot the system, and run the CHKDSK command to see how much memory is being used. Now let's take a look at the DEVICE= command.

The DEVICE= Command

The DEVICE= command in Figure 10 is shown with the example of the virtual disk (sometimes called a memory disk) device driver that has a file name of VDISK.SYS and is furnished with nearly all current DOS versions. A virtual or memory disk is useful because it is much faster than mechanical disks, either floppy or hard drives, because there is no mechanical movement required. In other words, the VDISK device driver actually "creates" something that looks like a standard disk drive as far as DOS is concerned.

The general command form used to implement a device driver is the standard one:

```

DEVICE=[d:][\path]driver-name drive
r-parameters

```

One of the nice features of the DEVICE= is that you can specify the location of the device driver file name by using an optional drive and/or path specification. This means that, even though the CONFIG.SYS file MUST be in the root directory, the device driver may be in any subdirectory provided that its location is properly identified with the driver letter and path. In this example, I have shown how to implement the VDISK device driver which is

contained on my drive C in the XDRIVERS subdirectory. One of the more interesting omissions from most of the current DOS documentation is that you can specify a "comment" as part of the VDISK driver parameters. This is an unpublished feature that works just fine in Zenith MS-DOS versions 3.2 and later, as far as I can tell. If you follow the manual explicitly, you will find an example similar to:

```

DEVICE=C:\DRIVERS\VDISK.SYS 150 512 64
\E

```

I think it is much more meaningful to include the comments when possible. Then you can look at a command line like:

```

DEVICE=C:\DRIVERS\VDISK.SYS SIZE=1536 S
ECTSIZE=512 RDIRS=64 /E

```

If you know a little bit about the VDISK device driver, it is easy to see that you are reserving 1536 kilobytes of memory with a sector size of 512 bytes and 64 root directory entries. Since this example was taken directly from one of my own files, it also includes the /E switch which creates the VDISK in Extended Memory

DOS version that contains a disk caching device driver called ZCACHE.SYS that will also help improve the performance of your system by using some system memory.

There are a lot of commands that can be entered in a CONFIG.SYS file, and they are not often required or used. If they are needed, you should find that information in the specific program documentation, and their addition to the CONFIG.SYS file will generally not cause any problems. For the most part, all of these commands can be entered in just about any order you want, but there is one major exception to that.

Let's say you have a Z-150 series computer (the exact model doesn't really matter) and you have added an Expanded Memory board (sometimes called an EMS board) to it. Virtually all EMS boards are supplied with a special device driver that is REQUIRED to be in the CONFIG.SYS file. To be consistent with the previous DEVICE= command example, we will assume that the EMS driver has a file name of EMS.SYS and is located on drive C in the \DRIVERS subdirectory as before.

One might assume that it is a simple matter to insert that command as the last

```

BUFFERS=30
FILES=25
DEVICE=C:\DRIVERS\VDISK.SYS SIZE=150 SECTSIZE=512 RDIRS=64 /A
DEVICE=C:\DRIVERS\EMS.SYS (Will not work)

```

Figure 10
Incorrect CONFIG.SYS with EMS Driver

— a special add-on card that I have in my '248 system.

When you create a VDISK, you treat it exactly like any other disk drive in your system because it is assigned the next available drive letter. For example, if your system has one or two floppy drives, the VDISK will normally be accessed as drive C, since A and B are normally reserved for floppies. If your system has a hard disk with four partitions like mine (C, D, E, and F), the VDISK is accessed as drive G. I can copy files to and from drive G, and I can change the current drive to G by simply entering "G:" as usual.

Other device drivers are furnished with current MS-DOS versions, and they are implemented in a similar way. You may also find that you have a Zenith MS-

line in the previous example so that you would have the CONFIG.SYS file shown in Figure 10.

The only other change from the previous example in Figure 9 is that the /A switch is now used to tell the VDISK driver to use Expanded (not Extended) memory. This CONFIG.SYS file will simply not work, and in some DOS versions, it will actually cause a system freeze. What's wrong? The problem is that you have attempted to use the Expanded Memory with VDISK before you loaded the EMS device driver. Although the logic of this requirement seems to be obvious once it is explained, some people have reported strange problems which turned out to be nothing more than the simple mistake of trying to use Expanded Memory before the appropriate driver was loaded. As-

```

BUFFERS=30
FILES=25
DEVICE=C:\DRIVERS\EMS.SYS (DOES work)
DEVICE=C:\DRIVERS\VDISK.SYS SIZE=150 SECTSIZE=512 RDIRS=64 /A

```

Figure 11
Correct CONFIG.SYS with EMS Driver

suming everything else is correct, the proper order for these commands is shown in Figure 11.

Although I have seen some reports to the contrary, this is the only situation I have found where the order of the commands is important. And it really does not matter which line you choose for the EMS driver, so long as it is before any other command requiring Expanded Memory, such as VDISK.

Next Time

The next article will talk about how to use Input/Output (I/O) redirection in

conjunction with the DOS filter commands: FIND, MORE, and SORT. We'll take a look at the difference between Command Piping and the other general forms of I/O redirection.

If you have any questions about anything in this column, be sure to include a self-addressed, stamped envelope (business size preferred) if you would like a personal reply to your question, suggestion or comment.

Products Discussed

MS-DOS Version 3
PC only (OS-63-31)

\$150.00

Heath/Zenith Computer Centers
Heath Company
Hilltop Road
St. Joseph, MI 49085

DS Manager
Design Software
1275 W. Roosevelt Road
Chicago, IL 60185
(800) 231-3088

\$49.95

(Orders only — except Illinois)
(312) 231-2225 (Illinois only)

*



IBM XT in an H/Z-100

Scottie Board W/ZPC \$149.

OPTIONS AVAILABLE

- PC Compatible Serial Port W/Cable \$50.
- 2nd Port W/Cable \$45. • Clock/Calendar \$45.
- No Solder H/Z-100 MOD Kit \$5.

Requires H/Z-100 MS-DOS 768 of RAM, H/Z-100 Modification.

FACTORY NEW..THE REAL THING

Z-217-1 W/DOCS, HDWE, CABLES, INSTRS

- With 20 Meg ST-225 Drive \$649.
- With 40 Meg MS3650 Drive \$789.

SCSI

CDR Host Adapter & OMTI 3100 SCSI Controller

- With 20 Meg ST-225 Drive \$749.
- With 40 Meg MS3650 Drive \$889.

MORE FOR THE H/Z-100

Floppy Drives: Fujitsu\$79.
Teac-5595.

Printers: Citizen 180-D195.
Citizen 120-D179.

MEMORY & SPEED UPGRADES, SERVICE & SUPPORT.

FOR PC's & COMPATIBLES, HARD DISKS

NEW imager 110 Mb VCR Tape Backup\$210.

Monitors: NANA0 14" 8060S Flexscan585.
(H-15.75Khz-35Khz, V-50-80Khz Automatic 0.28MM Dot Pitch)

Adapter Card for above monitor (800x600)179.
High Res Monochrome-Casper & Samsung89.

Modems: 1200 Baud Internal79.
1200 Baud External99.

**MANY MORE ITEMS, BRANDS & MODELS
AVAILABLE AT GREAT PRICES**

Please Call, Write, or Circle For Free Catalog

Scottie Systems, Inc.
2667 Cropley Ave. #123
San Jose, CA 95132
(408) 259-6226

IBM XT is a registered trademark of IBM Corp. • ZPC is a product of HEATH USER GROUP. VISA & MC ACCEPTED, ALL PAYMENTS SUBJECT TO APPROVAL

Circle #121 on Reader Service Card

Speed Mods \$34.95

- Easily installed. No Trace Cuts or Permanent Mods
- H/Z 150/160 Software Select 6.87 or 4.77 MHz. Free Hardware Reset, 8 MHz V20 \$12, 8 MHz DMA 8237 and Bus Controller 8288 \$39, High Speed ALS Chips \$8
- H-89 Software select 2 or 4 MHz, CP/M and HDOS

H-89 20 Meg Hard Disk, only \$475

- Boots from hard disk, 3 Logical Drives
- Supports Heath, Magnolia, and CDR
- Hard Disk can be used in H/Z 150/160
- All Software and Interface Card \$175
- H-37 Card, cable, MTR-90 EPROM \$150

100/148/150/248 Hardware

- SmartWatch (No extra Z-100 charge) \$34
- Hard Disks 20M \$279, 30M \$309, 40M \$449
- H150: MTPAL 704k RAM \$19.95, Super PAL 1.2 Meg RAM \$45, 2 Meg EMS RAM \$99
- H150: 80286 - 8 MHz \$299! 12 MHz \$319!
- H148: Expansion bus \$69 (with clock \$99), MT148 704k RAM \$19.95
- H248: 3 Meg EMS and backfill RAM \$109
- EGA+ \$145, Serial/Parallel/Game Card \$91
- Everex 2400 baud MODEMS, \$169 internal, \$199 external, Add \$10 for level 3 MNP

MS-DOS and CP/M Software

- MT Accountant \$19 Perfect Money \$29
- Perfect Printer \$19 Paycheck \$39

Micronics Technology

(205) 244-1597 BBS: 244-0192
Suite 159, 54 Dalraida Rd.
Montgomery, AL 36109

Checks, Cash, VISA, or MC accepted
Shipping \$7 hard disks, \$4 cards, \$2 other
Call or write for our FREE complete Catalog!

Circle #114 on Reader Service Card

Most home finance programs have serious limitations.

Like the dodo bird, they're slow, awkward and not very smart. You'll waste eons of time, only to realize they're dead ends.

HFS-III, the Home Finance System, is fast, loaded with features and intuitively smart.

- Easily manage up to 100 checking, saving, CD, IRA and other accounts.
- Print checks on any form.
- Organize financial summaries and activity reports based on 100 expense and 15 deposit codes you define.
- Flag tax-deductible or medical expenses.

In short, control your finances to save money, save time and simplify taxes.

Find out why so many leading computer authorities and users are calling HFS-III the "superior species." See it today at participating Heath/Zenith Computers and Electronics Centers, or call toll free for more info.

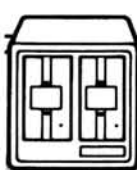
Only \$99

Requires: DOS 2.0 or higher, IBM* PC/ XT/AT or compatible or Zenith* Z100 computer; 2 disk drives or hard disk, 256K RAM.



Jay Gold Software, Inc.
P.O. Box 2024
Des Moines, IA 50310
(800) 541-0173

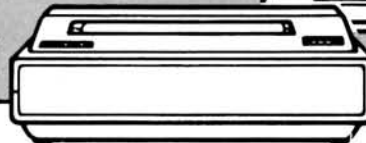
Circle #137 on Reader Service Card



Related Products



Jim Buszkiewicz
HUG Managing Editor



NOTE: The following information was gathered from vendors' material. The products have not been tested nor are they endorsed by HUG. We are not responsible for errors in descriptions or prices.

Sign Language Courseware now available. Data Assist of Columbia, Ohio announces the availability of its Sign Language Quiz (SLW) courseware package. SLQ is a program for the IBM PC that helps teach American Sign Language by using the computer's graphics capabilities to draw life-size pictures of hand gestures. The student then selects the correct interpretation of the picture from among several choices by entering a letter or number into the computer.

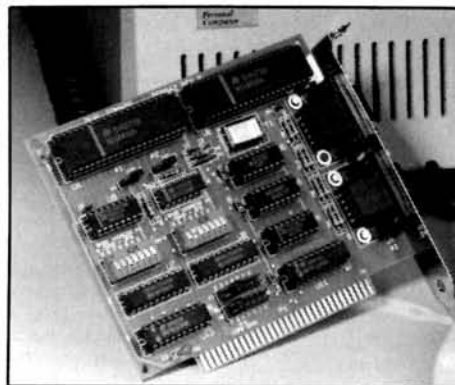
"We are very proud to offer Sign Language Quiz as our first courseware product," says Carol Sattler, Data Assist's Marketing Director. "SLQ will help many people learn sign language. The computer makes it fun for students, and that makes it exciting for teachers. We have seen students as young as five years old learning the gestures with SLQ."

SLQ was developed using PC-QUIZZER, the popular computer-based training package also available from Data Assist. The written lessons may be used for drill and practice or as flashcards or a computer-based quiz. A copy of the PC-QUIZZER run-time-system is included with the SLQ package.

SLQ lessons are presented to the student using text and graphics windows. Questions may be either ordered sequentially or randomly generated by the computer. Missed questions may be flagged for later retry or may be repeated immediately until the correct answer is given. A VOTRAX or ECHO speech synthesizer may also be used with SLQ.

Registered owners of PC-QUIZZER may modify the SLQ lessons for their special needs. Additional text, help screens, color graphics, and speech and music can be added. Organizations with a PC-QUIZZER site license are free to copy and distribute SLQ for educational purposes.

SLQ is available on 5.25-inch or 3.5-inch disks for any IBM PC or compatible with a CGA, EGA, MCGA, or VGA graphics adapter. Customers should specify on their order which disk type and graphics adapter they are using. SLQ courseware is priced at \$35.00 per copy and is not copy-protected.



MetraBytes' DUAL-422 is a two channel RS-422 interface board designed to provide high speed communications capability for IBM PC/XT/AT and compatible computers. The DUAL-422 allows data transfers at speeds of up to 57.6 Kilobaud over distances as great as 4000 feet. Baud rate is selectable over a wide range of values between 120 baud and 57.6 Kbaud. The 9600 baud limitation of most DOS and BASIC communication routines can be easily overcome through simple BASIC programming.

The DUAL-422 consists of two independent RS-422A communication ports, each with its own base address and interrupt selection controls. Either channel can be set at COM1; or COM2; or any other desired base address/interrupt level combination.

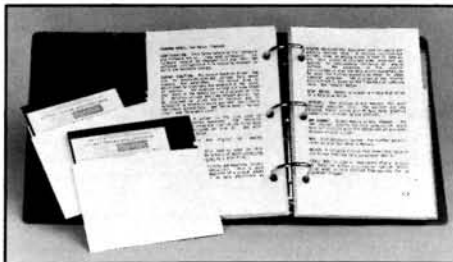
The design of the DUAL-422 is based around the INS 16450 UART. This gives the board compatibility with both the INTEL 8250 UART and IBM asynchronous card. Communication parameters such as 5, 6, 7, or 8 data bits and even, odd or no parity check modes are software selectable by the user. Additionally, Request to Send (RTS) and Clear to Send (CTS) data signals are provided.

The DUAL-422 plugs directly into any half- or full-size IBM PC/XT/AT expansion board slot. Connection to external serial busses are made through two 9-pin "D" connectors. Its combination of low cost, high data transfer rates and flexible design make the DUAL-422 the ideal choice for applications requiring interface of the microcomputer to printers, plotters, modems, networks and instrumentation. The DUAL-422 also provides an easy method of communicating with signal conditioning and control systems, such as Metra-

Bytes' M1000 series signal conditioning modules or MetraBus industrial control and monitoring system.

For more information, contact Metra-Byte Corp., 440 Myles Standish Blvd., Taunton, MA 02780. (508) 880-3000.

Dual-422	
Two Port RS-422 Interface Bd	\$260
SMC-9	
RS-422 9-Pin "D" Connector	12
C422-10	
RS-422 Cable	25



MetraBytes' SNAP-CALC is a general purpose software package that allows the user to analyze data as it is being acquired. Snap-Calc works in conjunction with Snapshot Storage Scope software (also available from MetraByte) to provide the capability of analyzing acquired test data and displaying both the acquired data and the calculated results.

Data may be processed during acquisition (concurrent processing) or after it is acquired and stored in a file (post processing). Concurrent processing provides immediate results, while post processing is used to analyze previously collected data. Concurrent and post processing may be used in combination or separately. Post processing can be performed without operator intervention to allow the automatic analysis of multiple files.

Snap-Calc requires no previous programming experience. Menus make setting up the required analysis easy. Mathematical operations available included calculation of arithmetic, trigonometric, calculus, logic, correlation, and statistical functions. Equations are defined in a user-friendly algebraic manner. Single keystroke commands are used to define many operators. For example, c = cosine and d = differentiation. The list of operators is displayed on the screen while equations are being defined. Frequently used functions, equations or subroutines

may be saved, recalled, and displayed as macros.

Snap-Calc analyzes both single data points and waveforms, enabling a large amount of data to be processed very quickly. Up to 8 waveforms can be plotted for any combination of inputs and outputs. Up to 32,000 points may be analyzed at one time.

For more information, contact Metra-Byte Corp., 440 Myles Standish Blvd., Taunton, MA 02780. (617) 880-3000.

Snap-Calc	\$495.00
w/Snapshot Storage Scope	895.00

Both Snap-Calc and Snapshot Storage Scope are compatible with MetraByte DAS-16 and DAS-8 data acquisition boards.

Superset Improvements

The "Superset" combines the features of the standard H19, HUG/Watzman, Super-19, and Ultra-ROM all into one. It features a vastly increased character set, improved graphics, faster operation, rock-stable "flicker-free" display, and much more.

The Superset is a new set of ROMs and RAMs for your terminal logic board, expanding its memory by 8:1. (Over 64K on the terminal!) No changes are made to your CPU board, and the Superset supports all ESC sequences of the standard H19 and HUG/Watzman ROMs so no software changes are necessary. The new parts are significantly faster, and reduce power supply loading for cooler operation. New features (*) include:

- 4 text fonts: standard H19, VT-100, T-PROM, and Greek. Optional Superfont adds 4 more; Z29, IBM PC, superscripts, subscripts, and *double-wide* characters.
- 4 character sets: H19, Z-100, DEC special graphics, G-PROM, and math. *Optional Superfont adds IBM PC, fineline, and all-point addressable 100x160 graphics.*
- *Windows! With Superclock, save/restore/swap any rectangular window on the screen with a 2nd page of off-screen memory. Allows animation, pop-up menus, etc.*
- On-screen menus for instant help and setup.
- True reverse screen operation (black characters on white screen).
- New attributes; 8th bit selects reverse video, blink, or 2nd character set.
- 4-mode function keys (normal/shift/control/ctl-shift).
- Transmit speed limiting for improved screen dumps.
- Improved interlace mode with reduced flicker.
- Superfast operation (38400 baud without handshaking).
- On-screen clock/calendar; *battery

backed-up with optional SUPER-CLOCK*.

- *Save/restore/swap terminal modes commands.*
- *"Soft" ESC z resets to last saved mode, not DIP switch settings.*
- *Local edit mode.*
- Native and transparent modes for special applications.

The Superset lets the TLB run at 38400 baud, and Heath CP/M supports this speed after boot-up. But the Heath monitor ROMs (MTR-89 and MTR-90) only support 9600 or 19200 on power-up. I have a patched version of MTR-90 that replaces the 19200 baud DIP switch option with 38400 baud. Price is \$15 in a CMOS part (recommended for 4MHz operation and to cut power consumption), or \$10 in a slower NMOS part.

The basic Superset with 4 fonts is \$69.95; the SUPERFONT expands it to 8 fonts for \$29.95. The SUPERCLOCK adds battery backup for the clock/calendar, and 2K of non-volatile RAM for screen save/restore and window operation for \$29.95. Customers can upgrade earlier versions at no charge by simply sending in their old ROMs (add \$10 per EPROM if your version does not use a 27C256 program ROM).

Write-Hand-Man Enhancements

Write-Hand-Man, the popular "side-kick" program for the H8/H19 or H89 has gotten even better. WHM lets you interrupt any CP/M program to instantly access a calculator, and returns you instantly to your interrupted program, right where you left off.

The new WHM makes use of the extra capabilities of a Superset-equipped H19/H89. Screen save/restore operations are instantaneous, and perfectly save and restore the entire terminal state (even the unreadable modes, like alternate/shifted keyboard status). The Superset version also cuts memory usage; 2K less than the earlier WHM.

The real news is EDIT, the WHM editor. It's a full-screen text editor, with the same command format as PIE. Like all WHM applications, EDIT can be called any time, from within another CP/M program. But in addition to editing a file on disk, EDIT can get its input from the screen itself; thus it can "load" the interrupted program's screen, edit it, print it, or save it on disk!

For example, you're in Microsoft BASIC and need to make some changes. LIST the lines so they are on the screen. Call WHM, and EDIT them with the convenience of a full-screen editor. When done, print them, save them on disk, or just return to BASIC and insert the edited lines directly!

The manual has been revised and expanded, and many small enhancements have been made in the applications as

well. Write-Hand-Man is \$49.95, or \$10 for upgrades if you send in your old disk. Order both products (Superset and Write-Hand-Man) from Lee Hart, 28612 Middle Crossing Road, Dowagiac, MI 49047, phone (616) 782-3980.

The Investment Simulation Program

(ISP) is a specialized spreadsheet for the IBM-compatible computer systems. ISP computes internal ROR's, equity ROR's, life-cycle cost, capitalized present value, and other conventional criteria using over 5,000 variables and up to 200 time periods. Computed results can be printed to hard copy, or saved to ASCII files for use in word processors or importing into most spreadsheets. The program diskette carries six different worksheets with example data for personal and business investments. Typically, a user need only load a worksheet, correct the title, time frame and data to suit a specific project.

Purchase includes a 51-page handbook with the 5.25-inch program diskette (not copy protected). The handbook shows how to use discounted cash flow analysis to develop business and capital budgeting plans; measure economic and equity efficiencies of projects; make buy, build or lease decisions; estimate property or business values based on future costs and revenue forecasts; estimate economic limits for new construction projects; set product, service and bid prices to provide a specified rate of return, and more.

Equipment Requirements: An IBM-PC or true compatible — including PS/2, PC, XT, and AT with 256 KRAM and DOS 2.0 or later. One 5.25-inch disk drive, a monitor, and a printer with 132-column or more compressed mode operation. Price: \$49.95 for the IBM-compatible version, plus 10% for shipping continental U.S. To order, or for more information, write: Pyxis Software, P.O. Box 18016, Colorado Springs, CO 80935. *



Want New And Interesting Software?
Check Out HUG Software

Classified Ads

REACH THOUSANDS OF DEDICATED HEATH/ZENITH COMPUTER OWNERS!!

Classified Ad Rates

There is a minimum of \$5.00 for 10 words or less. Each additional word is \$.40. Payment for all classified advertising must be in advance.

Display Classified is available at \$35.00 per column-inch. Minimum one inch. Display Classified is commissionable to recognized agencies.

How To Count Words

Count one word each for initials, standard abbreviations, whole numbers, name, address, city, state, zip, area code and telephone number. Dimensions (such as 6 X 9) are counted as one word. Box or department numbers are counted as one word each. All classified ads (not display) are set in the same size type. The first several words, depending on the ad, are set in all caps.

Continuity Discounts

Run your ad at least 3 times during the year; and receive a discount off the regular price.

Run 3 insertions — 3% off, 4 insertions — 4% off, 5 insertions — 5% off, . . . 12 insertions — 12% off.

Issue & Closing Date

Issue Date	Closing Date	Issue Date	Closing Date
January	November 15	July	May 15
February	December 15	August	June 15
March	January 15	September	July 15
April	February 15	October	August 15
May	March 15	November	September 15
June	April 15	December	October 15

REMark Magazine has the right to refuse any ad for any reason.



Classified Order Blank

First	10	words	at	a
cost	of	\$5.00	each	insertion.
11-\$5.40	12-\$5.80	13-\$6.20	14-\$6.60	15-\$7.00
16-\$7.40	17-\$7.80	18-\$8.20	19-\$8.60	20-\$9.00
21-\$9.40	22-\$9.80	23-\$10.20	24-\$10.60	25-\$11.00
26-\$11.40	27-\$11.80	28-\$12.20	29-\$12.60	30-\$13.00
31-\$13.40	32-\$13.80	33-\$14.20	34-\$14.60	35-\$15.00
36-\$15.40	37-\$15.80	38-\$16.20	39-\$16.60	40-\$17.00
41-\$17.40	42-\$17.80	43-\$18.20	44-\$18.60	45-\$19.00
46-\$19.40	47-\$19.80	48-\$20.20	49-\$20.60	50-\$21.00

Enclosed is a check or money order of \$_____ for _____ words.

(Minimum order: 10 words for \$5.00. Each additional word \$.40.)

Please insert this advertisement in the _____ issue.

Signature _____

(Please type or print)

Name _____ Phone _____

Company _____

Address _____

City _____ State _____ Zip _____

MAILING ADDRESS: Make check or money order payable to: Rupley's Advertising Service, Dept. — REM, 240 Ward Avenue, P.O. Box 348, St. Joseph, MI 49085, (616) 983-4550.

C SCIENTIFIC LIBRARIES Personal CSL \$85, Personal Statistics \$99, and Personal Matrix \$99. Personal libraries are subsets from CSL (440 functions) available for \$275, source codes optional \$75. Each routine is designed and documented for use by technical specialist. Call (408) 867-6575 or write Eigenware, 13090 La Vista Dr., Saratoga, CA 95070.

HEBREW TO ENGLISH TRANSLATION. Scientific, Commercial, Technical, Math. Shachar, Box 835, Dimona, Israel, Tel. (0)57-51307.

FOR SALE: Zenith Z-100 low profile, 768K, full color video RAM, two 360K drives, EASY-PC emulator, Zenith green screen monitor, lots of software, manuals. Jody, (919) 549-1213 (days). (919) 365-4715 (nights).

THIS COULD BE YOUR AD! For as low as \$5.00, you could advertise personal items in this space. And for as low as \$35.00, you could advertise commercial items. People read these ads, you did!

USED, BUT CHECKED and verified good, HS-218, Syquest, 10-meg cartridges. Original cost, \$99.95. Order yours today for only \$25 each. Call (616) 982-3838 EST.

Announcement!

HUG MEMBERS ONLY!!

The HUG-386 and HUG-386-C upgrade kits will be available shortly. Wheelin' Dealin' Jim has managed a super-fantastic deal on these two products for Heath Users' Group members who originally purchased an H-241 or H-248; *one-thousand two-hundred dollars* off the regular purchase price! That's right! If you originally purchased an H-241 or H-248, and you're a HUG member, you can get \$1200.00 off the regular retail price of either of these two upgrade kits!

The HUG-386 and HUG-386-C are upgrade kits that let you upgrade your H-241 or H-248 series computers up to a full H-386. Now, how do you determine which upgrade kit to buy? The H-386-C includes a dual winchester/floppy controller, while the H-386 does not include any disk controller. Since the old H-241 controller is not '386 compatible, you'll probably want the "C" model if you're upgrading a '241. If you're upgrading a '248, your decision will depend on whether you need a new dual controller or not.

Here are the three ways you can order your upgrade:

Write-In Orders

- Non-HUG members *can* order by including payment (with the upgrade kit order) for one year's membership in the Heath Users' Group.
- All orders should be submitted to the Heath Users' Group.
- Each order must indicate the model number of the upgrade kit desired, and which computer kit it was purchased for.
- Each order must have the persons HUG ID number written on it.

Phone-In Orders

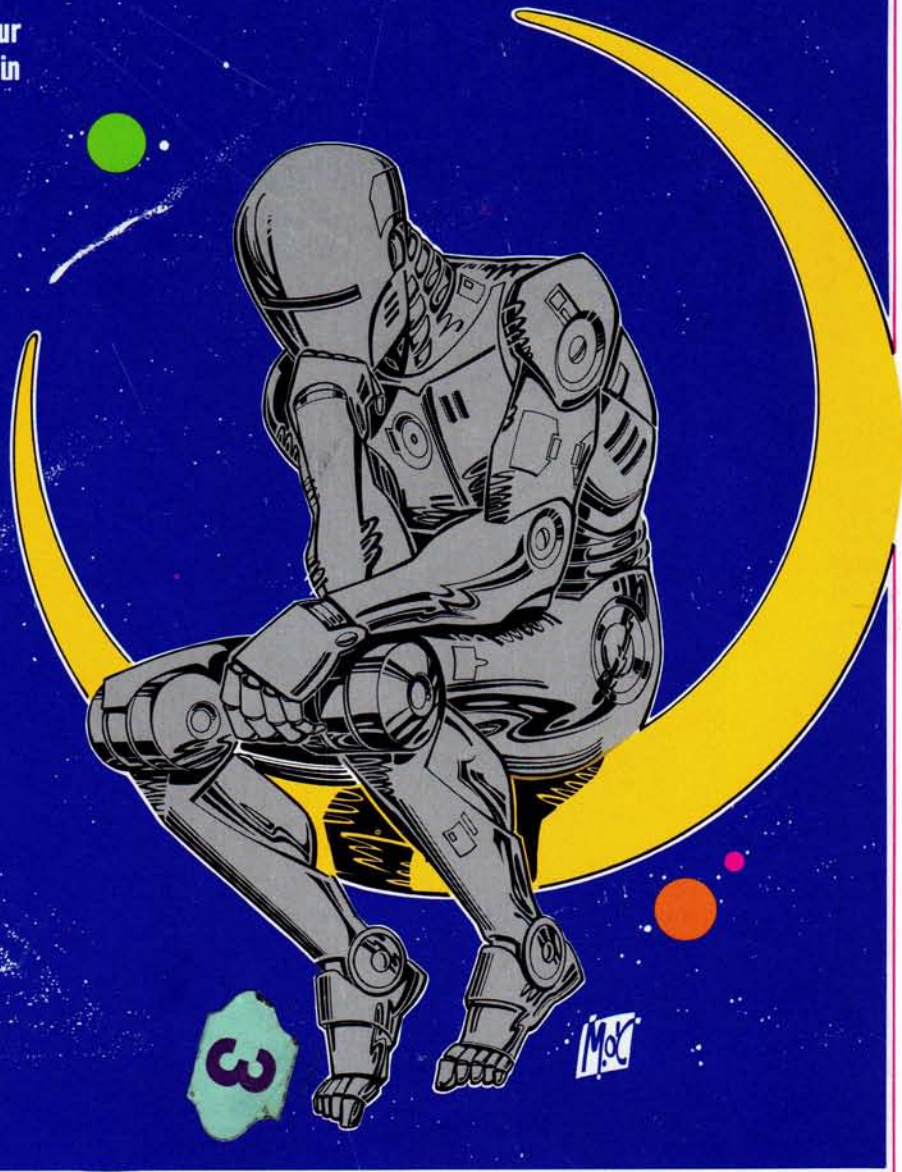
- Non-HUG members *can* order by first ordering a one year's membership in the Heath Users' Group.
- All orders must be phoned in to (616) 982-3838 from 8 AM to 4:30 PM EST.
- Each order must indicate the model number of the upgrade kit desired, and which computer kit it was purchased for.
- The person ordering must supply his/her current HUG ID number.

Heath/Zenith Computer Store Sales

- Non-HUG members *can* purchase an upgrade kit by first purchasing a HUG membership from the store.
- Orders for the upgrade kit can be taken in the normal fashion.
- Each order must have the buyer's HUG ID number on it.
- Each order should indicate which computer kit the upgrade was purchased for.



Don't be left alone with your
Heath/Zenith Computer, join
HUG today!!



P.O. Box 217
Benton Harbor, MI 49022-0217

BULK RATE
U.S. Postage
PAID
Heath Users' Group

POSTMASTER: If undeliverable,
please do not return.

P/N 885-2107