

REMark®

Volume 9, Issue 5 • May 1988

P/N 885-2100 Issue 100

Getting Started With . . .
Our Beginner's Columns!
See Pages 9 and 47

\$2.50



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

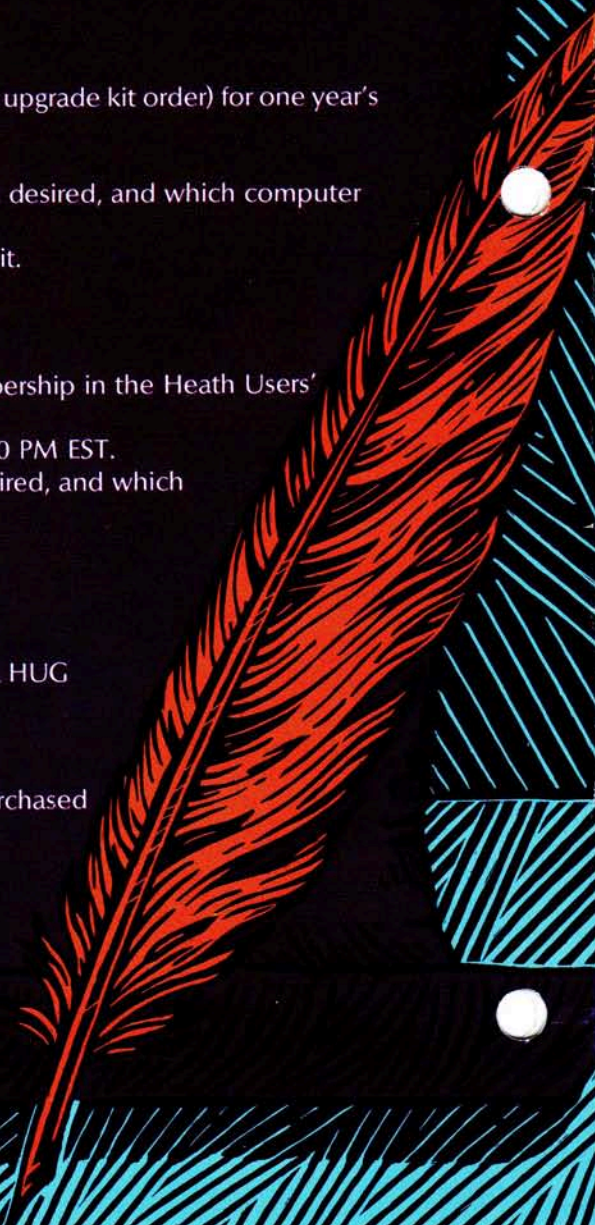
- Non-HUG members **can** order by including payment (with the upgrade kit order) for one year's membership in the Heath Users' Group.
- All orders should be submitted to the Heath Users' Group.
- Each order must indicate the model number of the upgrade kit desired, and which computer kit it was purchased 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.

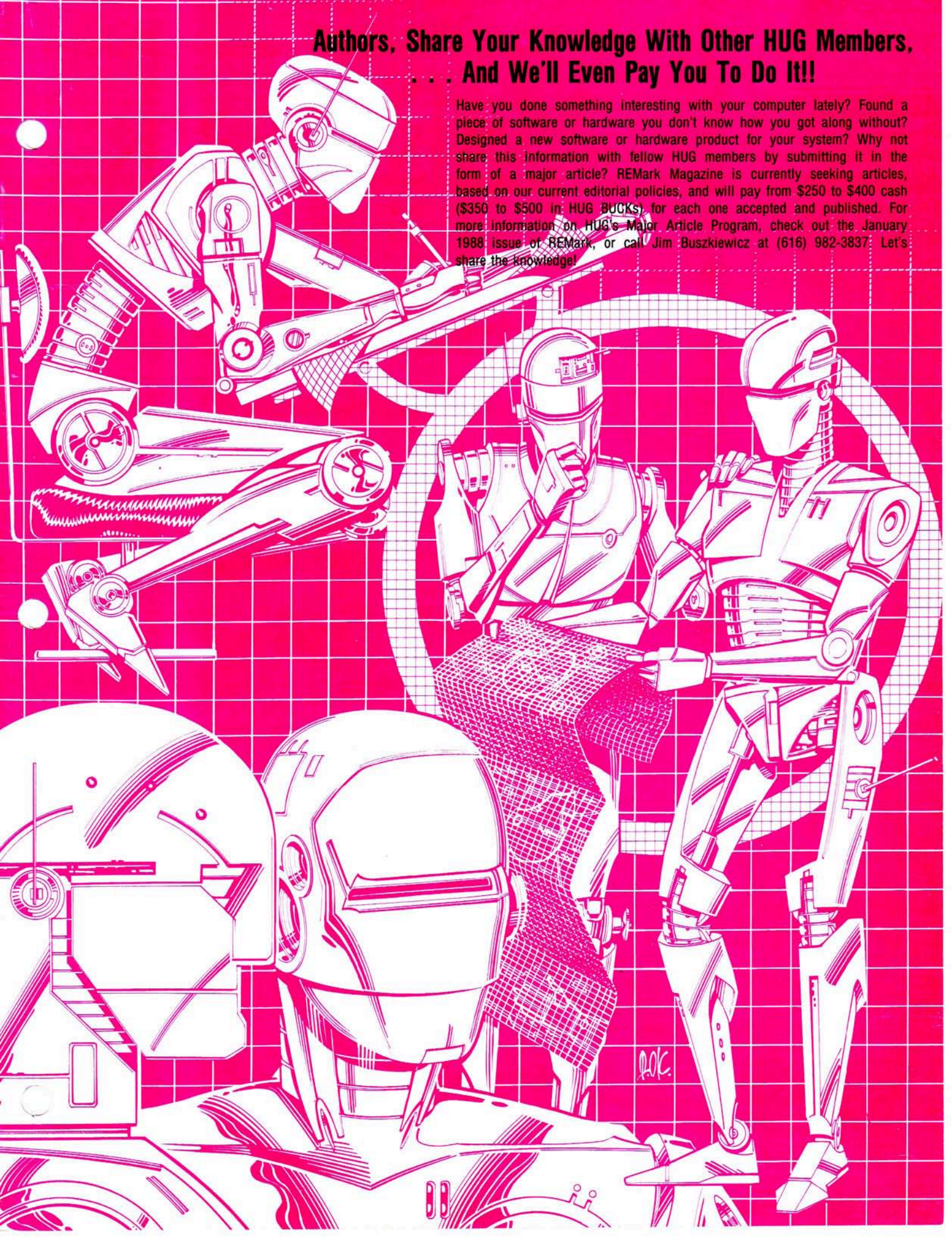
Heath/Zenith Computer Store Sales

- 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.



Authors, Share Your Knowledge With Other HUG Members, ... And We'll Even Pay You To Do It!

Have you done something interesting with your computer lately? Found a piece of software or hardware you don't know how you got along without? Designed a new software or hardware product for your system? Why not share this information with fellow HUG members by submitting it in the form of a major article? REMark Magazine is currently seeking articles, based on our current editorial policies, and will pay from \$250 to \$400 cash (\$350 to \$500 in HUG BUCKS) for each one accepted and published. For more information on HUG's Major Article Program, check out the January 1988 issue of REMark, or call Jim Buszkiewicz at (616) 982-3837. Let's share the knowledge!



D.K.

Heath/**ZENITH** Users' Group

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) 982-3838

Contributing Editor William M. Adney

Contributing Editor Joseph Katz

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



Heath/Zenith Related Products

Jim Buszkiewicz 7

Getting Started With . . . BASIC

Alan Neibauer 9

C__Power — Part 12

John P. Lewis 15

On The Leading Edge

William M. Adney 21

EGA, Desktop Publishing And ME

Eric L. Pang 27

ZPC Update #22

Patrick Swayne 31

HUG New Products 33

HUG Price List 34

Quattro Is One Better Than Uno-Dos-Tres
William M. Adney 36

Mainstream Computing
Joseph Katz 43

Getting Started With . . . Microsoft® Word
Jack W. Bazhaw 47

ENABLE — Part 6
George P. Elwood 51

Index of Advertisers

This index is provided as an additional service. The publisher does not assume any liability for errors or omissions.

Reader Service No.	Page No.
101 American Cryptronics	30
166 Designline	46
104 FBE Research Company, Inc.	6
*** HUG Authors	3
*** HUG MCP	68
*** HUG Members Only	2
*** HUG PBBS	67
107 Paul F. Herman	6
108 Hogware Company	6
137 Jay Gold Software, Inc.	66
111 KEA Systems Ltd.	45,63
136 Lindley Systems	66
114 Micronics Technology	63
117 Payload Computer Services	14
119 S&K Technology, Inc.	41
121 Scottie Systems	30
*** Seattle HUGCON	42
122 Secured Computer Systems	50
*** Veritechnology Elec. Corp.	8
154 WindowDOS Associates	26

On The Cover: Do you want the performance of a high-speed race car, instead of an old four-door family sedan? See Bill Adney's review of "Quattro" on Page 36.

FBE Products

For the H/Z-150, 160 Series

MegaRAM-150 — Modification kit allows memory board to be filled with 256K RAM chips (1.2 MByte). No soldering. Supplied with RAM disk software. **\$49.95**

ZP640 PLUS — Replacement PAL for standard memory board allows up to 2 banks of 256K and 2 or 3 banks of RAM chips to be installed for 640K or 704K maximum memory. **\$24.95**

COM3 — Replacement PAL allows installation of three serial ports (one an internal modem). Supplied with printer driver software for 3rd port. **\$39.95**

FBE Smartwatch

Calendar/Clock using Dallas Semiconductor's DS1216E SmartWatch module. Works with H/Z-110/120, 138/148, 150/158. Package includes SmartWatch with our software and documentation. Spacer kit (\$2) required for Z-100. **\$44.95**

For the H/Z-100 Series

ZMF100a — Modification package allows installation of 256K RAM chips in older Z-100 without soldering. Works only with old-style motherboard. **\$65**

ZRAM-205 — Kit allows 256K RAM chips to be put on Z-205 memory board to make 256K memory plus 768K RAM disk. Requires soldering. PAL (\$8) required for new motherboard. **\$49**

For the H/Z-89, 90 Series

SPOOLDISK 89 — 128K byte electronic disk and printer interface/spooler card. **\$195**

H89PIP — Dual port parallel interface card. Use as printer interface. Driver software included. **\$50 Cable \$24**

SLOT4 — Extender card adds 4th I/O expansion slot to right side bus. **\$47.50**

FBE

FBE Research Company, Inc.

P.O. Box 68234, Seattle, WA 98168

(206) 246-9815, M-F 9-5

UPS/APO/FPO Shipping Included.
VISA or MasterCard Accepted.

The Z100 Desktop Publishing Solution

If you're interested in producing professional looking documents on your Z-100, with the ability to intermix text and graphics we've got some solutions for you. Our Z-100 Publishing Package will let you use PFS:First Publisher under ZPC, with total mouse support. And it all runs on a stock Z-100 (no Scottie board required). And that's not all. We'll show you how to modify ZPC to run twice as fast as usual in high-res graphics mode. In fact, benchmark tests show that First Publisher runs *FASTER* on an 8mHz Z-100 with our package, then it does on a 4.77mHz IBM-PC with EGA card.

Z-100 Desktop Publishing Package \$ 119.00

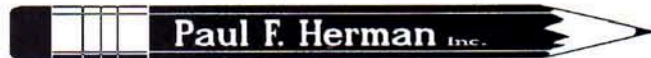
includes: PFS:First Publisher and patch info.
ZPCMOUSE driver with ZPC modifications.
Complete instructions.

Publishing Package w/Logitech Mouse \$ 191.00

System Requirements...

ZPC Version 2 (available from HUG) * 768K RAM Memory

From the leader in Z-100 Graphics...



Software Graphics Tools

3620 Amazon Drive

New Port Richey, FL 34655

800-346-2152

813-376-9347 in Florida



ShowOff lets you create outstanding graphics for presentations and reports with:

High Resolution 640 x 480
92 fill colors, 92 patterns
25 text styles at any angle
Capture and enhance any Z100 screen
Mouse and digitizer support

Now even better, the NEW ShowOff 2 has:

Desktop publishing compatibility
Read and write Macintosh files
Use clip art from commercial/public sources
Individual pixel editing with zoom edit
Text improved with sizes up to full screen
Smooth curve fitting -- true spline
Also available: new Art Gallery of hires graphics

HOGWARE COMPANY

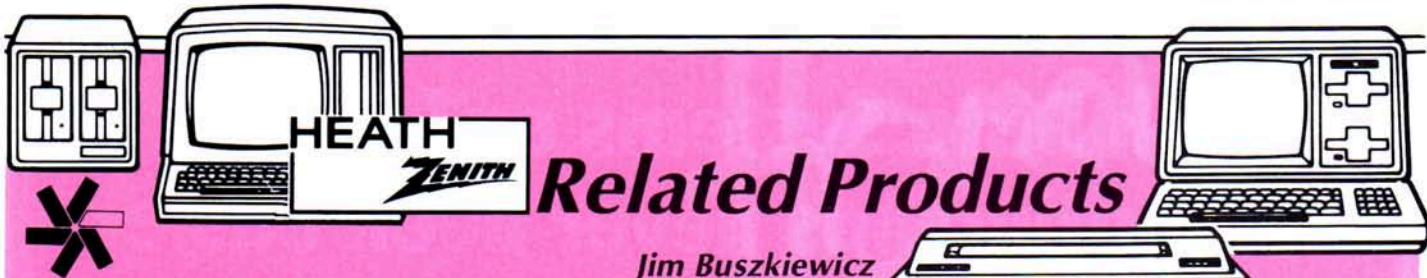
470 BELLEVIEW ST LOUIS MO 63119

(314) 962-7833

ShowOff \$95 ShowOff with Logitech mouse \$185 (s/h \$4)

ShowOff demo \$4 Art Gallery Sampler \$15 (s/h \$2)

ShowOff minimum requirements: Z100, 384K, MSDOS 2.0, Full video memory



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.

Axonix Corporation announced the portable streaming tape backup system for laptop computers.

The ThinStream XT/AT provides needed and convenient backup to hard disk files on PC-compatible laptops from Zenith.

The ThinStream cables to the floppy disk port on the laptop, and transfers data at 250K baud for the XT model and 500K baud for the AT version. For example, the AT version takes less than 20 minutes to backup 20 MBytes of data onto tape.

The ThinStream provides 40 MB of formatted data backup on a standard DC 2000 type mini-cartridge, and comes with friendly utility software which includes data access security protection. Data transfers utilize a unique algorithm which enables reading of more than 300,000 cartridges before experiencing a hard error.

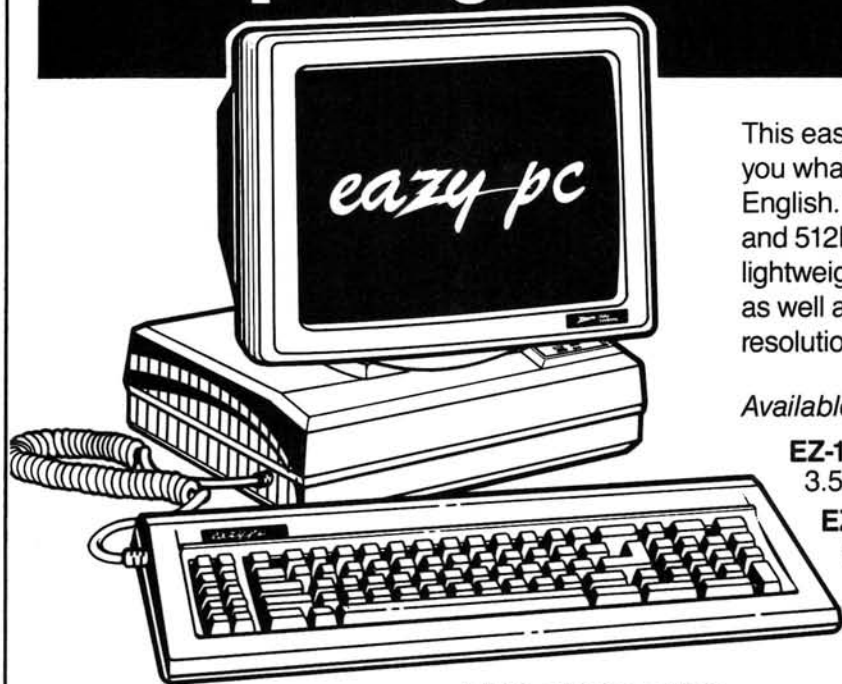
Lightweight and compact, the ThinStream weighs less than 3 pounds and measures 2.3" H x 4.5" W x 7.7" L. Sharing the ThinStream among groups of both in-office and remote laptop and desktop PCs makes the ThinStream very cost effective.

Because data losses are inevitable, particularly on laptops which are "well traveled", the ThinStream offers a much needed resource at an affordable price. Suggested list prices: \$849 XT model. For more information, contact your local Axonix dealer or Axonix Marketing at:

(801) 466-9797
2257 South 1100 East
Salt Lake City, UT 84106



Zenith Data Systems eaZy pc Computing has never been easier



This easy-to-use IBM PC compatible tells you what to do and when – on-screen in plain English. The EaZy PC comes with MS-DOS and 512K RAM expandable to 640K. Plus it's lightweight and has a professional keyboard, as well as a 14" monitor that provides high resolution text and graphics.

Available in three models:

EZ-1 with one 720K
3.5" drive ONLY **\$799**

EZ-2 with dual 720K
3.5" drives ONLY **\$959**

EZ-3 with 20 MB
Winchester ONLY **\$1359**

Also comes with our **FREE Partnership Pack** support package worth \$500 in savings!

**HUG DISCOUNTS
DO NOT APPLY.**

Available NOW at Heath/Zenith Computers & Electronics Centers in the U.S.

ARIZONA - Phoenix, 85017
2727 W. Indian School Rd.
602-279-6247

Tucson, 85711
5616 E. Broadway
602-745-0744

CALIFORNIA - Anaheim, 92805
330 E. Ball Rd.
714-776-9420

San Jose (Campbell), 95008
2250 S. Bascom Ave.
408-377-8920

El Cerrito, 94530
6000 Potrero Ave.
415-236-8870

San Diego (La Mesa), 92041
8363 Center Dr.
619-461-0110

Los Angeles, 90007
2309 S. Flower St.
213-749-0261

Pomona, 91787
1555 N. Orange Grove Ave.
714-623-3543

Redwood City, 94063
2001 Middlefield Rd.
415-365-8155

Sacramento, 95825
1860 Fulton Ave.
916-486-1575

Woodland Hills, 91364
22504 Ventura Blvd.
818-883-0531

COLORADO - Denver
(Westminster), 80003
8725 Sheridan Blvd.
303-429-2292

FLORIDA - Hialeah, 33012
4705 W. 16th Ave.
305-823-2280

Jacksonville, 32211
9426 Arlington Expressway
904-725-4354

Plantation, 33317
7173 W. Broward Blvd.
305-791-7300

Tampa, 33614
4019 W. Hillsborough Ave.
813-886-2541

GEORGIA - Atlanta, 30342
5285 Roswell Rd.
404-252-4341

HAWAII - Honolulu
(Pearl City), 96782
98-1254 Kahaumanu St.
808-487-0029

ILLINOIS - Downers Grove, 60515
224 Ogden Ave.
312-852-1304

Skokie (Chicago), 60076
3606 W. Dempster
312-583-3920

INDIANA - Indianapolis, 46220
2112 E. 62nd St.
317-257-4321

KANSAS - Kansas City
(Mission), 66202
5960 Lamar Ave.
913-362-4486

KENTUCKY - Louisville, 40243
12401 Shelbyville Rd.
502-245-7811

LOUISIANA - New Orleans
(Kenner), 70062
1900 Veterans Memorial Hwy
504-467-6321

MARYLAND - Baltimore, 21234
1713 E. Joppa Rd.
301-661-4446

Rockville, 20852
5542 Nicholson Lane
301-881-5420

MASSACHUSETTS -
Peabody, 01960
242 Andover St. (Rt. 114)
617-531-9330

Wellesley, 02181
165 Worcester Ave. (Rt. 9)
617-237-1510

MICHIGAN - Farmington Hills, 48018
29433 Orchard Lake Rd.
313-553-4171

East Detroit, 48021
18149 E. Eight Mile Rd.
313-772-0416

St. Joseph, 49085
2387 Lake Shore Drive
616-982-3215

MINNESOTA - Minneapolis
(Hopkins), 55343
101 Shady Oak Rd.
612-938-6371

St. Paul, 55106
1645 White Bear Ave.
612-778-1211

MISSOURI - St. Louis
(Bridgeton), 63044
3794 McKelvey Rd.
314-291-1850

NEBRASKA - Omaha, 68134
2311 N. 90th St.
402-391-2071

NEW JERSEY - Ocean, 07712
1013 State Hwy. 35
201-775-1231

Fair Lawn, 07410
35-07 Broadway (Rt. 4)
201-791-6935

NEW YORK - Amherst, 14226
3476 Sheridan Dr.
716-835-3090

Jericho, L.I., 11753
15 Jericho Turnpike
516-334-8181

Rochester, 14623
937 Jefferson Rd.
716-424-2560

N. White Plains, 10603
7 Reservoir Rd.
914-761-7690

NORTH CAROLINA -
Greensboro, 27407
4620-C W. Market St.
919-299-5390

OHIO - Cincinnati
(Springdale), 45246
131 West Kemper Rd.
513-671-1115

Cleveland, 44122
28100 Chagrin Blvd.
216-292-7553

Columbus, 43229
2500 Morse Rd.
614-475-7200

Toledo, 43615
48 S. Byrne Rd.
419-537-1887

OKLAHOMA -
Oklahoma City, 73139
7409 South Western
405-632-6418

OREGON - Portland
(Tigard), 97223
10115 S.W. Nimbus Ave.
503-684-1074

PENNSYLVANIA - Frazer, 19355
630 Lancaster Pike (Rt. 30)
215-647-5555

Philadelphia, 19149
6318 Roosevelt Blvd.
215-288-0180

Pittsburgh, 15235
3482 Wm. Penn Hwy.
412-824-3564

RHODE ISLAND - Warwick, 02886
555 Greenwich Ave.
401-738-5150

TEXAS - Dallas, 75218
12022C Garland Rd.
214-327-4835

Fort Worth, 76116
8825-A Green Oaks Rd.
817-737-8822

Houston, 77008
1704 W. Loop N.
713-869-5263

San Antonio, 78216
7111 Blanco Rd.
512-341-8876

UTAH - Salt Lake City
(Midvale), 84047
58 East 7200 South
801-566-4626

VIRGINIA - Alexandria, 22303
6201 Richmond Hwy.
703-765-5515

Virginia Beach, 23455
1055 Independence Blvd.
804-460-9997

WASHINGTON - Seattle, 98109
505 8th Ave. N.
206-682-2172

WISCONSIN - Milwaukee
(Wauwatosa), 53226
845 N. Mayfair
414-453-1161

Phone
orders
accepted.



Your **TOTAL** computer
center for Service

- Support • Software
- Accessories • User Training
- Competitive Prices

HZC-402

Getting Started With . . . BASIC

Alan Neibauer

11138 Hendrix Street
Philadelphia, PA 19116

Every time you use your computer you're using a computer program — a list of instructions that tell the computer what to do. Someone wrote the program, whether its the disk operating system (DOS) itself, a word processing program, or even a game. The program was written, converted into a form your computer understands, then loaded onto a disk.

So when you start your computer, you're actually transferring the DOS program from the disk into your computer's memory and the program takes control. When you type DIR and press Return, the DOS program interprets your request and shows you a list of programs and other files on your disk.

BASIC is just another computer program. But it's a computer program that lets you write programs. It lets you do so by using a special language — a series of words and phrases — that it converts into a form your computer understands so it can follow your instructions.

Now, BASIC is certainly not the first or only program of its type. In fact, in the history of computers, BASIC is still a relative newcomer. Programming languages, such as Fortran and Cobol, were widely used long before BASIC became popular. But BASIC offers some advantages that other

languages do not. It is easy to use, inexpensive, and you can learn how to write programs easily and quickly.

However, that doesn't mean that BASIC isn't as powerful as these other languages. On the contrary, BASIC is a full fledged computer language that you can use to write even the most complex and sophisticated programs.

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.

BASIC has these structures and if you keep them in mind, you'll see that programming is really easy. It is just a matter of deciding which structure you need for the job you want done, then writing the instructions using the BASIC language — the words BASIC understands and can

translate into a form your computer can follow.

This article is designed to make you familiar with the fundamental concepts of computer programming and the BASIC language. You'll learn how to use several BASIC language words and how to write, execute, and save some simple BASIC programs. But don't expect it to show you everything about BASIC or programming. Programming can be an exciting experience, but often a frustrating one. Like anything in life, the more you do it, the better you become. While it takes much practice to become a professional programmer, you'll be able to write programs yourself in no time at all.

Sit down at your computer, read this article and follow the instructions presented. Then, if you like programming, read other articles in REMark, purchase a book on BASIC at a local book or computer store, or take an introductory course at a local college.

But now let's begin.

1. Start your computer.

Make sure the program BASICA is on the disk. Do this with the DIR command. Type DIR and press Return. If BASICA is

not listed, find the disk that contains it and insert it in drive A.

BASICA by the way is the name given to one particular form of the BASIC programming language. You may also hear it referred to as GW-BASIC. So if you can't find BASICA, look for some other program name that resembles BASIC.

2. Type BASICA and press Return.

DOS finds the BASICA program on your disk, loads it into the computer's memory, and passes control on to it. Your screen should look something like Figure 1. The prompt OK on the screen means that BASICA is in control and ready to accept your instructions. Now that you've started BASICA, you are ready to write BASIC programs.

The line at the bottom of the screen shows ten common BASIC commands and function keys that perform them. So to perform the LIST command you just press the function key labelled F1. You could also type LIST. More on that soon.

All BASIC instructions have a certain form: a line number, a BASIC command word, and something that the command works on, such as:

```
10 PRINT "HELLO"
```

This is line number 10, the BASIC command is PRINT, and it acts on the word Hello. The line numbers are used so BASIC knows the order of the steps. This line, by the way, displays the word Hello on the screen.

Let's write a computer program using the BASIC commands LET and PRINT. LET assigns a value to a variable and PRINT displays information on the screen.

Sequence

3. Type the following two lines.

```
10 Let A=10  
20 Print A
```

Press the backspace key if you make a mistake. Press Return after each line. Once you press Return you can't use backspace to correct mistakes on the line. If you notice a mistake after you press Return, just type the entire line all over again.

Make sure you press Return after typing the second line. BASIC commands, by the way, can be typed either uppercase or

lowercase. I'll explain this program after the next step.

4. Press the F2 key to execute the program, or perform the instructions you've just typed. (You could also type the word RUN and then press Return. The arrow after the word RUN at the bottom of the screen means that the Return key will be performed when you press F2.)

RUN is the command that tells BASIC to perform the instructions you've entered. So, in this case, the number 10 will appear, your screen should look like this:

```
10 Let A=10  
20 Print A  
RUN  
10  
OK
```

Take a close look at the two line program. Line 10 tells the computer to let the variable A have the value of 10. Now whenever the program "sees" an A, it gives it the value of 10. Line 20 uses the print command. This tells the computer to display on the screen the value of whatever follows. Well, A follows it and A has the value of 10. So 10 appears.

You can number your programs any way you want. But its best to start off using increments of ten. As you'll soon learn, this will let you add lines in between.

Since that's the last line in the program, the program ends and the BASIC OK prompt reappears.

If you made a mistake typing your program, the line containing the error may appear with the cursor at the error. For now, just press Return, retype the line, then press F2 again.

Before going on, a few words about variables. A variable is a name that gets assigned a value. It's common practice with BASIC to limit variable names to just two characters starting with a letter. So A, AC, and Z6 are all legitimate variable names. But 6G or A B are not legal — one starts with a number, the other has a space. They are called variables because their value can change within a program.

5. Type 15 A=5, then press Return. Your program now has three lines, 10, 15, and 20. Although they appear out of order on the screen, let's see what happens when we run the program.

6. Press F2 to run the program. The number 5 appears, not 10. That's because the three lines, in number order, now say this: Let the variable A equal 10, let the variable A equal 5, display the value of A on the screen. The value of A was 5 when the print line was executed or performed. It was changed from 10 in the second line (number 15) of the program.

7. Press F1 to display the word LIST on the screen, then press Return. The three lines appear again, but this time in order. The LIST command tells BASIC to display the instructions. Since there is no arrow next to the word LIST at the bottom of the screen, you have to press Return yourself.

Keep in mind the very important difference between LIST and RUN. RUN performs or executes your instructions. LIST simply displays the instructions you've entered, in line number order.

Notice that line 15 is now where it belongs, between 10 and 20. When you list a program, it places the lines in order of the line number, not in the order you type them in.

Let's now add another PRINT line to the program.

8. Type 30 PRINT "A". Be sure to enclose the A in quotation marks this time.

9. Press F2 to run, or execute, your program. The screen should show the number 5 on one line and the letter A on the next.

There are two things that the PRINT command can display. In line 20, you instructed the computer to print the value of the variable A. Since A was not in quotation marks, it printed the last value assigned to it. But in line 30, you told BASIC to display the letter A because you enclosed it in quotation marks. You can also print a third type of item, a number directly. For example, the command PRINT 35 would display the number 35 on the screen.

Note that after every print command, the computer does a carriage return and line feed, often written as CR/LF. This means that the cursor goes down to the next line and to the left of the screen before printing something else, like pressing the carriage return key on a typewriter or the Return key on your keyboard.

What if you didn't want the computer to do a CR/LF after the print command? You may want to print several items on one line. You can do this easily.

Let's modify the program you've just written. You can change, or edit, a program in several ways. For instance, to delete or erase one entire instruction line just type its number and press Return. The line may still appear on the screen, but it will not appear when you list the program or be performed when you run it.

(You can also edit a program line as it appears on the screen. Press the up arrow key to place the cursor on the line you want to edit. Enter new characters or delete others by pressing the key marked Del. Then press Return when you are done. You must press Return or the changes you made on the line will not be accepted even though they appear on the screen.)

10. Type 30, then press Return. You've just deleted line 30 from the program.

You can also change a line by typing the same line number, but with new instructions.

11. Type 15 A = 34 and press Return. You now have two line 15s on the screen. Which one do you think will be performed?

12. Press F2 to run the program. You'll see the number 34 on the screen, not 5. That's because the new line automatically replaced the other one with the same line number.

13. Prove this by listing the program — press F1, then press Return. The new line 15 appears in the listing.

14. Type the following line carefully, then press Return.

```
20 PRINT "A";A
```

This line now combines two types of printed matter, something in quotation marks that will print exactly as they appear (called a literal) and the value of the variable A. The semicolon just separates the two types of elements.

15. Press F2 to run the program. The line appears A= 34.

These examples demonstrate the concept of sequence. The computer per-

forms each of the instructions you've entered in the sequence of line numbers.

Selection

Now let's try selection using the IF THEN command. The general pattern looks like this:

```
IF condition-is-true THEN perform-this
  instruction
```

Let's erase the entire program you've just written and start a new one.

16. Type NEW, then RETURN. Now try to list your program. Nothing's there! The NEW command erases all of your program from memory, not from the screen. While you might still see it listed above the NEW command, it's really been cleared from the computer's memory and cannot be listed or run.

Get into the habit of using the NEW command when you're completely done with one program and want to start another.

17. Now type the following program.

```
10 Let A=10
20 If A=10 then PRINT "A is ten"
```

18. Run the program. The words 'A is ten' appear on the screen. Let's see what these lines do. Line 10 assigns the value of 10 to the variable A. Line 20 says this — if A has a value of 10, then print the words "A is ten." The condition part of the IF statement is A=10. If the condition is true, that is, if A does in fact equal 10, then BASIC performs the statement following the word THEN. Only one statement is allowed here. On the other hand, if the condition is false (in this case, A does not equal 10), BASIC skips the statement following THEN and goes to the next line number if there is one.

The conditions can be any of the following:

- < 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 or equal to it)
- >= Greater than or equal to as in B >= 0 (B is greater than 0 or equal to 0)

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 GOTO, the FOR NEXT, or the WHILE WEND commands. Each of these have their own uses and powers.

Use the FOR NEXT method when you know the exact number of times you want to repeat a line. Use WHILE WEND when you want the computer to determine the number of repetitions. The GOTO command is another way to repeat instructions, or to merely change the sequence of events. Let's save that for last.

19. Erase the program from memory by typing NEW and pressing Return.

20. Now type the following program.

```
10 FOR X=1 to 5
20 PRINT x, " Hello"
30 NEXT X
```

21. Run the program. Your screen should look like this:

```
1 Hello
2 Hello
3 Hello
4 Hello
5 Hello
```

The general pattern of the FOR command is:

```
FOR Variable = starting-number TO ending
  -number
(BASIC lines)
NEXT Variable
```

The starting and ending numbers can be any whole (integer) numbers, as long as the ending number is higher than the starting one. Because this is complex, let's look at this program in a little more detail.

When the program begins, it assigns the starting value to the variable X, called the control variable. It then performs any lines between the FOR and NEXT instruction. In this case, it prints the value of X, number 1, and the word Hello. When BASIC reaches the NEXT instruction it goes back to the FOR line and adds 1 to the value of X, making it 2. It checks to make sure the value of X is not greater than the ending value and, if not, repeats the line. It does this again to assign X the value of 3, then again to assign it 4, then again to assign it 5. Finally, it sees that adding one to X another time would make it greater than


```

GW-BASIC 2.02
(C) Copyright Microsoft 1983,1984
GW-BASIC version 2.13
(C) Copyright Zenith Data Systems 1984
60650 Bytes free
Ok
-

```

```

1LIST 2RUN 3LOAD 4SAVE 5CONT 6, "LPT1" 7TRON 8TROFF 9KEY 0SCREEN

```

Figure 1

the ending value. So it stops the repetition.

The space between the number and word Hello, by the way, is caused by the comma in the PRINT line. You already know that you use a semicolon to separate different elements on a PRINT line. You can also use the comma. However, the comma serves as a TAB command, inserting extra spaces between the two elements that it separates.

Let's look at the other repetition statement, WHILE WEND.

22. Type NEW and press RETURN to erase the program.

23. Type the following:

```

10 Let A=1
20 WHILE A <=5
30 PRINT a, "Hello"
35 A = A + 1
40 WEND

```

24. Run the program. It prints exactly the same thing as the FOR program just entered. Here's how it works. Line 10 gives the variable A a value of 1. Then line 20 starts the repetition. It means "As long as A has a value less than or equal to 5, perform all of the state-

ments between this line and the WEND command."

Since A is equal to 1, it performs line 30 by displaying the number 1 and the word Hello. Then it performs line 35. Well, that's certainly a strange formula with the same variable name on both sides of the equal sign — something your old math teacher would cringe at. Actually, it is quite legal here.

In fact, line 35 is pretty powerful. It's called a counter. Each time this line is executed — with each of the five repetitions — the value of A is incremented, or increased by one. It means "assign the new value of A to be equal to the old value of A plus 1." So let's continue to follow the program.

Line 35 takes the old value of A (1), adds 1 to it, and assigns the total to A. So A becomes 1 plus 1 or 2. Then, the WEND line returns the program back to line 20. Since A is still less than or equal to 5, the sequence is repeated — 2 Hello is printed, then A is increased by one again and made 3; 3 Hello is printed and A is made 4; 4 Hello is printed and A is made 5; 5 Hello is printed and A is made 6. But now when control is sent back to line 20, the condition is no longer true. A is not less than or equal to 5. SO, the repetition

stops and the program would perform the line following WEND if there were one.

The last way to repeat a BASIC instruction is to use the GOTO command. This command passes control directly to another line.

25. Type NEW and press Return to clear the current program.

26. Type the following:

```

10 Let A = 1
20 Print A
30 Let A = A + 1
40 GOTO 20

```

27. Press F2 to run the program, but don't expect it to stop.

28. Press and hold down the key marked CTRL, then as you keep it down, press the Scroll Lock key. This is called Break and it stops the execution of a program.

Can you figure out what went wrong with the program? Actually nothing. BASIC did exactly what you told it to do. Let's see.

Line 10 assigns A the value of 1. Line 20 prints the value of A on the screen, then in line 30, A is incremented to 2. Line 40

means "go back to line 20." So the very next line executed is 20 again and 2 is printed. A is incremented again in 30, then 40 sends control back again to 20. This is repeated over and over again until you stop the execution of the program.

This is an example of poor programming called an "endless loop." But let's modify the program to make it end in some natural way.

29. Type the following:

```
40 IF A <= 100 THEN GOTO 20
```

30. Press F2 to execute the program. Now the numbers 1 to 100 are printed on the screen and the program stops. That's because we made the GOTO conditional. We've said only go back to 20 if the value of A is less than or equal to 100.

Saving Your Programs

Now let's save the program you just wrote. Saving a program means recording the program on the disk. When you save a program, BASIC automatically adds the extension BAS on the end, if you don't give it your own extension.

31. Type SAVE "PROGRAM" and press Return. The program you just wrote will be recorded under the name PROGRAM.BAS. You can name your programs anything you want, as long as the name is eight characters or less without any spaces. But be sure to enclose the name in quotation marks.

If you want to recall the program at some later date, use either of the following commands once you're in BASIC:

```
LOAD "PROGRAM"
```

or

```
RUN "PROGRAM"
```

When you load, or recall a program from the disk, you can then run or list it, it won't appear automatically. When you use the Run command, the program is first recalled, then executed immediately.

Before getting out of BASIC 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 line LET A=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. Print lines 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 an example, take a look at a concluding program in Figure 2.

also be characters, called string variables. To use a string variable, place a dollar sign after the variable name, as in A\$. Then, surround the value being assigned in quotation marks. So the following program:

```
10 A$="HELLO THERE"  
20 PRINT A$
```

prints the words HELLO THERE on the screen.

Figure 2

```
10 INPUT "ENTER A FAHRENHEIT TEMPERATURE", F  
20 C = 5 / 9 * (F - 32)  
30 PRINT F;" DEGREES FAHRENHEIT EQUALS ";C;" DEGREES CELSIUS"
```

This program has one input line (10), one process line (20), and one output line (30). Line 20 just performs the calculation that converts degrees fahrenheit to celsius. Line 30 is the output line that displays a combination of words and values. The value of the variable F will be displayed followed by the words DEGREES FAHRENHEIT EQUALS. Then the value of the variable C will appear followed by the words DEGREES CELSIUS. The value of C is calculated in line 20.

Line 10 may be the only one totally unfamiliar to you, since it uses the BASIC command INPUT. The general format of INPUT is

```
INPUT "a prompt telling what you're  
asking for" variable-name
```

In this case, when line 10 is executed here's what happens. The words in the prompt appear on the screen and the program pauses. It's 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, BASIC assigns the number to the variable at the end of the INPUT line.

In this case, the number entered is assigned to the variable F. Then the calculation is performed and the results are displayed by line 30.

Try typing the program and running it. You'll see the prompt appear that you typed in line 10. Type some number, then press Return.

The variables used in this article are numeric variables because they are all represented with numbers. But variables can

Okay. That's really a lot for your first time with BASIC. So let's exit BASIC and get back into the DOS. Type SYSTEM and press Return.

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.

Certainly, programming is not for everyone. Most computer users today may never write a program in their life. That's because they're just using other programs written for them — programs like WordPerfect, WordStar or Lotus 1-2-3.

*



FINAL Z-100 SOFTWARE CLOSE OUT

Zenith packages with software, manuals and registration cards for the original Z-100 computer series (not for the IBM compatibles).

PART NUMBER	DESCRIPTION	LIST PRICE	SALE PRICE
MS-463-1	Z-Basic (16 bit) interpreter	\$175.00	\$10.00
MS-463-7	Multiplan	\$195.00	\$10.00
MS-253-1	BASIC-80 (8-bit)	\$175.00	\$10.00
CD-463-2	Condor File Manager	\$299.00	\$10.00
PK-100-4	All 4 listed above	\$819.00	\$38.00

*** UPGRADE ACCESSORIES FOR Z-100 *** SERIES COMPUTERS

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. \$244.00 Dual Drive Unit \$389.00

COLOR GRAPHICS UPGRADE. All memory chips (16 pieces 150 ns) required to update Z-100 Series computer for color. Installation instructions included. Order Memory Kit #100-64-16 \$45.00

MEMORY UPGRADE. All memory chips (9 pieces 150 ns) required to upgrade from 128K to 192K RAM. Installation instructions included. Order Memory Kit #100-64-9 \$25.50

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 alone or \$289.00 with 27 256K RAM chips included.

UCI MEMORY UPGRADE PAL CHIP SET. For Z-100's with the newer motherboard (181-4918 or greater). This set permits installation of 256K RAM chips on the motherboard. PAL set \$64.00 or \$293.00 with 27 256K chips required for 768K of memory on the motherboard.

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. Retail \$699.000, Payload \$477.00

UCI EASY 87. 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

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 With RAM installed and fully tested, 512K \$432.00, One MEG \$576.00, Two MEG \$864.00 Add \$35.00 for EasyDrive RAM Drive Software if desired.

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-Calendar \$125.00

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 \$578.00, 31 MEG \$598.00 System without Drive and Controller \$239.00

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

*** PANASONIC AND HEWLETT PACKARD PRINTERS ***

KPX - 1080i 120 cps 10"	\$219.00
KPX - 1090i 120 cps 10"	\$272.00
29cps NLQ	
KPX - 1092i 120 cps 10"	\$338.00
33 cps NLQ	
KPX-1592 180 cps 16.5"	\$421.00
38 cps NLQ	
KPX-1595 240 cps 16.5"	\$462.00
Hewlett Packard Laser Jet	
Series II printer with toner	\$1775.00

*** SEAGATE HARD DISK DRIVES ***

ST-225 20 MEG Winchester Hard Disk	\$259.00
With Western Digital Controller & Cables	\$289.00
ST-238 30 MEG, Requires RLL type controller	\$289.00
With RLL Controller & Cables	\$315.00
ST-4038 30 MEG High Speed for Z-200	\$479.00
ST-4053 40 MEG High Speed for Z-200	\$619.00
ST-4096 80 MEG High Speed with Software	\$919.00
ST-251 40 MEG High Speed Z-150/200	\$439.00

ALL DRIVES FORMATED AND TESTED.

*** HALF HEIGHT FLOPPY *** DISK DRIVES

MITSUBISHI M501	5.25" 48 TPI DS/DD	320K/360K	\$105.00
MITSUBISHI M504	5.25" 96 TPI DS/DD	360K/1.2 MEG	\$152.00
MITSUBISHI MF353	3.5" in 5.25" frame	720K	\$109.00
TOSHIBA ND04	5.25" 48 TPI DS/DD	320/360K	\$89.50
TOSHIBA NDO4	with grey faceplate	instead of black	\$89.50
TOSHIBA NDO8	1.2 MEG for Z-200 or AT Computer		\$119.00

*** POWER SUPPLIES AND CASES FOR DISK DRIVES ***

Rugged steel construction with heavy duty power supplies. Purchase with drives and we will install drives in case.

Single 5.25" unit for Half Height Drives	\$68.00
Dual 5.25" unit, for Half Height Drives	\$92.00
For WINCHESTER Drive, with fan	\$145.00

*** MITSUBISHI COLOR MONITORS ***

MODEL	MODE	DOT SIZE	RESOLUTION	LIST	PAYLOAD
XC1409C	CGA	0.4 MM	640x200	\$519.00	\$296.00
XC1410C	EGA	0.4 MM	640x350	\$650.00	\$419.00
XC1430C	EGA	0.31 MM	640x350	\$739.00	\$489.00
AUM 1371	EGA	HI-RES	920x580	\$849.00	\$519.00

*** UPGRADE ACCESSORIES FOR Z-148/150 *** SERIES COMPUTERS

VMM 150 Video Card Eliminator for Z-150/160 computers. Allows use of EGA color card \$45.00

SmartWatch from FBE Research. Installs in ROM Socket on CPU Board in Zenith computer series Z-100/148/150/158/160. This jewel of a product contains a ten year battery and keeps your computer informed of both time and date at each boot-up. Complete instructions and software included \$38.00

MEMORY KIT #150-256-18. Includes a ZPAL chip which allows use of 256K RAM chips included (18 pieces 256K 150 ns RAM chips). Kit increases 128k memory to 640K or 256K memory to 704K. All chips plug into your existing Zenith Memory Board. Unbelievable but true. For Z-150/160 only \$164.00

Winchester Hard Disk Drive Internal Set-up. Includes Winchester drive, controller/interface card, cables and all hardware. With 20 MEG (formatted) drive \$289.00 30 MEG \$315.00 May be installed in Z-148 using an Expansion Card sold below.

PTZ-148 Expansion Card for Z-148. Includes 2 expansion slots plus a clock/calendar. \$118.00

Winchester Hard Disk Drive External Set-up. Includes Winchester drive, controller/interface card, power supply and case with fan. With 20 MEG (formatted) drive \$434.00 With 30 MEG (formatted) drive \$460.00

SECOND INTERNAL DISK DRIVE with instructions and hardware \$105.00

ANCHOR INTERNAL MODEM. Fully Hayes compatible 1200/300 baud with powerful Com software included \$95.00 Same but 2400/1200/300 BAUD \$182.00 External 2400/1200/300 BAUD \$198.00



PAYLOAD COMPUTER SERVICES

15718 SYLVAN LAKE, HOUSTON, TEXAS 77062
PHONE (713) 486-0667



Please MAIL or PHONE your order today and expect prompt service. MASTERCARD and VISA gladly accepted with no additional charge. All hardware carries a 90 or more day warranty. Add \$5.00 to all prepaid orders for handling and shipping, we pay the balance. Texas Residents please add 8.0% sales tax. We accept purchase orders from schools, government and approved accounts.



C__Power

Part 12

John P. Lewis

6 Sexton Cover Road
Key Largo, FL 33037

The last C__Power article presented the reader with a usable, if rather cumbersome, editor. This month, we are going to add quite a bit of sophistication. After entering the included listing (Edtxt ver. 1.12), compiling the source code and eliminating the inevitable "typos", you will have a quite serviceable editor. Still substantially short of even a "bare bones" commercial version, but we are rapidly approaching that goal. The most important aspect of this project is not the acquisition of another editor, but the programming knowledge gained through following this series to fruition.

Another very real asset, gained through the study of this program (or any other comprehensive effort), is the acquisition of some particularly useful routines that can be used in your own creations. The screen handling routines alone have proved to be an extremely valuable addition to my "C" library (wait until you see the really fast screen printing functions in this issue).

If you are a "C__Power" veteran who has followed the series from its inception, or a reader who has joined us since Part 9 (the earlier issues dealt with a database prog-

ram utilizing arrays for its structure), then you should have gleaned quite a bit of knowledge about linked lists, pointers, structures and the dynamic allocation of memory. These topics are exemplified in the focus of the current articles.

We are going to cover some new ground in the present issue, that of writing characters directly to the screen memory in an effort to make the video update much quicker. The results have been very gratifying to say the least. Of course, we have to pay a price for the increase in speed and that is in the area of reduced (non-existent) portability. Since each character written to the screen, using the new routines, will involve addressing memory locations unique to the IBM and its compatible brethren, we have narrowed the scope of portability to those machines which emulate the IBM very closely. Another sacrifice is in the area of CGA/MONochrome detection, or should I say lack of same? In this article, we will provide the code to write to the screen memory, using a #define statement to differentiate between the two monitors. Next month, we will provide automatic detection of the video mode. I will cover this subject more thoroughly when we go over the listing.

For those programmers who would like to dress up their programs with some screen enhancements, such as: reverse video, boldface (increased character brightness), underlining, and blinking, you will find the additions to the "scrnlib.c" library very useful. These routines are used in "Edtxt ver 1.12" to display column and row (line) information on the twenty-fifth line, as well as adding a little window dressing to the menu. Since a pretty face is such an asset, we also included a "drawbox" function, used to draw borders around the menu screen, rather an attractive addition.

The most important addition to our editor has a rather narrow scope of application and was certainly the cause of more head scratching than any other routine. I'm talking about the "join_line()" and "add_ch()" functions. These two routines, in conjunction with the "del_ch()" and "ins_line()" functions, enable the reformatting of sentences and/or paragraphs. Not the Page 1 kind of operation that is needed in many programs, but a very valuable routine when string handling and manipulation is the order of the day. I'll have to confess that many of my routines are adaptations of functions that I've found in magazines like REMark or books on programming; however, my rather ex-

tensive library of books on "C" or "Pascal" had no prototypes on this subject to draw from, so I had to create these two routines from scratch. I'll make no claims for elegance or optimization of source code, but the bottom line is: they work!

To round out version 1.12 of Edtxt, you will find a printer driver that should work on any printer, but is rather unsophisticated. It will enable the user to make a hard copy of any document created with Edtxt, or most other editors for that matter. A replica of the screen image will be output to the printer. What more could you ask? I was hoping someone would be looking for a printing module that would justify text. A "pretty printer", if you will. You asked, I'll deliver in "C__Power — Part 13".

Let me qualify that last statement a bit before going on. Printers come in many shapes and sizes with a host of differences between them, hence a proliferation of specialized printer drivers. The focus of the "pretty printer" routine found in the upcoming "C__Power" article is a Radio Shack model 210 (daisy wheel) printer. Don't sneer and turn up your nose! The code can be altered to work on most printers that have the ability to change the character spacing (pitch) from software. I'll even show you how to do it from the #define statements in the source code. My printer inventory consists of the previously mentioned daisy wheel printer (which you made fun of) and a Hewlett Packard Laser Jet II which is used almost exclusively with "PageMaker" (a very powerful tool, but a subject which requires more room than I can devote here). If I get enough inquiries, I'll devote an article to the subject of using Aldus' Pagemaker with an HS-158 and the previously mentioned Laser Jet printer. You have heard reports to the contrary, but my experience with this combination has been more than satisfactory.

OK, I'll climb back down from my soap box and get back to the subject at hand. C__Power — Part 13 will round out our editor with not only a more sophisticated printer driver, but also a "tabbing" function, a carriage bell (read beep), a "page up", "page down" utility, and last, but not least, a "block move" routine. Also under consideration is an enhancement to the "insert character" function, enabling the storing of a line (past the screen borders) while utilizing the insert mode, as well as a "word wrap" feature. If enough "C__Power" readers are interested, perhaps

we could make "Edtxt" a "pop up" utility, possibly incorporating a calendar and calculator. Please let me know through a card or letter if you are in favor of such an enhancement. Maybe we can make our editor a truly useful utility.

I should have your attention by now, so let's take a look at the enhanced source code for "scrnlib.c". You will find two functions that report the cursor position, "curpos_row()" and "curpos_col()". They contribute towards the portability of their running mates, "print_string()" and "print_with_attr()" by furnishing the cursor coordinates that will be used in conjunction with the address of the first byte of screen memory to write to an individual address and its successors. Please note the use of "far pointers" to indicate the location of the screen memory being written to. The use of far pointers enables the access of memory residing in a segment beyond the usual 64K boundary otherwise imposed by the (small) memory model. On the other hand, we could invoke the large memory model when setting the compiler parameters, thus defaulting to 32-bit pointers which enable the addressing of memory not residing in the 64K segment. Being by nature parsimonious, I have opted for the small memory model and the use of far pointers. I do not intend for this article to become a dissertation on memory models or far pointers. Suffice it to say that since the memory in a PC is addressed by the CPU in chunks containing 64K of memory, by using a segment offset, plus the local address, any byte can be accessed.

Now look at the "define" statement which creates the address for the starting point of screen memory. The far pointer is implemented here and the hex address (0xB000L << 16) is shifted left, effectively multiplying the address and creating a LONG integer from an UNSIGNED integer. The end result is that we have a base from which we can directly write to any screen coordinate by incrementing the offset and adding that number to the base just created. I hope I haven't confused the issue here; look at the "prnt (string,row,col)" function in the scrnlib.c listing. Notice that the pointer is made equal to the screen address and then is incremented by eighty for each row after row (1) and increased by one for each column. The bottom line is that we wind up with a unique address for each screen character, resulting in a very high speed video write routine.

We have now arrived at the Rubicon, we must choose between a CGA board or a MONOCHROME installation. The reason being is that the actual address will be different for the two video modes and those readers who are using a CGA board for display will need to substitute a different hex address for the base of video memory. The magic number is: (0xB800L << 16). If you are using a color monitor, substitute this number for the SCREEN define statement. The compatibility between the two modes is non-existent at this point, but we will fix that problem in "C__Power — Part 13" by providing an automatic mode detection routine.

We did provide some portability for the printing functions by adding the "curpos (row)" and "curpos(col)" functions. By using these routines to determine cursor position, we have relieved the program of having to keep track of the current row and column.

Now that you have that under your belt, let's look at the listing for "Edtxt ver 1.12". We have added quite a few "#define" statements, most of them provided for the use of the screen printing routines. They are, for the most part, self-explanatory so I won't elaborate, except to say that these constants are used to provide the attribute information to the "print_with_attr()" function. By using these define statements, along with their companion routines, you can dress up your own programs to look quite professional. They are used in "Edtxt" to provide the name of our program at the top of the menu in BOLD face (brightness intensified) and at the bottom right corner of the screen to print the cursor coordinates in reverse video.

The next to last define statement might be considered redundant, but has been added to our program in an effort to facilitate debugging. Notice the "unsigned int print" addition to the struct declaration. This member of our structure is made equal to IDENT (from the define statement) within the "create()" function, thus giving us a means of checking the integrity of our linked list. If a check made on the value of the "print" structure member against the IDENT constant returns positive results each time a traversal of the linked list is made or a link is added during file creation, the totality of the list can be assured. The new "check(ptr)" function has been provided to take care of this job. Please note that a call to this routine has been added to "case 72", "case 80"

and "case 13" within the "get_ch()" function. It has also been added to the "ins_line()" function, as well as the "first()" and "rest()" functions. Several other changes were made to the code at the "case 13" location, enabling line insertion with a RETURN.

Before we peruse the rest of the listing, note the addition of "#include <string.h>" near the top of our source code. This should have been part of the last listing (Edtxt ver 1.11) but it slipped by me, sorry about that! By way of an explanation, the omission of the above include statement became obvious during some protracted debugging operations.

The next item of interest is the "update_screen()" function, a routine that provides the cursor coordinates in the form of line number and column information at the bottom right of the screen. The display is in reverse video as mentioned previously and gives the user some helpful information about the document he/she is working on. Both of the variables (ptr->lineno and col) are converted to strings using the "itoa()" library routine and then concatenated on to "line =" and "col =" strings, respectively, creating a "sign" that is displayed on the screen using the new "print_string()" function.

The next item of interest is "ins_line()". The only thing different here is the name itself. As luck would have it, I just got my copy of "Turbo C ver 1.5" in the mail and had to try it out. Well, I got a bunch of error messages indicating that there was a conflict with a library function named "insline()", hence, the name change. Actually, I also made some other rather small changes to the counter variables, substituting register types for variables of type integer where a time savings could benefit the program. The same technique has been used wherever I thought a similar change might benefit "Edtxt".

I do not intend to pass lightly over the mention of Borland's newest tool for "C" programmers without mentioning that I intend to give version 1.5 some more elaborate tests followed by a report, right here in "C_Power". In the meantime, I will continue to write the code for "Edtxt" to compile under the older version since the compilers are upward compatible.

Moving on down the source code listing to the "del_ch()" function, notice the current version has the comment delimiters removed to activate that portion

of the function which was disabled in version 1.11. The addition of the new code enables "del_ch()" to detect the end of a line (ptr->strng[i]==VACANT). When this condition is encountered, "Edtxt" assumes the user wants to append the line below to the existing line so a call to "join()" is made. Here, the current line is parsed in order to substitute spaces for any VACANT characters between the end of the occupied line and the cursor position. Next, the modified line is read into a larger storage area (strcpy(buff2,addon)), then the next line is appended (strcat(buff2,ptr->next->strng, 80)). Now we have the current line, plus the line below the cursor residing in a storage area large enough to hold two complete lines.

The next couple of operations get a little involved, but stay with me, they really aren't bad if you take the logic a step at a time.

We are faced with two problems, the first one being the inability to store the new appended line in our "ptr->strng" storage area, since that was declared at compile time to be large enough for eighty-two characters. The next problem is that the screen is only eighty characters wide, so we couldn't print a longer string to the screen even if we had a place to store it. How do we solve this dilemma?

First, we are going to measure the new entity in order to determine how much of the new line won't fit on the screen. We're still within the "join()" function and we will employ (len=strlen(buff2)) to get the overall length. Next, assuming a length in excess of 79 characters, we'll put the surplus characters in a global buffer that is not altered throughout the entire operation. By using an offset of 79 characters (buff2[j+78]), we can read the surplus into the global storage (buff) and then cap this part of the operation by appending a '\0' character to both storage areas. OK, we're moving along very nicely now, hang in there, it's going to get easier. We just cut our appended string off at the screen border by inserting the '\0' character at (buff2[78]='\0'). If we read the newly modified entity back into "addon" (strcpy(addon,buff2,80)), we can return the proper string to the calling function ("del_ch()") to be printed to the screen. The surplus characters are already stored in "buff" where we can utilize them by appending individually or en masse, depending on what the user has in mind. The "add_ch()" function will ration the characters out one at a time or they can

be inserted where desired by using the "ins_line()" function. How does "Edtxt" know which condition is to be employed? Notice that "flag_long" was set equal to one if the appended string was in excess of 79 characters. By using this information within the "del_ch()" function and the previously mentioned "ins_line()" routine, we can make the program behave in a quite intelligent manner.

We already mentioned "add_ch()", now I'll briefly explain the "what" and "how". A que is needed to parcel out the FIRST character in the buffer "buff". Each time it is accessed, the routine must shift the string contained within "buff" forward one position, after concatenating buff[0] to the string that was passed to it. Another job must be tended to, the flag_long must be reset when the last character is being appended. Examine the source code within "add_ch()" to see if you can figure out how this is accomplished.

I saved the easiest for last. "Drawbox()" does just that, a rectangular box is drawn on the CRT at the coordinates passed. The "#define" statements preceding our graphics function are used as an aid in determining where to put the parts of the puzzle. Our old friend, "locate(row,col)" is used extensively in combination with "for" loops to create a rectangle on the screen. There is nothing very sophisticated about this routine. In fact, it is in dire need of some speeding up. Why don't you write a faster function and send me the source code for inclusion in a succeeding article? Thanks, you're a real pal!

OK, I know, we aren't quite done yet. The printing module, Case 3 from the menu, is so easy I won't elaborate on it. This routine merely employs "read_file()" to create a linked list in memory from the file specified, and then just parcels the characters out to the printer using "putc". A much more elaborate printing module will be part of "C_Power — Part 13, along with a great deal more sophistication in other additions.

If you have trouble understanding the logic or methodology employed in "Edtxt", I have found two new books to be quite helpful. They are: "Turbo C Programming for the IBM" by Robert Lafore (Howard W. Sams & Company) and "C: The Complete Reference" by Herbert Schildt (Osborne McGraw-Hill). Although both are quite recent additions to my library, they are getting dog-eared already. A better endorsement would be rather verbose.

```

/* ***** SCRNLIB.C by John P. Lewis ***** */
/* ***** Created 8/1/87 ***** */

#define VIDEO 0x10
union REGS regs;

void locate(x,y) /* positions cursor at row, col */
int x,y;
{
    union REGS regs;
    x--;y--; /* enable first = 1 */
    regs.x.ax=(2 << 8)+00; /* set cursor position */
    regs.x.dx=(x << 8)+y;
    regs.h.bh=0; /* video page = 0 */
    int86(VIDEO, &regs, &regs);
}

void cls()
{
    union REGS regs;
    regs.h.ah=7;
    regs.x.cx=(00 << 8) + 00; /* top left of screen */
    regs.x.dx=(23 << 8) + 79; /* bottom right of screen */
    regs.x.bx=(7 << 8) + 00; /* 80 X 25 monochrome */
    regs.h.al=0; /* does not clear the 25th line */
    int86(VIDEO, &regs, &regs);
    locate(1,1);
}

void clr_dn(row1,row2)
int row1, row2;
{
    union REGS regs;
    row1--;row2--; /* enable usage first row = 1 */
    regs.h.ah=7;
    regs.x.cx=(row1 << 8) + 0;
    regs.x.dx=(row2 << 8) + 79;
    regs.x.bx=(7 << 8) + 00;
    regs.h.al=0;
    int86(VIDEO, &regs, &regs);
}

scroll_up(row)
int row;
{
    union REGS regs;
    regs.h.ah=6;
    regs.x.cx=(row-1 << 8) +0;
    regs.x.dx=(23 << 8) + 79;
    regs.x.bx=(7 << 8) + 0;
    regs.h.al=1;
    int86(VIDEO, &regs, &regs);
}

scroll_dn(row)
int row;
{
    union REGS regs;
    regs.h.ah=7;

```

"C" you soon.

Please let me remind you, before closing, that the code for writing to the video screen is specific to either CGA OR MONOCHROME. Be sure you use the appropriate line from scrnlib.c (use comments to remove the redundant code) for your video monitor.

If you have problems concerning "Edtxt", please send me a note describing your trouble along with an SASE. I will try to help.

If you lack the time and/or inclination to key in the listing for "Edtxt ver 1.12", you may obtain a copy of the source code, along with an executable file (Edtxt.exe) by sending a FORMATTED disk, either 5-1/4" or 3-1/2" to me at the address given at the beginning of this article. Please include \$5.00 to cover shipping and handling.

```

regs.x.cx=(row-1 << 8) + 0;
regs.x.dx=(23 << 8) + 79;
regs.x.bx=(7 << 8) + 0;
regs.h.al=1;
int86(VIDEO, &regs, &regs);
}

curpos_row(row)
register row;
{
    regs.h.ah=3;
    regs.h.bh=0;
    int86(VIDEO, &regs, &regs);
    row=regs.h.dh;
    row++;return row;
}

curpos_col(col)
register col;
{
    regs.h.ah=3;
    regs.h.bh=0;
    int86(VIDEO, &regs, &regs);
    col=regs.h.dl;
    col++;return col;
}

#include <conio.h>
/* ***** Video address, change to suit your needs ***** */
#define SCREEN ( (int far *)0xB000L << 16) /* use with MONOCHROME */
/* #define SCREEN ( (int far *)0xB800L << 16) */ /* use with CGA */
/* ***** */
#define ATTR 07

print_with_attr(p,row,col,attr)
char *p;
register col;
int row, attr;
{
    long offset;

```



```

char string[81], chr, filename[13], buff[160]; /* global variables */
FILE *fd, *f2;

typedef struct line
{
    char string[82]; /* structure declaration used with */
    struct line *next, *prev; /* linked list */
    unsigned int print;
    int lineno;
} list;

list *ptr, *runptr, *base, *nxtptr, *tmptr, *fptr;

create()
{
    register i;
    ptr = (list*)malloc(sizeof(list)); /* create storage area for above */
    ptr->print = IDENT;
    for(i=0; i <= 79; ++i) /* clean out the garbage */
        ptr->string[i]=VACANT;
}

check(ptr)
{
    struct line *ptr;
    if(ptr->print == IDENT)
        return 0;
    else {locate(1,25);printf("Link failure, save data ");
        chr='D';flag=1;return-1; }
}

go() /* see commented code in text, this code */
{
    /* is similar */
    union REGS regs;
    regs.h.ah=0;
    int86(0x16, &regs, &regs);
}

get_ch() /* see above */
{
    union scan {
        int c;
        unsigned char ch[2];
    } sc;skip=flag=0;
    sc.c=go();
    if(sc.ch[0]==0)
        if(chr=sc.ch[1];flag=1; }
    else chr=sc.ch[0];
    if(flag==1)
    {
        switch(chr) /* further decoding of input done using */
        {
            case 75:if(col > 1) col--:i=col-1; /* left arrow, move cursor */
                locate(row,col):in=0;break; /* left, decrement string subscript */
            case 77:if(col < 79) col++;i=col-1; /* right arrow, move cursor */
                locate(row,col):in=0;break; /* right, increment string subscript */
            case 72:fix_line():if(row > 1) {row--;} /* up arrow, move cursor up */
                ptr=ptr->prev;check(ptr);) /* get new pointer */
            if(ptr->prev != 0 && row==1) /* check for top of screen */
                {ptr=ptr->prev;scroll_dn(row);check(ptr);}
        }
    }
}

```

```

int far *fp,*p;fp=SCREEN;col--;offset=((80*row)-80)+col;
*(fp+offset)=(attr << 8)|*p;
}

print_string(attr,ptr)
int attr;
char *ptr;
{
    register i, col;
    int len, row;row=col=0;
    row=curpos_row(row);col=curpos_col(col);
    len=strlen(ptr);
    for(i=0; i <= len; ++i)
        { locate(row,col);print_with_attr(ptr,row,col,attr);col++;ptr++;}
}

print(string,row,col)
char *string;
int row, col;
{
    char ch;
    register i;
    long offset=0;
    int far *fp;fp=SCREEN;col--;
    offset=((80*row)-80)+col;i=col;
    while(*string && i < 80)
        { ch=(*string++);*(fp+offset)=(ATTR << 8)|ch;offset++;i++;}
    for( ; i <=79; ++i)
        {*(fp+offset)=(ATTR << 8)|' ';offset++;}
}

/* ***** Edtxt by John P. Lewis ***** */
/* ***** ver 1.12 last update 2/16/88 ***** */

#include <string.h> /* header file used with strxxx functions */
#include <stdlib.h> /* header file used with malloc functions */
#include <dos.h> /* header file used with int86 interupts */
#include <stdio.h> /* standard header file (I/O) */
#include "scrnlb.c" /* our own screen handling file */
#include <\sys\stat.h> /* used with file I/O */
#include <fcntl.h> /* as above */
#define VIDEO 0x10
#define TRUE 1
#define FALSE 0
#define VACANT 0
#define SPACE 32
#define NORMAL 7 /* definitions used with print_string library */
#define BOLD 15 /* function to pass attributes */
#define BLINK 128
#define RVID 112 /* reverse video */
#define BRVID 240 /* blinking reverse video */
#define UNDERLN 1
#define IDENT 0x1234 /* provides an identity for each link */
#define MAXLEN 80

int flag, init, row, col, i, end, in, skip, lines, flag_long, flag_in=1;

```

```

locate(1,1);prnt(ptr->string,row,1); /* print updated screen */
locate(row,col);in=0;i=col-1;break;
case 80: fix_line(); /* down arrow */
if((row < 24) && ((ptr->lineno) < lines)) /* test for file & */
{row++;ptr->ptr->next;check(ptr); /*screen position, get new pointer */
locate(row,col);in=0;} else
if(row==24 && ptr->lineno < lines)
{ptr->ptr->next;scroll_up(1);locate(24,1); /* cause a screen */
check(ptr);
case 67:if(row==24) {scroll_up(1);row=23;}
row+=1;i=col-1;ins_line(ptr->string,i); /* insert line */
refresh_screen();break;
case 66:del_line();break; /* delete line */
case 83:fix_line();i=col-1;in=0;
del_ch(ptr->string,col-1);break; /* delete character */
case 82:in=1;ins(i);ins_ch(ptr->string,col-1);break; /* insert char */
default;break;
}
}
else
witch(chr) /* further decoding of input */
{
case 13:if(in=1)
{ row++;i=col-1;ins_line(ptr->string,i);
if(row > 23) { scroll_up(1);row--;clr_dn(25,25);}
refresh_screen();in=0;break;}
col=1;i=col-1;if(row==24) scroll_up(1);
if(row < 24) row++;
if(ptr->next==0) /* test for ptr pos in file */
{i=80;update_screen(); /* force loop exit if last line */
locate(row,col);}
else
{ chr=VACANT; /* if not last line */
ptr->ptr->next;check(ptr);if(row==24)/* get new ptr, skip linker */
refresh_screen();} break;
case 3:exit(0);break; /* control/c to exit */
case 8:putchar(8);putchar(32); /* backspace char */
putchar(8);if(col > 1) col--;in=0; /* move cursor back, print */
ptr->string[col-1]="; /* space, move cursor back, decrement */
i=col-1;break; /* string subscript */
default:if(chr >31 && chr < 127 && col < 80)
{i=col-1; /* must be alpha/numeric character */
ptr->string[i]=chr; /* add character to string */
putchar(chr); /* print it to scrn */
col++;break;}else
{ ptr->string[i]=VACANT;i--;}break;
}
if(flag_in i=in) ins(); /* update overstrike/insert flag */
update_screen(); /* print col & row info (to screen) */
if(flag==1 && chr != 68) chr="; /* null special keys */
return chr;
}
}
update_screen()
{
char linum[4], colnum[3], display[24]; /* prints row & col info */
itoa(ptr->lineno,linum,10); /* at bottom of screen in */

```

```

itoa(col,colnum,10);strcpy(display," Line ="); /* reverse video */
strncat(display,linum,3);strncat(display," Col = ",7); /* thanks to */
strncat(display,colnum,2); /* print_screen function */
locate(25,59);print_string(RVID,display); /* found in updated library */
locate(row,col);
}
first()
{
base=ptr;ptr->prev=0; /* first line, initialize next */
ptr->next=0;runptr=ptr;
init=TRUE;lines=1; /* & prev to 0, create ptr to 1st line */
ptr->lineno=lines;update_screen();
check(ptr);
}
rest()
{
runptr->next=ptr;ptr->prev=runptr; /* link previous line (runptr) */
ptr->next=0;runptr=ptr; /* to this one (ptr), link prev ptr to */
skip=FALSE;lines++; /* previous line - see text */
ptr->lineno=lines;check(ptr);
}
ins_line(stng,pos) /* inserts a new line at */
char stng[]; /* cursor position */
int pos;
{
register i, j;
int temp, lyne;
char string1[80], string2[80], dummy[2];fix_line();
clean_it(string1,80);clean_it(string2,1,80);
strcpy(string1,stng);j=pos; /* copy current line to storage */
for( j <=79; ++j) stng[j]=VACANT; /* clean out garbage from cursor */
locate(row-1,col);lyne=ptr->lineno; /* position to end, store line no */
for(j=pos; j <=79; ++j) putchar(32);j=pos; /* erase (screen) from cursor */
for(i=0; i <=79-pos; ++i, ++j) /* store balance of line in */
string2[i]=string1[j];locate(row,1); /* string2 */
if(flag_long==1) {string2[i]='\0';
strcat(string2,buf);flag_long=0;}
print(string2,row,1); /* move to new line & print it */
if(ptr->next!=0)
{nextptr=ptr->next;tmptpr=ptr;create(); /* establish link with new line */
tmptpr->next=ptr;ptr->prev=tmptpr;
ptr->next=nextptr;nextptr->prev=ptr;} /* see text */
else
{ tmptpr=ptr;create();tmptpr->next=ptr;
ptr->prev=tmptpr;ptr->next=0;}
strcpy(ptr->string,string2);
ptr->lineno=lyne+1;runptr=ptr; /* since we have inserted a */
check(ptr);
update_lineno(lyne+2);lines++;col=1; /* new line, line numbers must be */
/* updated from this pos to end */
}
refresh_screen() /* prints updated screen from */
{
int temp; /* current row to line 24 */
tmptpr=ptr;temp=row;clr_dn(row,24); /* store pointer and row info, */
}

```


On The Leading Edge

William M. Adney

*P.O. Box 531655
Grand Prairie, TX 75053-1655*

Disk Backups, Microsoft C, Subdirectory Names, LABEL Names, Expanded and Extended Memory

Every once in a while, I manage to find a topic that seems to generate an unusually large volume of mail. Before this year, the last such article was the one about the problems I had installing the Easy PC emulator hardware in my Z-100. That article appeared in the June 1986 issue of REMark. At that time, I received 14 letters from Huggies that had similar or worse problems with the Easy PC, and three letters indicating no problems whatever. Other letters were also written to and published in REMark (Buggin' HUG) supporting both the "for" and "against" considerations for the Easy PC.

Although all of the returns aren't in yet, it appears that my comments about Microsoft C in the February issue have generated a similar amount of interest. Interestingly enough, the mail that I have received so far indicates that six Huggies have had NO problems with either Microsoft C release, and one Huggie did have a hard disk wipeout with QuickC. Let's see what the situation is by looking at backups first.

The Importance Of Backups

In order to place the problem in perspective, I have decided to reverse the

order of the original discussion to help make the point. And the real point of this is to encourage you to make backups — especially if you use a hard disk — before you make any major changes to your system whether it is new software (including updates) or hardware changes. As I have mentioned before, I always make backups of my hard disk partitions before I make changes in the hardware, and I usually make complete backups before new software installation or updates as well. From my own experience, I know that it can still take hours to recover from a hard disk wipeout even though you have complete backups of everything.

In both instances, everything on my entire 26 megabyte Z-100 hard disk was lost. In the first instance, the SuperBlock on the hard disk was apparently clobbered due to a two-second power "blip" which required a complete hard disk restart and reload beginning with the PREP command. Fortunately, I only lost about 20 minutes of work since I had just copied the file (an article) from the hard disk to a floppy as is my normal practice. In the second, an old ROM on the Easy PC emulator was not compatible with anything larger than a 10 megabyte hard disk, and the SuperBlock was clobbered again.

Since I have worked with mainframe computers for 20 years or so, I learned that it is important to make regular disk backups, and this experience has proved valuable in microcomputers as well. At least it has saved me a lot of time, not to mention the fact that I have not lost much data or time due to frequent backups.

Even if you do not have a hard disk, don't underestimate the importance of making backups. You can also lose data from a floppy disk due to a mechanical problem, media problem or mistake. In my mind, the most frequent problem here is the fact that some DOS versions (ESPECIALLY Z-DOS version 1 and some PC-DOS versions) would very cleverly FORMAT the default drive if you did not enter a drive letter in the command line. I managed to FORMAT more than one system disk in drive A with Z-DOS, and I had to recover a hard disk at UTA that someone formatted with PC-DOS version 3.10 because of the same problem. That's why I always recommend you enter a drive letter with the FORMAT command.

And, of course, there is always the DEL *.* command that can erase all files from any directory or an entire disk. Many people recommend (and I do too) that you enter

a DIR *.* with the appropriate asterisk and/or question mark wildcards first. Check the directory display to be sure that you really want to delete those files; then you can type in DEL, press the F3 Function Key to copy the rest of the command line, and press RETURN to delete the files. In order to recover from these mistakes that we all make from time to time, it's a good idea to backup important files even on a floppy disk system.

Hardware, and ROM firmware, are one potential source of problems. Mistakes, typos, and other kinds of errors that all users make are another. And software of all kinds is another. Although you can generally cope with the first two kinds of problems with a little knowledge and care, it is not quite as easy to identify subtle, and not-so-subtle, problems with software.

Checking Out Software

As far as I'm concerned, the name of the manufacturer on the software label doesn't make much difference whether it's Zenith, IBM, MicroPro or Microsoft with respect to how much I trust a new release or an update. In most cases, it also depends on the type of software (e.g., DOS, word processing, spreadsheet, etc.) and the experience that I have had with it. Since I already picked on Microsoft in the February issue, and various versions of IBM's PC-DOS have had a number of well-documented problems in various publications, I'll pick on Zenith this time.

In my mind, the worst piece of application software ever to be released from Zenith was WordStar for the Z-100, but it was not MicroPro's fault. If you ever used this special WordStar version for the Z-100, you know that it is SLLOOOWWWW. Part of the problem is that disk I/O is incredibly slow, and Pat Swayne even wrote a patch to fix that problem which was published several years ago. But that's not the end of the problem. In the one Zenith version I have, MailMerge just does not work. When I called Software Consultation several years ago, I was told that it was a known bug, and I could get an update (to FIX the problem) for \$35 or so — I forget the exact cost. I couldn't believe that I would have to pay for a fix to a known bug, so I returned to using the CP/M version of WordStar on my Z-100 that did work correctly.

In short, the Zenith WordStar implementations were so bad that I have not

used them because I don't trust them. I know of enough problems, such as these, that I can't help but wonder if there aren't other problems just waiting to jump out. As you probably know, Zenith has gotten out of the application software business for the most part, and perhaps this kind of problem explains why.

On the other hand, I think that Zenith does an absolutely superb job on operating systems software, such as MS-DOS and OS/2. I have used a lot of DOS versions from other manufacturers, and the Zenith MS-DOS is clearly the best in my experience. That doesn't mean that it's perfect by any stretch of the imagination, but I have found it to be the best and most reliable.

Even so, I still carefully check out new releases of Zenith MS-DOS before I trust them in my production system for writing. Fortunately, I have two hard disks in my '248, and I typically use one for testing new operating system software before I install it on my production hard disk. Yes, that may be excessive caution, but I don't lose much, if any, data when something unexpected happens. Still, I can't afford the time to recover from "killer" situations that can cause extensive data loss because of deadlines for articles and books.

Since all computer systems are vulnerable to hardware and software bugs and failures, as well as user error, I always recommend that everything be checked out as thoroughly as possible before using anything in a production system.

In the January article, I mentioned the trivial example that Zenith MS-DOS version 3.20 now uses the /V switch to prompt for a volume label instead of verifying the FORMAT. All version 3 releases of IBM's PC-DOS have used the /V to prompt for the volume label, and apparently Zenith changed this in version 3.21 to make it the same as PC-DOS. Zenith's version 3.20 (page 3.73) notes that the /V switch is included for compatibility with previous versions, although FORMAT verification is automatic and the switch performs no function.

This example points out the necessity of carefully reading the manual, since this is a detail that you might otherwise overlook. In general, this kind of thing might be a nuisance, but there are cases when changes to DOS programs could lead to strange results and possibly even loss of

data under certain circumstances. There is an easy solution.

In my opinion, the ZDS documentation should include a list of new commands and enhancements like previous manuals did up through 3.10. This section should document major changes or enhancements to existing programs (such as the changed meaning or syntax of a switch), as well as the new commands/features that are part of that version as compared to the previous release. I have already suggested to ZDS that this be added to the documentation to make it easy for users to identify where major changes were from the previous DOS version. Although the Zenith documentation is generally very good, I think that it would be better if this information was again included in the Introduction. We'll see if that happens.

One could argue that computer users should not have to worry about software and hardware bugs causing problems, but that view is not very realistic. Even automobile manufacturers have bugs in new cars. Sometimes the bug is safety-related, and they are forced to issue a recall on a car model. In most cases, computers present a significantly different problem than automobiles. The computer system hardware and operating system software must be able to cope with an incredibly wide variety of third-party hardware configurations and application software. Automobile manufacturer's don't, and they still have problems.

For example, let's say you wanted to install a Corvette 350 cubic-inch fuel-injected engine in a 1972 Pinto, and suppose you called Ford for some information on how to do so. After recovering from the initial shock (and perhaps some laughter) of the request, you would probably be told, in so many words, that the engine and the car were not compatible. There is a similar analogy in today's microcomputer market. You can buy "turbo" boards that fit into an IBM PC and effectively change it to a Pinto with a 350 engine.

Although most people would recognize that Ford is not reasonably obligated to help you in this situation, most of those same people (including me) would say that microcomputer manufacturers and software manufacturers have a much broader obligation to purchasers of its product. But problems occur in all kinds of products, and it is best to plan for them. Such was the case with the prob-

lems that I mentioned in the Microsoft C versions.

Microsoft C — Revisited

In the interests of identifying problems in a timely manner, I am frequently torn between the decision of whether to write about something on the basis of incomplete information or not. In general, I usually opt for waiting for complete information so that I can write about both sides of the story. In all cases, I research the information by checking with a vendor if appropriate or by personal testing on my own systems depending on the situation. Such was the case with the problems I reported with Microsoft C as I mentioned in the February article.

At the time I checked with Microsoft on these problems, they did know there were problems, but they were unable to pinpoint the causes. Although I attempted to approximately identify the date my comments were written by noting a similar report in the November 30, 1987 InfoWorld, several people seem to have missed this.

Given the popularity of Microsoft C, my concern was (and still is) that some of the "old" new C and QuickC Compilers are still in inventory waiting to be sold. I believe it is appropriate to warn users about potential software or hardware problems and/or limitations as the information becomes available even if it is incomplete. At this point, I have enough of the details of the problems that may be helpful to you if you are using a current version of either compiler.

The QuickC problem was the most serious because it scrambled the File Allocation Table on a disk rendering that disk useless. Because it did not ALWAYS occur on every system, it was the most difficult to identify, and its cause is one of those obscure things that you would never think of. It does occur in Western Digital hard disk controllers as I mentioned, but it only occurs if you have EPROM #62-000043-010.

The problem occurs because there is an interrupt conflict (Interrupt 7, I believe) between that ROM and QuickC. Since QuickC only uses that interrupt when necessary, you can still have that specific ROM and find no problems until the conflict occurs and scrambles the disk's FATs. Although nobody seems to want to admit anything about the problem, you can

write to Western Digital for information about a new ROM if you have that specific version. Microsoft says they will change QuickC to eliminate the conflict by the end of 1988.

The reported problems with the version 5.01 are varied. The major ones are not particularly obvious and won't affect you unless you have the Windows Development Kit. It seems that the old version 1 Windows libraries are not compatible with this new compiler. That causes all kinds of strange problems from a program compile failure to all kinds of wondrous happenings when the program is run including weird things occurring with AUTOEXEC.BAT if the program is loaded by that batch file. Results appear to be unpredictable, and there are many reports as to what a Windows compatible program did when these old libraries were used. Microsoft simply says that the old Windows libraries cannot be used with the version 5 compiler. But that doesn't seem to be the end of the problem.

Install Programs

A friend of mine has the Microsoft version 5 C Compiler, and he reports that his AUTOEXEC.BAT and CONFIG.SYS file "disappeared" after he ran the INSTALL program for the compiler on his hard disk. Although that has been some time ago, my recollection is that Microsoft told him to return the original distribution disks for replacement. There was no statement as to the problem, but it seems strange that those two files disappeared after INSTALL was run. Both files were later recovered by an "unerase" utility. But there is a moral to this story too.

I must admit that I personally feel uncomfortable when there seems to be a need to run an install program for any software, and it doesn't matter who the manufacturer is. For example, I recently installed WordStar 2000 Plus version 3 on my hard disk, and there is no reasonable way to install all 21 distribution disks (that's not a typo — it really was 21) without using the install program. I felt the same way when I installed the Ecosoft C Compiler version 4 because the installation was changed from a batch file to a program. And the Zenith OS/2 installation is accomplished with an install program too, and that is the only one that I installed right the first time.

I can't help but wonder what all of these install programs are adding to my hard

disk, and if there is one small glitch in the program or if I make a mistake, I usually have to rerun the entire install procedure. Besides, I also wonder if there is anyone who has installed one of these complex programs for the very first time who hasn't had to reinstall because it was incorrectly installed. Most manufacturers seem to think that their install programs are the solution to all problems, but when something unique (usually called an error of one kind or another) occurs, there is no way to figure out what went wrong.

Even worse, the reliance on these install programs is so heavy that there may not be any documentation as to what files go where. MicroPro did a nice job on explaining which files do what, although it is included as a document file on the disk. I don't object to that since I know that file names do change, but at least it is there. In my opinion, all vendors who supply install programs with their software should include some documentation on the files on the disks. If the install program creates subdirectories, those should also be explained in a document file. For example, the WordStar 2000 install created about a half dozen subdirectories that I didn't know about until the program was finished. I don't feel nearly as comfortable with many other programs simply because I don't know exactly where all of the files are or, in some cases, what they do.

As software becomes more sophisticated, I suppose all vendors will have to go to install programs instead of batch files, but I think it is critical in that case to backup a hard disk BEFORE installing new software. If there is a glitch in the install or you make a mistake, you can always reload from the backup and start over.

Microsoft C — Just Say When

As I mentioned in the introduction to this column, I received a number of letters about that item on Microsoft C. In a rather interesting exercise in logic, one of my friends told me that he has both compilers, and since he had not had any problems, the problems did not exist. Hmmm . . . no doubt Microsoft will be interested to hear that.

At this point, the most interesting letter that I received was from Jim Reeb (Parker, C). Jim is also chairman of a C interest group in the Denver area, and he reports that nobody in his group has found any problems with either compiler. He also

points out that he is concerned about inaccurate problem reports, and so am I. That's why I spend considerable amounts of time verifying something before you see it.

It is unfortunate that these kinds of things occur, but I believe that there are enough people who use a C compiler that the information is worthwhile. I chose the Microsoft C Compiler examples because they were current, and they also demonstrate that "killer" problems can be caused from totally unexpected software bugs. I think most programmers would agree that one would not reasonably expect any compiler to scramble a disk's File Allocation Table under ANY circumstances — a program with a mistake, sure . . . but NOT a compiler.

All of these explanations seemed necessary to explain more about the problems, but the moral of the story is still the same — Back up your disk(s) before installing any new software. Now on to bigger and better things.

The 640 K Barrier

John Membrino (Berwyn, PA) has a number of interesting questions that center on the old 640 K barrier that is sometimes mistakenly called the DOS limit. In particular, he wants to know how the 80386 systems and OS/2 break that 640 K limit, so let's go back in history for a minute.

The original 640 kilobyte limit was part of a design limitation in the IBM PC. The first limitation was a specific hardware limitation in the 8088 CPU. It could not address more than one megabyte (1024 kilobytes) of memory which includes all conventional system memory (i.e., RAM), video memory, ROM, and some other things. Part of the reason for this restriction is that the 8088 CPU uses an internal 16-bit architecture, although it communicates with an 8-bit bus. Without going into a detailed memory map of a PC, suffice it to say that IBM decided to reserve the top 384 K of memory for these certain functions. The remainder (640 K) was available for conventional system memory which is usually called RAM for Random Access Memory. That's the reason for the 640 K limit in the original PC systems. As a side note, the Z-100 can use up to 768 K of RAM because its hardware design is different from the PC.

The 80286 CPU has an internal 32-bit architecture, and it communicates through

a 16-bit bus. The 80386 CPU has an internal 32-bit architecture, but it talks to a 32-bit bus. Both CPUs can address significantly more memory than the old 8088 because they can use "bigger" addresses. For example, the 80386 CPU can talk to four gigabytes (4,096 megabytes) of memory which is probably a lot more than we'll ever need, at least for the next year or two. By way of comparison, many large mainframes use some multiple of 16 megabytes for main memory (i.e., RAM), and it is not unusual to find 64 MB of memory on a mainframe. Since these smaller CPUs for microcomputers can address that much memory too, it is not likely that there will be a problem as far as the hardware is concerned. But there is more to it than that.

Whatever operating system is used must, of course, be capable of addressing the memory in the system. As I recall, OS/2 can address up to 16 megabytes of memory which is strictly an artificial limit as far as the software is concerned. Adding that much memory to any system would be tricky at best, and impossible in some cases, because there may not be enough expansion slots for the required memory boards. To get back to John's question about the 640 K barrier, I will summarize what all this means.

First, it is unlikely that any PC compatible DOS version will break the 640 K barrier on any system since it also has to run on 8088-based systems. The 80286 and 80386 systems essentially break that barrier because they can address much more than one megabyte of memory. And finally, the operating system, such as OS/2, must be written to take advantage of the increased memory addressing capability of the newer CPUs. In order to take advantage of most of these features, you must have at least an 80286 CPU, plus the appropriate operating system and application that can use it. And, to add to the confusion, there are different kinds of memory that John also asked about.

Extended And Expanded Memory

It's difficult to remember which kind of memory does what, but the essential key is whether your system uses an 8088 CPU or an 80286/386 CPU. I will ignore a discussion of the 8086 and the 80186 CPUs since they are not used in our systems. Since extended memory is the easiest to explain, I will start with that.

EXTENDED MEMORY begins at one megabyte period. As a result, it cannot be added to an 8088-based system, such as the Z-150 series (or IBM PC) computers because of the one megabyte limitation that I mentioned before. Therefore, it can only be added to a Z-200 or Z-386 system, and you can buy special memory boards for that purpose. Unfortunately, there is more to it.

In order to use extended memory, you must also have special software that can use it. If you are using MS-DOS, you are pretty much limited to some of the device drivers, such as VDISK, which are specially programmed to be able to use extended memory. For OS/2, you must have at LEAST 1.5 megabytes of memory that includes the usual 640 K plus an extended memory board that will bring your total memory up to that. While extended memory is not especially useful in the DOS world unless you need a large RAM disk, it is essential if you intend to run OS/2 on your system because OS/2 cannot inherently use EXPANDED memory for its system software. How expanded memory is used is a somewhat different story.

EXPANDED MEMORY is used to "expand" conventional system memory beyond the normal 640 K barrier, and it can be used on nearly any PC- or AT-compatible system. This type of memory is frequently called EMS for Expanded Memory Specification that was developed jointly by Lotus, Intel, and Microsoft (LIM). For that reason, it is also referred to as the LIM/EMS. There is also an Enhanced Expanded Memory Specification (EEMS) that does not seem to have achieved much popularity. But the installation of expanded memory is more than just a memory board.

Virtually all expanded memory boards include a special device driver that must be installed in the CONFIG.SYS file in order to use expanded memory. A friend of mine bought an expanded memory board for his '151, and he called me to help him figure out why it wouldn't work with the Zenith VDISK device driver. In most cases, the order of the commands in the CONFIG.SYS file does not matter, but when expanded memory is used, the order is absolutely critical. You MUST install the EMS device driver BEFORE you attempt to install any other device driver that uses it. For example, the CONFIG.SYS file would contain some lines like:

```
DEVICE=EMS.DRV (Right way)
DEVICE=VDISK.SYS /A
```


The EMS.DRV is a made-up example of the device driver name for expanded memory, and I haven't included all of the parameters for the VDISK device driver, except for the /A switch, which tells Zenith MS-DOS 3.21 to use expanded memory. A /E switch is used for extended memory. But if you have the lines in the reverse order, such as:

```
DEVICE=VDISK.SYS /A (Wrong way)
DEVICE=EMS.DRV
```

You have attempted to load VDISK in expanded memory before initializing the memory with the EMS driver, and it simply won't work. Most of the documentation I have seen for expanded memory boards points out that the EMS driver must be installed before any other driver that uses the memory, but sometimes it is hidden. My friend simply added the EMS driver as the last line in the CONFIG.SYS file, and he changed the VDISK driver to use the /A switch. In general, you need to be somewhat careful about the order of the DEVICE= commands for device drivers, and the EMS driver is an example of how critical that can be. If you have added a new device driver and something doesn't work, try changing the order of the DEVICE= commands. In the worst case, you may have also discovered an incompatibility between device drivers you use. Remember that device drivers are essentially memory-resident programs, and you can have conflicts among them just like the ones you load on a command line. Now, back to extended and expanded memory.

To summarize, extended memory begins at one megabyte and can only be used on Z-200 and newer systems. Extended memory is not particularly useful in a DOS system, since there are few programs that can use it, but it is required for OS/2. On the other hand, expanded memory can be added to nearly any PC- or AT-compatible system, but it requires that a special device driver be installed to use it.

For both types of memory, you must have software that can actually use that specific type of memory. Expanded memory is perhaps the most limited in that regard because an application program must be specifically written to use it. For example, Lotus 1-2-3 is one of the few programs I know of that can use expanded memory. My suggestion is to not buy any type of memory unless you have a specific use for it, and you know what kind of memory you need. You may have spent hundreds

of dollars only to find that you can't use it in the way you thought.

Subdirectory Names

As I mentioned in the February issue, I received a postcard from a Huggie named Bob who asked me why DOS subdirectories are limited to eight characters. Although I try to mention an individual's first and last name in the article when I discuss an answer to a specific question in my column, the postcard was not signed with a last name nor did it have a return address. The cancellation stamp was blurred, but it looked like the state began with an "M", such as MI, MD, MA or MN, but I couldn't tell for sure. In any case, I received letters from several Bobs, including Bob Hudson (San Diego, CA) indicating that a subdirectory name CANNOT be more than eight characters.

Although he (Bob Hudson) did not specifically mention what DOS version he was using, there is a high probability that we are both right. For example, I have a release of PC-DOS version 3.00 that has all kinds of problems with subdirectory names. If you enter a valid command line like "B:XDOSXFORMAT", that PC-DOS version does not even attempt to find the FORMAT program in the XDOS subdirectory on drive B. Instead, it simply changes the default drive to B and ignores the rest of the command line. Although I don't know whether this is an IBM or a Microsoft bug in the DOS, it may possibly explain why Zenith did not release a version 2.00 or 3.00 of MS-DOS. Regardless of the situation, let's take a look at the general philosophy of file names in the disk directory.

All file names in the general form of "filename.typ" are stored in the disk directory subject to the various DOS restrictions. The first 11 bytes of each 32-byte directory entry are reserved for the 11 characters in the file name — that is the maximum of eight characters for the filename and three letters for the file type or extension. The period (.) between the filename and the file type is not stored in the directory.

As it turns out, byte number 12 is called the "Attribute Byte". The Attribute Byte is bit-mapped which simply means that a 00 hex is equivalent to a 00000000 in binary, and each bit position has some special significance to DOS. In a bit-mapped byte, such as this, each bit is numbered from left (called the high-order

bit) to right from bit 7 to bit zero. The right-most bit, called bit 0, can be changed with the ATTRIB command because that bit is the Read Attribute. If the value of bit 0 is zero (the usual case), the file can be written to as well as read, and it is usually called a Read/Write file. If the value of bit 0 is one, the file is a Read-Only file, and you can read it, but not write it. You can change the Read Attribute of bit 0 with the ATTRIB command.

When you create the first subdirectory below the root directory with the MKDIR (or MD) command, DOS updates the disk (i.e., root) directory with the file name in the form of filename.typ as usual. Because it is a subdirectory, the Attribute Byte is set to 00010000 in binary or 20 in hex, where the 1 in the specific position shown means that the directory attribute is turned on. Then, DOS knows that the file name is really a subdirectory name, and it is used, for example, by the DIR command to display a <DIR> indicator for each subdirectory.

From a DOS perspective then, a subdirectory is nothing more than a file name that has a special Attribute Byte indicating that it is a subdirectory. And since a subdirectory name is really a file name in the general form of filename.typ, it is also subject to the usual DOS file name restrictions as to what characters are valid and so on. Although you may have a specific DOS version that does not allow you to have a subdirectory name that is larger than eight characters, there is no technical evidence that Microsoft ever intended to have this restriction.

You can check out your specific DOS version by entering the command: MD FILENAME.TYP, and you should find that DOS creates a subdirectory called "FILENAME.TYP". In some cases, I have found that users have attempted to create a subdirectory called "FILENAMETYP" which does not include the period between the filename and the file type — that is not a valid DOS file name, and DOS will not accept it as a valid subdirectory name either.

Although this discussion has dealt with the maximum length of a subdirectory name, it's best to keep these names as short as possible consistent with a name that is meaningful to you. For example, I use XDOS for the subdirectory that contains all of the MS-DOS programs, XWS4 for WordStar version 4, XHUG for HUG software (e.g., HADES), XWORD for

Microsoft Word, and XMAce for the Mace Utilities. By keeping these names short and easy to remember, I've found that I rarely have difficulty finding a file on my hard disk, and short subdirectory names minimize the number of key-strokes required to change directories too. As you may have noticed, I have in some cases used a subdirectory name that is exactly the same as the program name to make files easier to find. You might find that useful too.

Volume Labels

If you have Zenith MS-DOS version 3.10 or later, you have probably noticed the LABEL command that is used to add a volume label, sometimes called a volume ID, to a disk. Technically speaking, the LABEL command simply updates the disk directory with a "file name" that may consist of up to 11 characters and includes an Attribute Byte of 00001000. Interestingly enough, the some versions of the LABEL command seem to accept all kinds of strange combinations of ASCII characters (including spaces) that are not normally accepted as a complete DOS file name. For example, you can use FILE-NAMETYP as a perfectly valid label since

the characters are stored in the usual 11 bytes reserved in the disk directory for the file name. This command is the only exception I know of to the usual DOS rules on file names, since it performs special processing to create a volume label.

Powering Down

In the last few months, I've had a number of letters about various kinds of utilities that you can use to make you and your system more efficient. As a result, I've tried to obtain a variety of utility software that is new and helpful, and I think you will find some interesting products discussed here in the next few months. For example, I have 13 interesting utilities that are part of the Baker's Dozen that was thoughtfully suggested by Andy Brunskill (Seattle, WA) that we will take a look at next month.

But I have not limited this to only software. I have some interesting new hardware that I have been using with my '248, and it really is great. I think you'll like it, too.

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

Software

MS-DOS Version 3.2 PC only (OS-64-61)	\$149.00
--	----------

Hardware

Advanced Personal Computer (HS-248-T)	\$1599.95
Advanced Personal Computer (HS-248-TX)	1999.95
H-386 Desktop Computer (HS-386-A)	3349.95

Heath/Zenith Computer Centers
Heath Company Parts Department
Hilltop Road
St. Joseph, MI 49085
(800) 253-7057
(Heath Catalog orders only)



If You Don't Have WindowDOS 2.0, You're Wasting Time!!

Once you've experienced *the convenience of instant access to DOS commands*, you'll never be satisfied with returning to DOS to list files, format diskettes, or copy, rename, or erase files. Nor will you be happy with a DOS shell, because shell programs are just as inaccessible as DOS when you are using an application program. *Only one program combines memory-residency with the power of a full-featured disk manager: WindowDOS Version 2.0.*

Features Not Found In DOS

- ◆ Sort directories in 8 ways
- ◆ Copy, erase, or move groups of files
- ◆ Find any file in seconds
- ◆ Display default directory of any drive with a single keystroke
- ◆ Global copy & erase commands
- ◆ Copy function prompts you to insert another disk if necessary
- ◆ Display hidden files/subdirectories
- ◆ Display file contents in various formats and page forward/backward

- ◆ Display graphic tree
- ◆ Unique RAM Environment function shows name, size, location, and interrupts of every program in memory
- ◆ Rename subdirectories for instant reorganization
- ◆ Hide & unhide subdirectories
- ◆ See & change file attributes
- ◆ Send control codes to printer
- ◆ Switch default printer
- ◆ Password "lock" your system
- ◆ 5-minute screen-blanking function

Enhances These DOS Functions

- ◆ Format disks (faster than DOS)
- ◆ Make or erase subdirectories
- ◆ Copy, rename, or erase files
- ◆ Copy files to printer or COM ports
- ◆ Display disk free space

Other Information

- ◆ Not copy protected
- ◆ Uses only 51K of memory
- ◆ Supports EGA & Hercules
- ◆ Uninstall command
- ◆ For PC/XT/AT/100% Compatibles
- ◆ **Order Today--Only \$49.95**

WindowDOS Associates • Box 300488-B • Arlington, Tx 76010 • 817-467-4103

EGA, Desktop Publishing And Me



Eric L. Pang

1530 Nehoa Street
Honolulu, HI 96822-2008

This article will serve two functions. First, it will review ATI Technologies' EGA Wonder board. Next, I'll give you my impressions of Media Cybernetics' HALO DPE (Desktop Publishing Editor). Finally (did I say two?), I will go into the reasons I chose this hardware and software.

Introduction

I have a Zenith 158 and a ZVM-135 color monitor. Not being particularly wealthy, actually, not being wealthy at all, but still wanting to get high resolution graphics (I hear you, Zenith, Tandy and TI offered "higher" resolution graphics even before that "other" vendor introduced their microcomputer) and wanting to do some hacking with desktop publishing, I decided to get an EGA graphics card and a desktop publishing program.

I thought I would first get a multi-mode video card that emulates the EGA, CGA and Hercules video modes. A few of these cards will also allow you to display a few EGA-mode programs on a regular RGB monitor. Initially, I could run Microsoft Word and Windows in color, then eventually, upgrade to an EGA monitor. Since part of my job is involved with writing user documentation and performing user training, I also wanted to get a desktop publishing program. With this software, I would be able to include illus-

trations with the text (without having to physically cut and paste them together onto a piece of paper) and create overhead transparency slides (for presentations) with both words and pictures.

The ATI EGA Wonder Video Card

After reviewing the literature for many EGA-compatible cards and having used the Video-7 VEGA Deluxe at work, I decided to get ATI Technologies' EGA Wonder video card. It is a half-height card that will run software developed for the EGA, CGA, MDA and Hercules video modes on any of the major monitor types; an Enhanced Graphics Display, multi-sync monitor, RGBi monitor, TTL monochrome monitor or composite display. It is hardware compatible with the four video modes and has an automatic mode switching feature to change between the different color and monochrome software standards without having to change monitors or resetting switches. The EGA Wonder is also compatible with VGA mode 11 (640 × 480, 2 colors) and mode 12 (640 × 480, 16 colors). Drivers are included to run Microsoft Windows, Digital Research Gem and AutoCad at resolutions of 640 × 480, 752 × 410 or 800 × 560. For these higher resolutions, though, a multi-sync monitor is required.

Installing the EGA Wonder is very simple. Disable the built-in video on the Zenith

158's video/floppy controller card by moving section 4 of switch SW301 to the off position. Next, set the EGA Wonder switch setting for the CGA (80 × 25) video mode, RGB color monitor and enhanced features switch disabled (on-on-on-off-on-on-off-off). A table is included for other video mode/monitor combinations.

The Z-158 will not boot up properly with the EGA Wonder's enhanced features switched on. (The computer will complain about the 8087, which is not present in my system.) Therefore, disable EGA Wonder's enhanced features by turning switch 8 off and enable EGA Wonder's features by running the "SuperSwitch" utility program, typing:

```
SMS ENAB
```

at the DOS prompt. You can enable the EGA Wonder board automatically by placing the above line in your AUTO-EXEC.BAT file.

A caveat; the EGA Wonder uses interrupt level 2. If characters or graphics appear distorted on the screen, it may be due to another peripheral, such as a serial mouse or a real-time clock that is also using interrupt level 2. Because of this, I had to buy the Logitech Bus Mouse so that I could change the interrupt level of the mouse. Logitech's serial mouse, which does not work with the EGA Wonder under certain

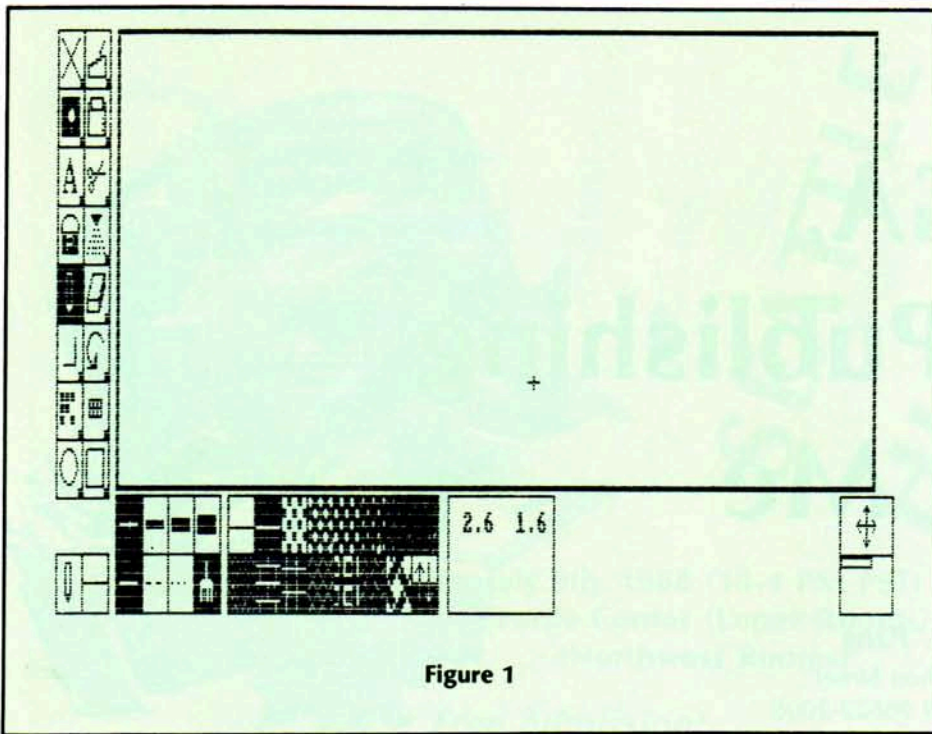


Figure 1

video modes, uses interrupt level 2 and could not be changed.

The feature that most attracted me to the EGA Wonder was the ability to run EGA programs and have them displayed on an RGB monitor. They accomplish this feat by interlacing the video signal. This is where alternate scan lines are used to refresh or update the (phosphor on the) screen.

For an RGB display, the scan lines go down the screen from 0 to about 200, while pixels will go across the screen from 0 to about 640 (the exact numbers being dependent on the resolution of your monitor). In interlace mode, the "in between" scan lines are used, "doubling" the resolution.

scan line	column 0123456...	scan line	column 0123456...
0	0
1	1
2	3
		4

normal display interlaced display

With interlacing, the even-numbered lines are refreshed during one cycle (every 60th of a second), while the odd-numbered lines are refreshed during the next cycle. In the meantime, the phosphor making up the even-numbered scan lines

is fading (as they are not refreshed). This will cause a flickering effect which is especially noticeable with the color white. With my ZVM-135 and a polarizing filter, I find the interlacing tolerable. Better results could be achieved on a monitor with long-persistence phosphor.

Although resolution is doubled, the image will come out "small" and will not fill the screen as we are now squeezing twice as many lines in the same space it takes to display 200 lines. While acceptable in many applications, this will not be suitable for critical CAD/CAM applications. Displaying 132 columns on the RGB monitor, such as in a spreadsheet application, is unsatisfactory.

HALO DPE

HALO DPE (Desktop Publishing Editor) version 1.2, written by Media Cybernetics, Inc., is a program that enables users to merge text with graphics onto a single page. DPE uses the concept of the "virtual page," allowing you to view and edit the entire page at once, or zoom in on a particular section as a "window" into the virtual page. Numerous tools are provided to manipulate text and graphics. For example, more than a dozen different type styles are provided. With some type styles, you can scale the font to any point size desired. The characters can also be rotated, unfilled, underlined and shadowed. DPE provides a number of powerful drawing tools not found in "ordinary"

paint programs, such as curve fitting of polygons, three dimensional rectangles, different airbrush densities, negative reproduction of an image, flipping an image and rotation of an image.

Support is provided for a variety of printers, optical scanners, video cards and pointing devices. Also included is a multi-board screen grabber utility (to capture a graphics image in any display mode) and a "slide" show program.

Installation of DPE is straight forward. For a hard disk, all you have to do is make a subdirectory and copy the distribution disks (three of them) to the hard disk. For a floppy system, you will have to copy the contents of the master disk to a floppy and then copy the appropriate device drivers (for your system) to the floppy. A table of device type and file name is given to aid with this process. The SETUP program is then run after copying all the files over.

HALO DPE is entirely icon driven (Figure 1). Their icon concept is a little different from the pull down menus found with Windows and GEM, though, and it took me a little time to get accustomed to their conventions. Selecting an icon may bring up other icons, depending on which button on the pointing device was pressed. I will briefly describe a session with DPE, but will not go into details as the commands issued will depend on the type of pointing device used (i.e., brand of mouse, digitizer, keyboard, etc.).

Because DPE does not have any text editing capabilities, you will have to create the textual material with your favorite word processor and save it (corrected and edited, of course) as an ASCII text file. You should really let the word processor do the bulk of the formatting (e.g., column width, creation of double columns, inserting space for graphics, etc.), as DPE does not provide any text formatting tools.

Graphics can be created with the drawing tools provided by DPE, input from a scanner or captured from the screen using the GRAB utility.

You are now ready to create the page. First, select a type style and font size by choosing the letter icon. A menu will be displayed with the different font styles (Figure 2). Select a point size. Exit from this mode and read in the ASCII text by choosing the diskette icon. Another set of icons will be displayed. Select the text file

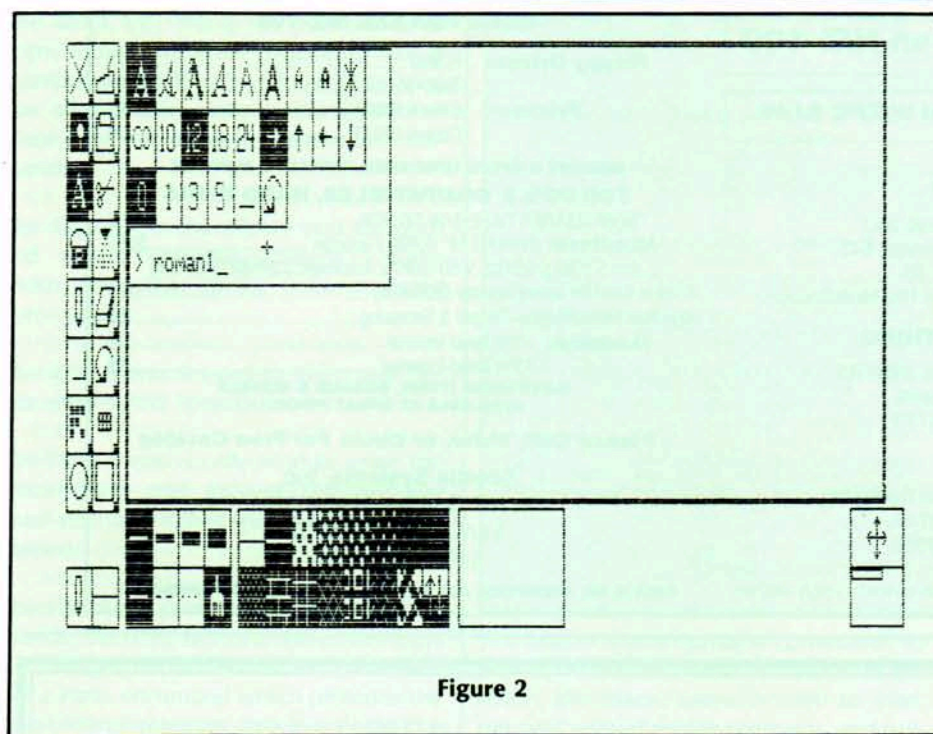


Figure 2

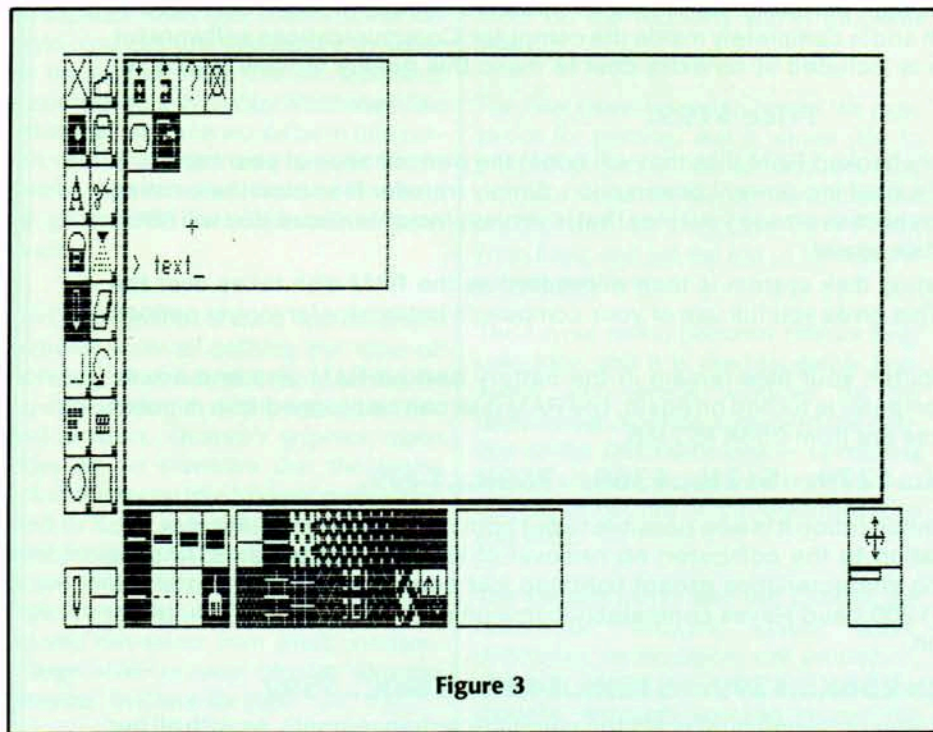


Figure 3

icon and then click on the disk icon with the upward pointing arrow to retrieve the text (Figure 3). A box will be displayed on the screen indicating the space on the page occupied by the text. Position the box on the page and deposit the text. You may want to be in virtual page mode (a representation of the whole page) when performing this operation.

Read in the graphics from disk and position it on the page, following steps similar

to that of reading in text. You can change the size and proportion of the picture as necessary.

Save the page by choosing the disk icon. Another set of icons will be displayed. Select the disk icon with the arrow pointing downwards. Finally, exit DPE by choosing the "X" icon from the main screen and then the dot icon that is subsequently displayed.

Conclusion

For a person interested in getting into the "EGA standard," the ATI EGA Wonder seems to be well worth its price. It is less expensive than many other multi-mode EGA cards and has a number of interesting features and capabilities that distinguish it from other video cards. Although I am currently using it with an RGB monitor, it appears that the EGA Wonder will allow me to upgrade to an EGA or multi-sync monitor with no problem, and even get resolutions higher than that with standard EGA.

Because DPE can only work on a single page at a time, it is not suitable for long documents, or even short ones. Modification to textual material is very difficult, if not impossible. For simple "desktop publishing" requirements (e.g., one-page material, such as overhead transparencies), HALO DPE should do nicely. DPE does have a good set of graphics tools and can be used to create graphics for Xerox Ventura Publisher. (Not mentioning Page-Maker was intentional, as it will not run on my machine.)

Products Discussed

HALO DPE	
Desktop Publish Editor	\$195
Media Cybernetics, Inc.	
8484 Georgia Avenue, Suite 200	
Silver Spring, MD 20910	
(301) 495-3305	
ATI EGA Wonder	\$399
ATI Technologies Inc.	
3761 Victoria Park Avenue	
Scarborough, Ontario	
CANADA M1W3S2	
(416) 756-0711	

Trademark Information

Many of the vendor and package names referred to in this article are registered trademarks, service marks or trade names.





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-55	95.
Printers:	Citizen 180-D	195.
	Citizen 120-D	179.

MEMORY & SPEED UPGRADES, SERVICE & SUPPORT.

FOR PC's & COMPATIBLES, HARD DISKS

NEW IMAGER TAPE B/W TO VCR	\$210.
Monitors: NANA0 14" 8060S Flexscan	\$585.
(H-15.75Khz-35Khz, V-50-80Khz Automatic 0.28MM Dot Pitch)	
Adapter Card for above monitor (800x600)	179.
High Res Monochrome-Casper & Samsung	89.
Modems: 1200 Baud Internal	79.
1200 Baud External	99.

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

INTERNAL MODEM A low cost, high performance, Hayes compatible internal modem. Operates at 1200 or 300 baud. It installs in a flash and is completely inside the computer. Communications software for file transfer and terminal emulation is included at no extra cost to make this quality modem truly a remarkable value.

Price \$199.

INTERNAL RAM DISK A battery backed RAM disk that will boost the performance of your lap top. It provides super fast disk access while slashing power consumption. Simply transfer files from the existing disk system to the RAM disk and enjoy access time to your files that is virtually instantaneous. You will never need to wait for the floppy or hard disk again!

The power consumption of the existing disk system is then eliminated as the RAM disk takes over the function of the floppy or hard disk. This gives you full use of your computer's battery for far longer periods away from a power source.

And when you shut down the computer, your files remain in the battery backed RAM disk and are instantaneously available when the computer is turned on again. The RAM disk can be plugged-in in minutes and is completely internal. Ram sizes are from 256K to 2MB.

Prices 256K...\$279. 512K...\$369. 768K...\$499.

COMBO Through advances in miniturization it is now possible to put both the modem and RAM disk into the 181/183 together. No modification to the computer; no removal of existing components. And no changes in the computers operating characteristics except lightning fast operation at a fraction of the power consumption with complete 1200 baud Hayes compatible communication capability. Completely internal with easy plug-in installation.

Prices: Modem with 256K...\$399. 512K...\$489. 768K...\$599.

American Cryptronics is in its fourth year providing quality lap top computer enhancements. As with all our products the modem and RAM disk come with complete instructions and all required software is included at no extra charge. Our goal is your satisfaction. We proudly offer these enhancements with a 30 day full refund, satisfaction guarantee. All our products carry a one year warranty.

VISA, MasterCard, Check or Money order
 UPS Ground add \$2.00
 UPS 2nd Day Air add \$4.00
 COD add \$5.00
 CA residents add 6% Sales Tax

AVAILABLE NOW...To Order Literature
Call (714) 540-1174 FAX (714) 540-1023

AMERICAN CRYPTRONICS INC.

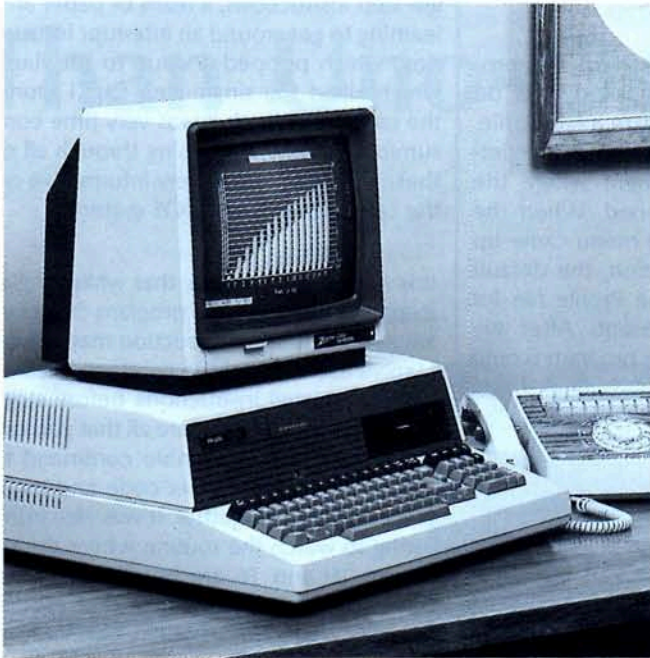
1580 Corporate Dr., Suite 123
 Costa Mesa, California 92626

ZPC Update

#22

Patrick Swayne

HUG Software Engineer



This is the twenty-second in a series of articles in support of ZPC, a program that allows you to run IBM PC software on H/Z-100 (dual processor) computers. ZPC is available from HUG as part no. 885-3037-37. An upgrade disk for ZPC is also available as part no. 885-3042-37.

This ZPC Update article is in the form of a letter submitted by Ronald J. Vince of Springfield, Ohio. He supplied patches for Quick DOS Version 2 and PCFile +. Everything seems to be in order in his patches, but you should be advised that I have not personally tested either program. Here is the text of his letter.

Dear Pat,

I'm sending in this article so that you can use whatever part of it you wish in your next ZPC Update. Also, it allows me to contribute some information back into the Heath Users' Group from which I have received so much.

First of all, this was my first attempt at debugging any programs written for the IBM so that they could be used with ZPC on my Z-100. To say the least, it has been quite a learning experience.

After reading the instructions in the ZPC manual for debugging programs, along with the examples in the ZPC Update arti-

cles, I felt sure that with a little work, I too would be able to patch the programs. The interest had become a lot greater to have these programs working and installed on my new SCSI 65 Meg Seagate hard disk and CDR Disk Controller from Ram Technology which was not divided into two 32 meg partitions, one for the Zenith Z-100 and one for IBM emulation.

Quick DOS Version 2

First I would like to submit the patches for Quick DOS Version 2 which includes the editor program QED.EXE. Since I do not have the ZHS card in my system, I started by following the examples for patching Quick DOS Version 1.21 found in ZPC Update No. 13, dated Feb. 1987.

The patches that were needed are not in the same area as for version 1.21, but this was used as a guide in looking for the same pattern of changes. Not very scientific, but it helped. By adding these lines to the PATCHER.DAT file, the program seems to work just fine.

```
Q-DOS II version 2.00
Insert the disk containing QD2.EXE,
QDSTART.EXE, QDCOLOR.COM and QED.EXE
QD2.EXE
3C0,90
1CCF,90
304C,90,90,90,90,90,90,90,90,90
8AFA,90
```

```
9F66,90,90,90,90,90,90,90,90,90,90
9F82,90,90,90,90,90,90,90,90,90,90
9FB1,90,90,90,90,90,90,90,90,90,90
DD48,90
x
QDSTART.EXE
FDE,90
FEF,90
1BBA,90,90,90,90,90,90,90,90,90,90
1BF1,90,90,90,90,90,90,90,90,90,90
1COD,90,90,90,90,90,90,90,90,90,90
x
QDCOLOR.COM
B7,90
1E0,90
231,90,90,90,90,90,90,90,90,90,90
598,90
5A5,90
5EA,90,90,90,90,90,90,90,90,90,90
x
QED.EXE
9CD,90
189F,90,90,90,90,90,90,90,90,90,90
18D6,90,90,90,90,90,90,90,90,90,90
18F2,90,90,90,90,90,90,90,90,90,90
27C8,90
3E49,90
z
```

Now, as before, make the patch with PATCHER and then run QDSTART to install QDOS II as described in the manual.

One thing I found that is very useful in learning to make these patches, is having examples of the original code to follow and see why the patch was needed. Your examples have been a lot of help to me in my trying to learn the programming code

used. If all people would take the time to supply this information with their submitted patches, it would be great. As you can see, I did not follow my own wishes with this first program.

Implementing PCFile + On The Z-100

Next, I would like to get into the horror story of debugging PCFile +, Version 1.0. In the November 1987 Issue of REMark, Joseph Katz gave a review of PCFile +. Being a little familiar with its use from the office on the IBM, I wondered if it could be used on the Z-100. After having patched the QDOS program, I thought I was now really on my way to understanding things.

***** "WRONG!" *****

This, I must say, was really a bear. Every piece of literature I have on programming was used. The Programmers' Utility Package was found invaluable to have with its SYMDEB.EXE program, along with the BRKOUT.COM program found as one of the utilities on the HUG DEBUG SUPPORT program disk.

There are still some questions about what I found that I have not been able to answer or understand. Maybe you can shed some light on this.

After loading the programs to drive E, which is the Z-100 default drive for the Hard Disk, and entering ZPC Mode 7, the program complained of an error writing on the disk and refused to go any farther. With this I returned to the Z-100 Mode and entered the assign command to let C=E. Returning to the program, it gave me a Path Error of C:\PCFILE, and then asked for the drive letter and Path Statement of the database. The program accepted my input and then gave me a list of database files to choose from.

The manual states that the Profile file (PCFILE.PRO), which is a configuration file for PCFile +, must be changed to define the default drive and path statement in order for it to be able to find its database files on start up and avoid the above error. Within the program there is a menu that takes you to a selection of utilities which then allows a selection for the Profile files for set-up configuration. This is where the big problem begins.

Additional testing of the program indicated an error when I tried to send a report to the printer. Again, the configuration for the printer lies within the Profile

file. In attempting to enter the Profile file to change the default settings, either from within the program at the menu selection or from entering the PCF SETUP command from the start, the block wall was hit.

Upon entering this portion of the program, the first thing that it asked for is, do you want to modify or create a Profile file. I entered modify along with the drive letter E and a path statement where the Profile files were positioned. When the appropriate places in the menu came up for entering this information, the default values, as defined by the Profile file for PCFile were always present. After my entries were entered, the program would always come back with a path error showing the path statement as entered, but the drive letter character would be changed. Sometimes a drive D: would be shown and sometimes a graphics character was displayed. So with this, off to debug land I went.

Upon loading the program from the SYMDEB command line and seeing the message "Warning Packed Code", I knew I was in for trouble.

First, I looked for the usual things that are described in the ZPC manual. By finding an area which accesses the video port 03AD, with the TEST, IN, JNZ and JZ instructions, I applied the NOP patches as shown in the manual, except for the CLI instructions. Retesting the program, it would now operate in ZPC Mode 2, 3 or 7, but I could not get around the path error.

Going back into debug and looking for a conflict of the Graphic Character Table proved to be fruitless, but I did find some accesses to the keyboard ports 60 and 61. The patches were applied to call interrupts 90 and 91 as supplied with ZPC. This probability did some good in making the key entry more correct, but still did not help the path error problem in the Profile set-up menu.

At this point, I decided to load the Profile files to the floppy drive B:. Then when drive B: was entered with only the root character "\ " entered for the path statement, the program worked correctly and I was able to set the default drive and path statement in to the Profile file and copy it back to drive "E". Now I tried running the Profile file set-up portion of the program again and found the default settings, as I had entered them, did appear. This

also allowed the program to find the database files. The path error problem, however, was still there.

After many frustrating hours of debug single-step instructions, a ream of paper and learning to get around an interrupt instruction, which popped a value to the flags, which killed the single-step flag, I found the cause. Although it was very time consuming, the single-stepping through all of these instructions was very informative on the operations of the DOS system.

It is interesting to note, that while single-stepping through the program in some areas, a single-step instruction may actually solve each step of the program, but skip through several instructions before stopping to display them. Here all that you can do, is use the unassemble command to display the skipped over code and try to piece the steps together. It was also interesting to watch the routine where the interrupt 10 and 16 are handled and observe the code segment of that routine actually change its code value at one memory location, depending on the interrupt being set up.

The printer must be configured for output on the COM1 port for it to work, because the program goes to the DOS system looking for an LPT1 file which it could not find.

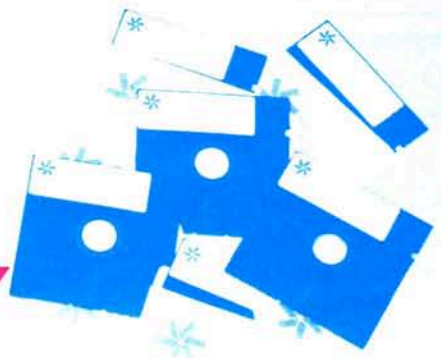
The problem with the drive letter change and path error turns out to be caused by the entry of a path statement other than a root directory. Determining which entry was at fault was more than half of the battle. Testing showed, that if a path to a subdirectory, even on floppy drives A and B, causes the error. This is because, if a path statement is entered, the program takes the entered drive letter and looks for a match in a table located in segment 005C:002F, which is somewhere above the start of the IBMBIOS segment 0040. If a match is found of the ascii drive letter character, or if 00 is found, the program then takes this table location pointer value and goes to another table beginning at segment 0010:0001, with an offset value from the first table, and returns this value to the program's drive and path statement buffer.

At this point, I do not understand what these tables are. I think the second table must be part of the interrupt vector table. What the first table is, I haven't a clue.

Continued on Page 64



HUG NEW PRODUCTS



- 10 - Very Good
- 9 - Good
- 8 - Average

TABLE C Product Rating

Rating values 8-10 are based on the ease of use, the programming technique used, and the efficiency of the product.

- 7 - Hardware limitations (memory, disk storage, etc.)
- 6 - Requires special programming technique
- 5 - Requires additional or special hardware
- 4 - Requires a printer
- 3 - Uses the Special Function Keys (f1, f2, f3, etc.)
- 2 - Program runs in *Real Time**
- 1 - Single-keystroke input
- 0 - Uses the H19 (H/Z-89) escape codes (graphics, reverse video)

Real Time — A program that does not require interactivity with the user. This term usually refers to games that continue to execute with or without the input of the player (e.g., 885-1103 or 885-1211[-37] SEA BATTLE.

ORDERING INFORMATION

For VISA and MasterCard phone; telephone Heath/Zenith Users' Group directly at (616) 982-3838. Have the part number(s), description, and quantity ready for quick processing. VISA and MasterCard require minimum \$10.00 order. By mail, send your order, plus 10% postage/handling (\$1.00 minimum, \$5.00 maximum) to: Heath/Zenith Users' Group, P.O. Box 217, Benton Harbor, MI 49022-0217. Orders may be placed, by mail only, using your Heath Revolving Charge account. Purchase orders are also accepted by phone or mail. No C.O.D.s accepted.

Questions or problems regarding HUG software or REMark magazine should be directed to HUG at (616) 982-3463.

NOTES

The [-37] means the product is available in hard-sector or soft-sector. Remember, when ordering the soft-sectored format, you must include the "-37" after the part number (e.g., 885-1223-37).

All special update offers announced in REMark (i.e., ZPC II update) must be paid by check or money order, payable to the Heath Users' Group. **NO CREDIT CARDS ACCEPTED.** ZPC II contains only one disk. It is a combination of ZPC I and the ZPC Support disk, plus added improvements. Thank you.

HUG P/N 885-3048-37 Both Sides Printer Utility..... \$20.00

The Both Sides Printer Utility is a program that lets you print disk text files that are formatted into pages on both sides of the paper. With Both Sides, you can print all of those "shareware", public domain, and HUG program .DOC files using half the amount of paper and half the notebook space that you would normally have to use.

Both Sides works by printing all of the odd pages (1, 3, etc.) first. Then, it stops and allows you to reload your paper with the other side facing the print head, and then it prints the even pages (2, 4, etc.). Both Sides is very easy to use, and can print several files with one command. If you have a lot of .DOC files to print, it will pay for itself in saved paper and notebook space.

Requirements: Both Sides will run on any MS-DOS or PC-DOS based computer (including the Z-100) with at least 32k free memory and MS-DOS or PC-DOS version 2 or above.

Program Author:
Patrick Swayne, HUG Software Engineer

The Both Sides disk contains these files:

```

README .DOC
BS      .DOC
BS      .COM
REPRINT .COM
WRAP    .COM
BS      .ASM
  
```

```

REPRINT .ASM
WRAP     .ASM
  
```

Here is an explanation of the files:

BS.DOC — This file contains the instructions for using the Both Sides utility. And yes, you can use Both Sides to print this file on both sides of your paper.

BS.COM — This is the Both Sides Printer Utility.

REPRINT.COM — This program allows you to redirect the printer output from a program to a disk file. Although it has the same name as the REPRINT utility on HUG disk 885-3025-37, it is a completely new program. It will work with virtually any program that prints, and it is faster than the old REPRINT. With this program, you can prepare disk files for processing with Both Sides that contain the special codes that your word processor would normally use when it prints directly to a printer. **Note:** Files containing formatting codes for Diablo and compatible printers may not be handled correctly by Both Sides.

WRAP.COM — Both Sides is operated by entering the files to be printed on the MS-DOS command line. A command line containing a number of files could be longer than the width of your screen. On a Z-100 (not PC) computer, characters typed past the end of the line will be lost unless the "wrap at end of line" feature is enabled. The WRAP program allows you to enable or disable the wrap feature, as needed.

BS.ASM, REPRINT.ASM, WRAP.ASM — This is the Assembly source code for the programs on this disk.

TABLE C Rating: 4, 10



HUG Price List

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.

PRODUCT NAME	PART NUMBER	OPERATING	DESCRIPTION	PRICE
		SYSTEM		
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
ASCRTY	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	885-8009-37	CPM	GAME	20.00
GAMES 1	885-1029-37	HDOS	GAMES	18.00
HARD SECTOR SUPPORT PACKAGE	885-1121	HDOS	UTILITY	20.00
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 1982	9.75
HUGMAN & MOVIE ANIMATION	885-1124	HDOS	ENTERTAINMENT	20.00
INFO. SYSTEM AND TEL. & MAIL SYSTEM	885-1108-37	HDOS	DBMS	30.00
LOGBOOK	885-1107-37	HDOS	AMATEUR RADIO	30.00
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 TRANSCEIVER	885-8016	HDOS	AMATEUR RADIO	20.00
MORSE CODE TRANSCEIVER	885-8031-37	CPM	AMATEUR RADIO	20.00
PAGE EDITOR	885-1079-37	HDOS	UTILITY	25.00
PROGRAMS FOR PRINTERS	885-1082	HDOS	UTILITY	20.00
REMARK VOL 1 ISSUES 1-13	885-4001	N/A	1978 TO DECEMBER 1980	20.00
RUNOFF	885-1025	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	885-1134	HDOS	LANGUAGE	30.00
SOFT SECTOR SUPPORT PACKAGE	885-1127-37	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
VOLUME I	885-1008	N/A	SOFTWARE LISTINGS	9.00
VOLUME II	885-1013	N/A	SOFTWARE LISTINGS	12.00
VOLUME III	885-1015	N/A	SOFTWARE LISTINGS	9.00
VOLUME IV	885-1037	N/A	SOFTWARE LISTINGS	12.00
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	885-3023-37	MSDOS	PRINTER PLOTTING UTILITY	20.00
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
KEYMAC	885-3046-37	MSDOS	UTILITY	20.00
KEYMAP	885-3010-37	MSDOS	UTILITY	20.00
KEYMAP CPM-85	885-1245-37	CPM	UTILITY	20.00
MAPLE	885-8023-37	CPM	COMMUNICATION	35.00
MATHFLASH	885-8030-37	MSDOS	EDUCATION	20.00
ORBITS	885-8041-37	MSDOS	EDUCATION	25.00
POKER PARTY	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
SPELL5	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	885-3008-37	MSDOS	UTILITY	20.00
Z100 WORDSTAR CONNECTION	885-3047-37	MSDOS	UTILITY	20.00
ZBASIC DUNGEONS & DRAGONS	885-3009-37	MSDOS	GAME	20.00
ZBASIC GRAPHIC GAMES	885-3004-37	MSDOS	GAMES	20.00
ZBASIC GAMES	885-3011-37	MSDOS	GAMES	20.00
ZPC II	885-3037-37	MSDOS	PC EMULATOR	60.00
ZPC UPGRADE DISK	885-3042-37	MSDOS	UTILITY	20.00

H/Z-100 And PC Compatibles

ADVENTURE	885-3016-37	MSDOS	GAME	10.00
ASSEMBLY LANGUAGE UTILITIES	885-8046-37	MSDOS	UTILITY	20.00
DEBUG SUPPORT UTILITIES	885-3038-37	MSDOS	UTILITY	20.00
DPATH	885-8039-37	MSDOS	UTILITY	20.00
HADES	885-3040-37	MSDOS	UTILITY	40.00
HELP	885-8040-37	MSDOS	CAI	20.00
HEPCAT	885-3045-37	MSDOS	UTILITY	35.00
HUG BACKGROUND PRINT SPOOLER	885-3029-37	MSDOS	UTILITY	20.00
HUG EDITOR	885-3012-37	MSDOS	TEXT PROCESSOR	20.00
HUG MENU SYSTEM	885-3020-37	MSDOS	UTILITY	20.00
HUG SOFTWARE CATALOG UPDATE #1	885-4501	VARIOUS	PROD 1983 THRU 1985	9.75
HUGMCP	885-3033-37	MSDOS	COMMUNICATION	40.00
HUGPBBS SOURCE LISTING	885-3028-37	MSDOS	COMMUNICATION	60.00
HUGPBBS	885-3027-37	MSDOS	COMMUNICATION	40.00
ICT 8080 TO 8088 TRANSLATOR	885-3024-37	MSDOS	UTILITY	20.00
MATT	885-8045-37	MSDOS	MATRIX UTILITY	20.00
MISCELLANEOUS UTILITIES	885-3025-37	MSDOS	UTILITIES	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	884-4008	N/A	1987	25.00
SCREEN DUMP	885-3043-37	MSDOS	UTILITY	30.00
UTILITIES II	885-3014-37	MSDOS	UTILITY	20.00

PC Compatibles

ACCOUNTING SYSTEM	885-8049-37	MSDOS	BUSINESS	20.00
CARDCAT	885-6006-37	MSDOS	CATALOGING SYSTEM	20.00
CHEAPCALC	885-6004-37	MSDOS	SPREADSHEET	20.00
CP/EMULATOR II & ZEMULATOR	885-6002-37	MSDOS	CPM & Z100 EMULATORS	20.00
DUNGEONS & DRAGONS	885-6007-37	MSDOS	GAME	20.00
EZPLOT	885-6003-37	MSDOS	PRINTER PLOTTING UTILITY	20.00
GRADE	885-8037-37	MSDOS	GRADE BOOK	20.00
HAM HELP	885-6010-37	MSDOS	AMATEUR RADIO	20.00
KEYMAP	885-6001-37	MSDOS	UTILITY	20.00
LASERWRITER CONNECTION	885-8050-37	MSDOS	PRINTER UTILITY	40.00
PS's PC UTILITIES	885-6011-37	MSDOS	UTILITIES	20.00
SCREEN SAVER PLUS	885-6009-37	MSDOS	UTILITIES	20.00
SKYVIEWS	885-6005-37	MSDOS	ASTRONOMY UTILITY	20.00
TCSPELL	885-8044-37	MSDOS	SPELLING CHECKER	20.00
ULTRA-RTTY	885-6012-37	MSDOS	AMATEUR RADIO	20.00

You've got a screen full of important technical data that would be nearly impossible to memorize, and you already have writer's cramps from the last screen full. With **SCREENDUMP** from HUG, you can reproduce a complete video screen on a dot matrix printer, including both text and graphics without having to exit the current program. **SCREENDUMP** supports most of the more popular dot matrix printers, including the newer 24-pin and laser jet models. The latest version of **SCREENDUMP** is **HUG P/N 885-3043-37**.

"Thank Heaven for **HADES**!" That's what a lot of MS-DOS users are saying when **HADES** rescues a file that just got accidentally erased. Erased file recovery is only a small part of the capabilities of this program. **HADES** is HUG's *Absolute Disk Editing System*. Within the realms of MS-DOS, **HADES** allows you to directly edit any part of any disk. Directories, files, file attributes. FATS: nothing can hide from you when you use **HADES**. **HADES** works on ANY computer that can run MS-DOS version 2 or greater. Order **HUG P/N 885-3040-37** today!

Want to keep your H/Z-100? Want to run a lot of that good PC compatible software out there? Don't want to buy a PC compatible though? Then get **ZPC II**, **HUG P/N 885-3037-37**, and the **ZPC II upgrade disk**, **HUG P/N 885-3042-37**.

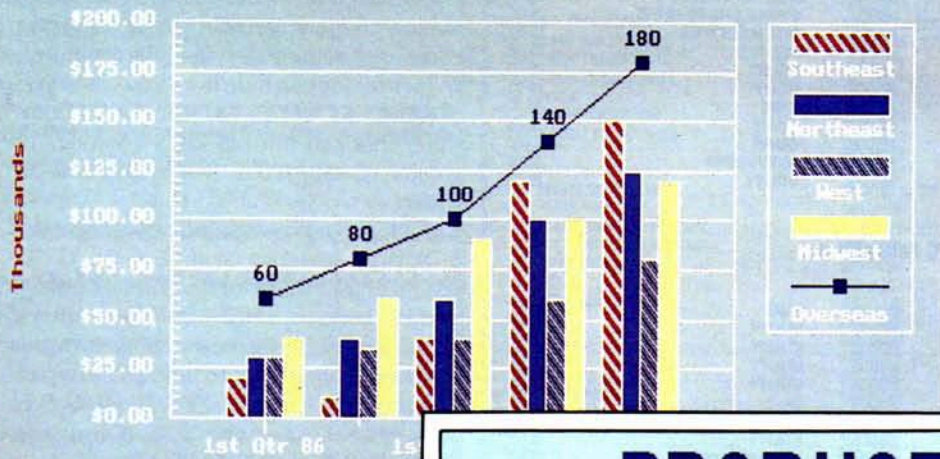
ORDERING INFORMATION

For VISA and MasterCard phone orders, telephone the Heath Users' Group directly at (616) 982-3838. 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.

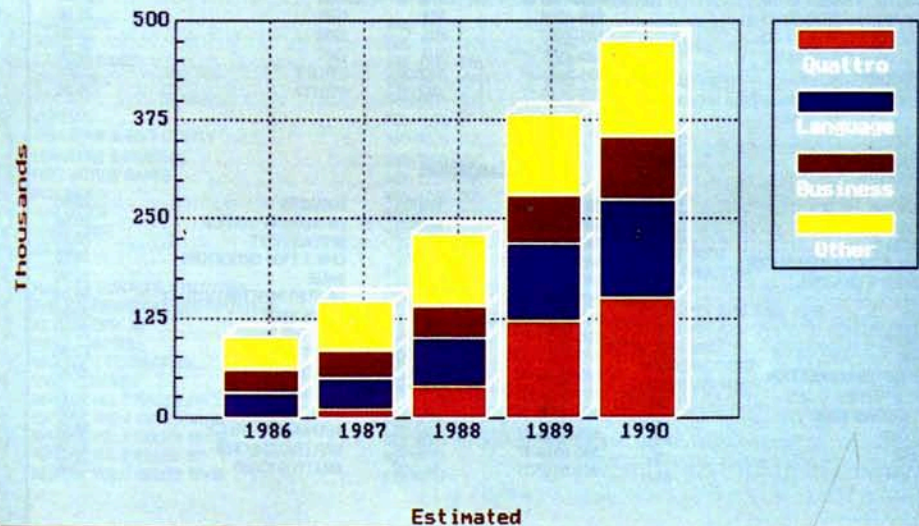
Quattro Is One Better

PRODUCT SALES by Division



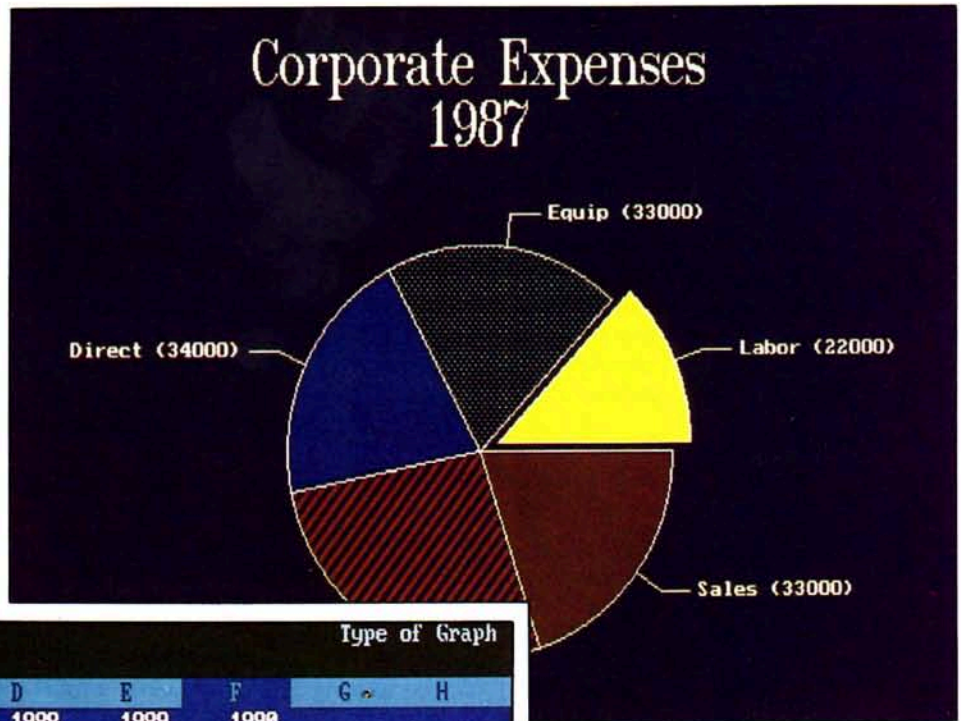
PRODUCT SALES

All Divisions



Than Uno-Dos-Tres

William M. Adney
 P.O. Box 531655
 Grand Prairie, TX 75053-1655



	A	B	C	D	E	F	G	H
	1986	1987	1988	1989	1990	Type of Graph		
Revenues						Graph		
Quattro	0	10				Graph Type	Stacked-Bar	B1..F1
Languages	30	40				Line		
Business	30	35				Bar		
Other	40	60				3 Dimensional Bar		
Total	100	145				XY		
Product Cost						Stacked-Bar		
Books	1	2.5				Pie		
Disks	1	2.5				Area		
Wrap	0.1	0.3				Rotated Bar		
Assembly	0.1	0.5				Markers		
Total Cost	2.2	5.8				Combined Lines and Markers		
Other Costs	40	87						
Total Costs	42.2	92.8	143.4	233.7	290.5			
=====								
Total Profit	57.8	52.2	86.6	146.3	184.5			

F4: 125
 16-Sep-87 06:55 PM MENU

There are a few companies that seem to have a knack for introducing a computer product that is truly innovative and goes far beyond what many people expect. You can count those companies on the fingers of one hand, and when you do, Borland is the one that is always included in everyone's top five. Borland started its claim to corporate fame by introducing the popular SideKick desktop utility program. Today, Borland has introduced the new Quattro spreadsheet that promises to be at least as popular as SideKick.

I can hear it now. "Oh no," you're saying, "not ANOTHER spreadsheet!" Well, Quattro is not just another spreadsheet — it has a combination of features that you can't find on any other spreadsheet today. Let's take a look at some of the general features before we get down to details.

Quattro Is Fast!

Borland designed Quattro so that you get the performance of a high-speed race car, instead of the old four-door family sedan. In many cases, Borland says that Quattro can retrieve, load, and read spreadsheet files faster than Lotus 1-2-3, and based on my testing, I believe it.

I checked out Quattro's performance on a number of different Zenith computer models ranging from the old PC-compatible Z-151 to my own AT-compatible Z-248. Although I used a fairly simple template that I developed for computing taxes, I was able to see a noticeable improvement in the file load and recalculation time with Quattro as compared with Lotus 1-2-3 version 2.01. In fact, the slower the system (e.g., a Z-151 with two floppy disk drives), the more obvious the performance improvement became. Although I did not do any specific timing of these functions, I believe that most users are more interested in performance improvements they can SEE, instead of some nebulous benchmark numbers.

For most of my testing, I used Quattro on my own Z-248 with 640 K of memory and a 30 millisecond hard disk. I use an NEC Multisync monitor with a Vega 7 EGA video card since I have found this combination is best suited for my purposes. Quattro seems to like this configuration, and it performed beautifully. But Quattro has a couple of special features that move it from the realm of the simply excellent to the truly outstanding.

Quattro Is Flexible

Corporate America and many users shudder with trepidation when a new micro-computer product is announced, especially a new software product. Training costs, in terms of employee or user time spent in learning to use the product, can amount to many times the basic cost of the software or hardware. When software is introduced that has new and sophisticated features, sometimes this fear can turn into reality, and occasionally, even a nightmare.

Fortunately, Quattro is so flexible that it reduces this training problem to no problem at all. Quattro has a "soft" user interface that allows you to build any kind of menu structure that you want or need. Borland has thoughtfully included a Lotus 1-2-3 menu tree file so that current users will not have to learn a new command structure. Or, if you are a SuperCalc fan as I was, you can use the built-in Menu Builder function to create a SuperCalc menu tree. You can use the Menu Builder program to create any type of menu tree you want or you can use the normal Quattro menu tree that took me about 10 minutes of review to use effectively. But that is not the end of Quattro's flexibility.

One of the other major problems of implementing new software is the cost and time required to perform file conversion to the new format. In some extreme cases, this may require that all data be retyped using the new software, and the conversion can be extremely costly and time consuming. Quattro provides a solution for that problem, too.

For Lotus 1-2-3 users, Quattro can directly read and write files in the 1A, 2.01, and educational version file formats. Quattro is even compatible with Lotus 1-2-3 macros so that neither retraining nor file conversion costs are a problem. But now, let's take a look at how you can use Quattro.

Using Quattro

I used Quattro to prepare some tax information, although I already had a spreadsheet template prepared with SuperCalc. This gave me the opportunity to learn more about Quattro, and I found it fairly easy to learn. I say "fairly easy" because Quattro seems to use most of the Lotus 1-2-3 conventions, and it took me a couple of minutes to remember how that was done. For example, you usually begin formulas with a plus sign (+), such as +b3/b4. If you don't, Quattro assumes you want text instead of a formula for calculation purposes.

Quattro uses the pop-up menus and has an easy-to-learn and logical menu structure if you are familiar with any spreadsheet. The main pop-up menu is simple, and a sample is shown as Figure 1. The actual screen uses standard PC graphics characters and is much nicer, although I have shown the menu surrounded by the normal hyphen (-) and vertical bar | ASCII characters.

The main Quattro menu is activated by pressing the slash (/) key like many other spreadsheets. The menu shown in Figure 1 is the standard Quattro menu, but you can change any or all of the commands and the menu structure.

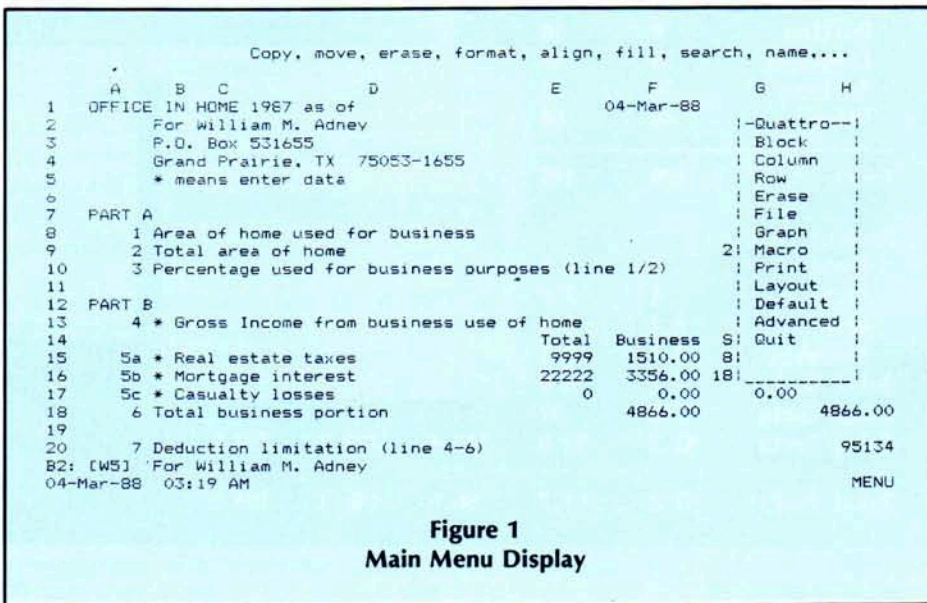


Figure 1
Main Menu Display

The Block (/B) menu has commands to copy, move, and erase blocks, as well as manipulate block names. In addition, you can align labels, perform the search/replace, and use other less common commands.

The Column menu allows you to insert and delete columns, change column width, and other similar column-related commands.

The Row menu is used to insert and delete rows in the spreadsheet.

The Erase menu is only used to erase the spreadsheet and recover the memory used so that another spreadsheet can be loaded.

The File menu performs file-related commands, such as retrieve, save, combine (i.e., merge), import, and erase. This menu has a Parse command which provides the capability to separate data in a single line imported from a standard ASCII text file into separate fields (i.e., columns). For example, you can use this feature to separate the various fields that are separated by commas in a WordStar MailMerge file so that the last name would be in one column and the first name would be in another. You can also display a directory and go to DOS to enter various commands.

The Graph menu is used for all graph functions, such as defining the type of graph (e.g., line, bar, pie, etc.), adding titles, customizing it, and printing the finished product. Quattro's graphics capabilities are so extensive that the entire back of the menu tree fold-out page is devoted to it. For example, there are 11 different fonts available: default, bold, triplex, san serif, small, simplex, triplex script, script, EuroStyle, complex, and gothic. And you can select from small, medium or large size; or you can let Quattro "autosize" the font for you.

Quattro has a nice selection of graph types. Figure 2 shows an example of the menu display with the /GG command, and this is just one example of the extended graphics features that are available. Again, please note that ASCII (the - and \) characters are shown surrounding the menus, although the actual screen display uses professional-looking graphics characters. Also, notice that Quattro can display a wide variety of graph types that you can use for data display.

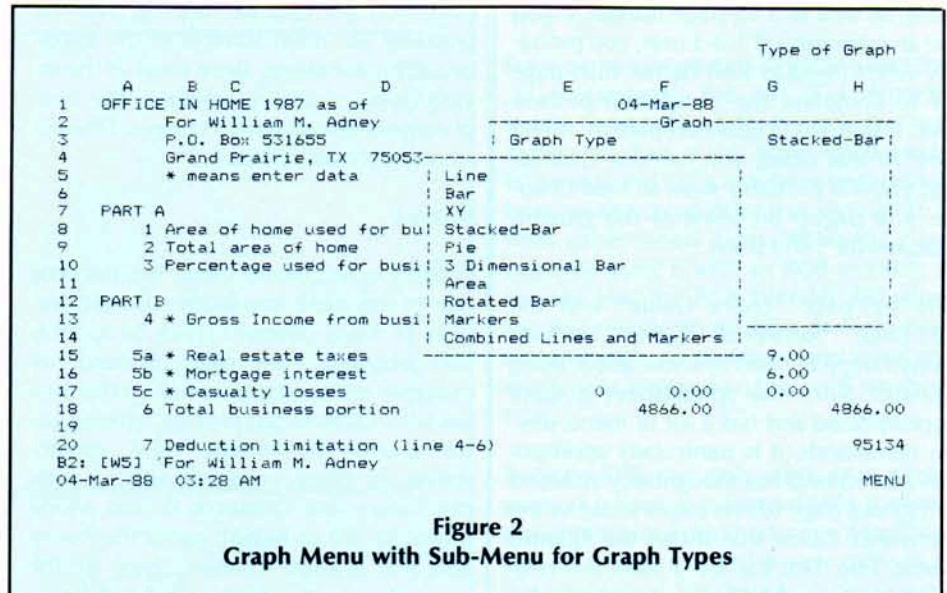


Figure 2
Graph Menu with Sub-Menu for Graph Types

The Macro menu contains commands to Name, Delete, and Execute macros. In addition, the Macro menu is used to load, run, and unload add-in programs, and two such programs are supplied with Quattro. More on the included add-in programs later.

The Print menu is used to format the page layout for printing, and it allows you to print all or part of the spreadsheet. You can also send some commands to the printer, such as Skip Line (Line Feed), Form Feed, and set the top of form to the current position.

The Layout menu performs title locking/unlocking, and it is used to define horizontal and vertical windows for the spreadsheet. You can also adjust the position of the Descriptor Line — containing the cell address, width, and contents — to the top or bottom of the spreadsheet as you wish.

The Default menu sets the Quattro defaults for hardware, colors, dates, directories, recalculation, cell protection, a startup macro called \0, and save the changes. Although you can change the hardware configuration with this menu, there is really no need because Quattro checks out your hardware each time. For example, it checks to see what kind of video card you have so the display is adjusted for the best "picture" — a nice feature. You can also change any or all colors on the spreadsheet, which can result in some spectacular displays.

The Advanced menu performs sorting, database search, and allows you to set up a form for easy data input to a database.

Although the sort feature is actually found under the Database sub-menu, it works just fine with a spreadsheet, too. If you do any work with a matrix and need to invert a matrix or multiply two matrices, you can do that quite easily for operations research problems. In addition, you can also create a sensitivity analysis table called (What-if), a regression analysis table or a frequency distribution table. I suspect that some of my college students would kill for some of these features to make matrix algebra easier for solving operations research problems.

Last, but not least, is the Quit command, which of course exits Quattro to DOS. These are a few of the major commands, and I have not discussed all of them available through the sub-menus.

Software Installation And Documentation

Quattro arrived with four 360 K disks: the System Disk, Resource Disk, Help Disk, and Add-Ins Disk. Since Quattro is not copy-protected, installation is easily accomplished with the COPY command. I installed Quattro on my Z-248 hard disk in the \QUATTRO subdirectory using the COPY command with no problems. You can also use Quattro on a floppy disk system with batch files provided for that purpose.

Documentation for Quattro includes three perfect-bound manuals, two keyboard templates (for the 84- and 101-key keyboards), and a reference card. The 109-page "Getting Started with Quattro" manual includes the installation procedure for both floppy and hard disk sys-

tems, as well as a 65-page tutorial. If you are an experienced 1-2-3 user, you probably won't need to read farther than page 14 to complete the installation process and install the Lotus-compatible menu tree. At that point, you can use Quattro, but you will probably want to read Chapter 4 (8 pages) on some of the Quattro features for 1-2-3 users.

The 439-page "User's Guide" and the 333-page "Reference Guide" include everything you need to know about using Quattro. Since this spreadsheet is quite sophisticated and has a lot of menu-driven commands, it is particularly appropriate that Borland has thoughtfully included a fold-out page (three pages wide) in the Reference Guide that shows the Quattro Menu Tree. Lest the 1-2-3 users feel neglected, a second fold-out page shows the 1-2-3 menu tree with the Quattro extensions — since I don't use 1-2-3 very much, I found this page particularly enlightening. From a documentation perspective, this is probably the best way I've ever seen to understand how a program works with its menus. As if that weren't enough, this sheet can be used like an index because reference page numbers are included in the menu block displays. This is a remarkably effective visual way to present the menu structure, and other vendors would do well to emulate it.

Since I have used a number of spreadsheets, I didn't have any difficulty learning the Quattro menu structure, and I suspect that the learning curve for current 1-2-3 users will probably amount to less than half an hour which also includes installation.

Once you get Quattro running, you will probably not need to refer to the excellent documentation, since most of the information is available on-line in the form of context-sensitive help screens. One example is shown in Figure 3.

Macros

In many ways, macro usage has become one of the more important software features of many different types of application programs. Here again, Borland has managed to "out-Lotus Lotus" in this area because Quattro can process existing Lotus macros in current 1-2-3 spreadsheets. But there is more. In addition, you can simply use Quattro's Record Mode (press ALT-F8 to begin), name the macro and the storage location, type in the keystrokes for the macro, and exit (press ALT-F8 again) the Record Mode. You can also write a macro as you do in 1-2-3, but the Record Mode is much easier. But suppose you are having a problem with a macro that you have not been able to locate.

Then you can use Quattro's debugging facility to single-step through a macro or set breakpoints. If you are not familiar with the term "breakpoint" for a debugger, it means that macro (or program) execution is stopped or suspended when a specified cell (i.e., address) is reached. Since this is an innovative feature for a spreadsheet, let's look at the macro debugging facility in a little more detail.

Macro Debugging

You can set up a macro in the usual way by using the Record Mode or by typing

the commands. Then, you can activate the macro debugger by pressing SHIFT-F8 which opens a special debugging window called the Trace window. By pressing the SPACE BAR, you can single-step through the macro which stops after each command or step and watch the effect on the spreadsheet in the Trace cells. Pressing the RETURN key executes the macro at full speed.

You can also set a standard breakpoint which suspends macro execution when a specified cell or block of cells is reached. Or you can specify a conditional breakpoint which suspends macro execution when a cell containing a "condition" is reached. I can imagine all kinds of interesting uses of macros in solving operations research problems, and the macro debugger would allow a student to step through a calculation to see the effects of changes in a solution.

I think the macro debugging facility is one of the outstanding features of this program, and its capabilities alone would justify switching to Quattro.

Quattro Functions

Since you may be interested in exactly what built-in functions Quattro has, I have included a list of all 104 of them in Figure 4.

As you can see, all of them begin with the standard "at-sign" (@) function indicator like 1-2-3. There are more than enough of these functions to satisfy just about any user.

Add-In Programs

If you have ever developed or changed a spreadsheet, perhaps you have been concerned that you will change something incorrectly and be unable to get back to a "correct" version. With Quattro, you get the Transcript add-in program which records every keystroke. If you are working late and make a catastrophic mistake, you can start the Transcript program to restore your spreadsheet values and formulas at any point. Transcript also will help you rebuild a spreadsheet that may not have been saved to disk in the event of a power failure.

If you are concerned about the disk space (who isn't?) that your spreadsheet files occupy, Borland has thoughtfully included the SQZ! PLUS add-in program for com-

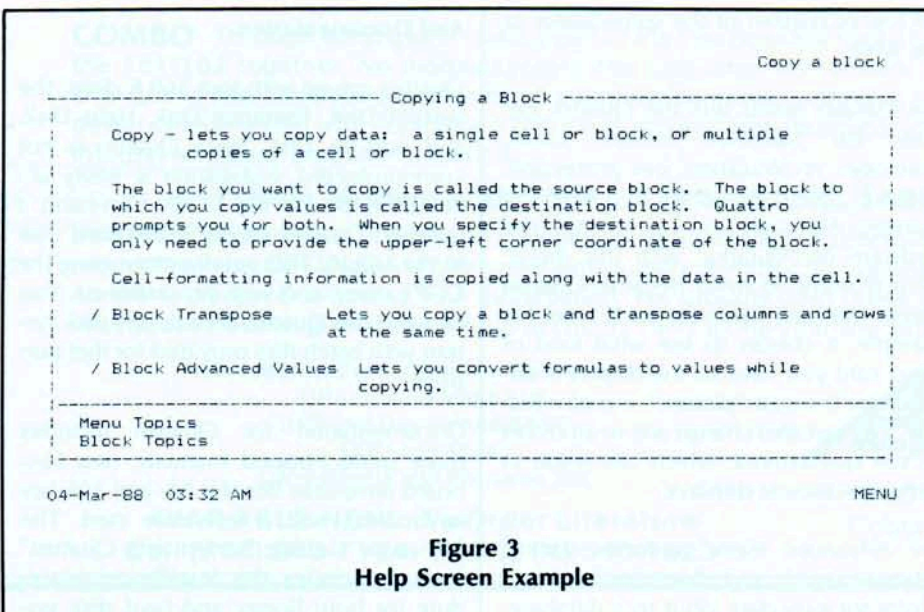


Figure 3
Help Screen Example

pression of large files. If you are not familiar with file compression programs, SQZ! PLUS is a recent upgrade to the SQZ utility from Turner Hall Publishing.

Although Quattro is not compatible with some of the 1-2-3 specific add-on utilities, such as auditing programs, I suspect it is only a matter of time before these programs are upgraded or enhanced to work with Quattro. In the meantime, you can enter simple values (e.g., 1 or 10) into the data cells to easily check most of the common calculations visually without a calculator. That is the easiest way I have found to "audit" a spreadsheet, and I feel more comfortable using that technique because I don't rely on a program to do the checking for me.

Quattro Features And Specifications

Borland uses an interesting approach in advertising this product. You may have seen the advertising slogan: "Quattro. Twice the speed. Twice the power. Half the price." That gives you an idea of Quattro's comparison to 1-2-3. Another slogan is: "Seeing is believing." Well, I have seen, and I believe that Quattro is the best spreadsheet value on the market today.

Quattro runs on PC compatible or PS/2 computers using DOS version 2.00 or later. It requires 384 K of memory, and it may be run on a floppy or hard disk system. I recommend a hard disk for best performance to minimize disk swapping. Quattro can access up to 8192 rows and 256 columns. And it is compatible with 1-2-3 macros and can perform direct reading and writing of those file formats. To minimize learning time, you can also install a 1-2-3 menu structure.

Quattro is not copy-protected, and it is easily installed on a hard disk by using the COPY command. Quattro is fast, and the documentation says that a recalculation of a cash flow model with 5 K cells is performed on 0.27 seconds as compared with Lotus version 2.01 value of 2.90 seconds. This speed improvement is made possible by Quattro's minimal recalculation feature. When you make a change to your spreadsheet, only the affected cells are recalculated — not the entire spreadsheet.

It seems churlish to mention that I did find one disadvantage with Quattro. It requires more memory than 1-2-3, and it may not be possible to load all of a large

@@	@DMAX	@LENGTH	@RIGHT
@ABS	@DMIN	@LN	@ROUND
@ACOS	@DSTD	@LOG	@ROWS
@ASIN	@DSUM	@LOWER	@S
@ATAN2	@DVAR	@MAX	@SECOND
@ATAN	@ERR	@MEMAVAIL	@SIN
@AVG	@EXACT	@MEMEMSAVAIL	@SLN
@CELL	@EXP	@MID	@SQRT
@CELLPOINTER	@FALSE	@MIN	@STD
@CELLINDEX	@FIND	@MINUTE	@STRING
@CHAR	@FV	@MOD	@SUM
@CHOOSE	@FILEEXISTS	@MONTH	@SYD
@CODE	@HLOOKUP	@N	@TAN
@COLS	@HOUR	@NA	@TERM
@COS	@HEXTONUM	@NOW	@TIME
@COUNT	@IF	@NPV	@SYD
@CTERM	@INDEX	@NUMTOHEX	@TAN
@CURVALUE	@INT	@PI	@TERM
@CLEAN	@IRR	@PMT	@TODAY
@DATE	@ISERR	@PROPER	@TRIM
@DATEVALUE	@ISNA	@TIME	@TRUE
@DAVG	@ISNUMBER	@TIMEVALUE	@UPPER
@DAY	@PV	@RAND	@VALUE
@DCOUNT	@RADIANS	@RATE	@VAR
@ddb	@ISSTRING	@REPEAT	@VLOOKUP
@DEGREES	@LEFT	@REPLACE	@YEAR

Figure 4
Quattro Built-In Functions

spreadsheet in conventional memory if you are limited by that. But there is an easy way to correct that problem. Quattro can access up to 4 megabytes of expanded memory (EMS), which should be more than adequate for any user, even if you have a real "monster" spreadsheet. I imagine this capability could be effectively utilized to create some really interesting mathematical models for all kinds of projects. At this point, it seems almost anti-climactic to note that Quattro cannot "link" spreadsheets, but I think this memory capability should be able to cover virtually all requirements.

I believe that Quattro represents the best value in a spreadsheet today. It has outstanding graphics, better speed, and extra features for a more reasonable price than other spreadsheets. If you are familiar with spreadsheets, you will love Quattro. If you aren't, Quattro will give you an introduction to what a spreadsheet should be.

Products Discussed

Quattro (PC Only)	\$247.50
Borland International	
4585 Scotts Valley Drive	
Scotts Valley, CA 95066	
(800) 255-8008 (Except California)	
(800) 742-1133 (California only)	
(800) 237-1136 (Canada only)	



PC WatchWord® from S & K Technology, Inc.

*WatchWord® is the ultimate in word processing for the sophisticated user. Its features include subscripts, superscripts, *l*.*u*.*mat*ting, DOS commands, arithmetic, mail merge, online help, long lines, large files, split screen and macros.*

PC WatchWord® : \$129.95

PC Resident Speller: \$ 99.95

PC Combination

WatchWord® and

The Resident Speller: \$199.95

Z100 WatchWord® : \$100.00

(requires 256K Ram)

Z100 Resident Speller \$100.00

S & K Technology, Inc.
4610 Spotted Oak Woods
San Antonio, Texas 78249
512-492-3384



SEATTLE

HUGCON

ONE GIANT DAY ONLY!!

July 9th, 1988 (10-4 PM PST)
Seattle Center (Lopez Rooms)
(Northwest Rooms)

- **Free Admission!**
- **\$100,000 in H/Z computer equipment at blow-away prices from the Heath Store**
- **Vendor Booths**
- **Speakers**

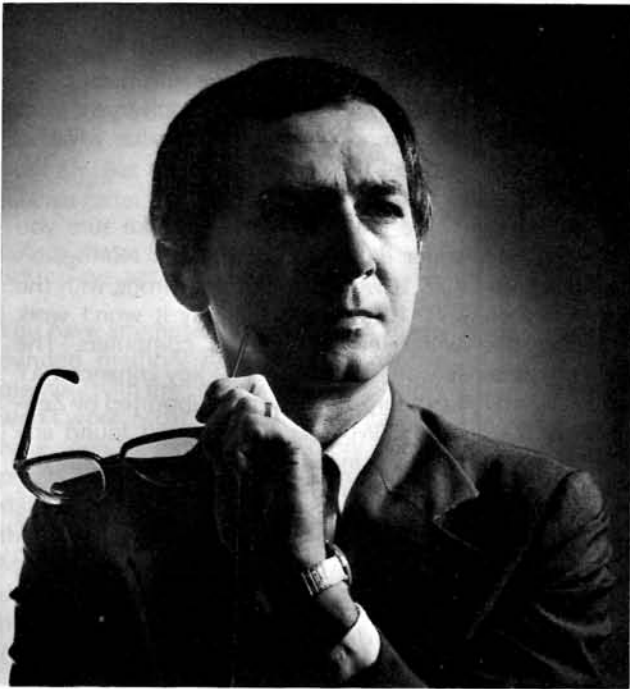
The Seattle Center is within 20 minutes of the SEA-TAC International Airport, and is situated in the heart of all major Hotel/Motel chains. Vendor booths are free, and interested vendors should contact Les Hall at (206) 234-2851 immediately! Don't miss this one!!



Please let us know
3-4 weeks in advance,
so you won't miss a
single issue of
REMark!

MOVING?





Mainstream Computing

Joseph Katz

10 South Edisto Avenue
Columbia, SC 29205

Copyright © 1988 by Joseph Katz. \$400 on publication for First North American Serial Rights only.

My mouse died. Of course, it chose a Saturday afternoon when I was in the middle of a project for which a mouse was essential and most of the local computer stores are closed. To make matters nearly impossible, we'd just had our worst snowfall in years. Side streets still were impassible, while the main streets were still bad enough to make driving risky. Those were my excuses for doing something that I found is foolish to do in these here parts, stranger: I actually telephoned the open stores to see if they had a Logitech C-7 serial mouse in stock. "All you want," said the salesman at a big discount store on the other side of the city. "I can see stacks of them from here," he said. So through the muck and mire I drove. I did it. I really did it, believing all the way that a Logitech C-7 serial mouse would be my reward at the end of the long slide down the highway.

Janet laughed knowingly as I left the house. She knew. She knew.

The reason why it's foolish to phone a salesman in Columbia, South Carolina, with an inquiry if a specific computer part or product is in stock is that the response often has no necessary relevance to objective reality. This store turned out to have not even one Logitech C-7 mouse in stock. In fact, it had only one mouse of

any kind in the house. That singular mouse, moreover, was a Brand X mouse that proclaimed itself as fully compatible with the Logitech, Microsoft, Mouse Systems, and apparently, all other name brand mice currently on the market, ever to be marketed, or even feebly stirring as an idea in the mind of someone somewhere. But I was desperate, the mouse looked and felt reasonably substantial, and the price was dirt cheap. So I bought.

Silly. You'd figure that even if I didn't write this column I'd at least read it enough to know better than to think I could beat the system first try. It served me right: that mouse of uncertain ancestry turned out to be unusable for any real work. Sure, if I hadn't been accustomed to the smooth actions of a good mouse, I might have adapted to the jerkiness of this one. But now I couldn't. There was no way I could get through a long weekend of work with this . . . thing.

Some day you have to meet my friend Dr. William L. Miles. Bill is not only a physician, not only an accomplished dBASE III programmer too, but also the owner of a spare Logitech C-7 serial mouse that he volunteered to lend me until I could get a replacement. No, I didn't ask why anyone would have a brand new mouse to spare as a loaner. (The rule is "Never look a gift

mouse in the mouth.") Yes, Bill really did volunteer the loan. And he drove from his end of town to my house through all that slush and snow to deliver it so I could meet my deadline. That's friendship. Bill's mouse tided me over until Monday, when I ran out and bought a Logitech C-7 bus mouse as a replacement for mine. I have it configured as an LPT port, right now, which frees up a COM port assignment.

When I finished my work, curiosity drove me to do some dissection. Inside the substantial housing of the generic mouse was insubstantial plastic fingers, a few insubstantial bits of metal as switches, and a lightweight rubber tracking ball. Inside the substantial housing of the Logitech C-7 everything was substantial and polished.

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).

My Logitech mouse looked better dead than the generic mouse looked alive.

Of course, it's possible to get real bargains in the computer marketplace, but those, I think, come from companies trying to carve out a niche they intend occupying for a long, long time. They are trying to build a better mousetrap. Boca Research is one of those companies, Zoom Telephonics is another, Cheetah International is a third, and there are others still. It gives me great satisfaction to know that it was I who introduced them here, when they were not generally known, and to see now that their products are recommended to others by users in the Heath and Zenith communities. Add Northgate Computer Systems to the list.

We haven't heard from all such companies yet, because some of them don't exist yet and because I don't claim to know even a small number of such companies that do exist. But that kind of company has a special claim on my interest. I confess to a fascination with them. I don't really know why. It's not simply that they charge less than the going price for similar products. My generic mouse met that one test and was a waste of money: the Logitech C-7 is a name brand that cost more and was worth every penny of the cost. In fact, I spend time each month looking through magazine ads for generic computer products and marvelling about the apparent bargains. A few times I even put together shopping lists for generic parts needed to roll my own complete clones. All those times I saved a bundle of money by abandoning the project each time. I already have enough junk stowed around to build several computers that don't work too well. What I want is stuff that not only works well, but also helps me work better. I'm happy if I can save a few dollars, but I know I can't save a cent with almost-rans.

You and I know how to learn of well-advertised products from giant companies. That's easy. It's the discoveries we both like to make — good stuff at a reasonable price from a good company that is not yet a giant — that are hard. When you find one, do everyone a favor and point the company at me. Then we'll all have some fun.

Let me show you what I mean, with new products from some of the companies I know or am getting to know.

Northgate Computer Systems' C/T Intelligent Keyboards

I spend so much time working with computers that their keyboards do make a big difference to me. Some keyboards are almost unusable, while some are merely unpleasant. I don't find anything really wrong with the keyboards on Zenith's mainstream computers, but they are a bit mushy for my taste. They're passive in the sense that each keystroke feels the same all the way down. They don't give me the kind of feedback — tactile or audible — I need while I'm working. The result is that I'm not as fast or as accurate as I am when working at a responsive keyboard. The only keyboard of any kind I really, really liked was on the IBM Selectric typewriter. I bought my first Selectric more than twenty-five years ago, wore it out writing books and stuff, then bought another. I really liked that keyboard because it was alive: it responded to me, I to it, and we worked together most happily. It's the kind of keyboard I've dreamed of having on my computers. Now I've found it, and it costs only \$79.

The computer keyboard of my dreams so far is Northgate's C/T 84. "C/T" stands for "Click/Tactile," the two kinds of responses for which I've been looking in a computer keyboard. The click is mechanical, not an electronic simulation, so it sounds right. Because it's mechanical, it also feels right. Each key travels downward smoothly to a detent that marks its contact point, then continues smoothly to the bottom of the switch. With other keyboards I find myself banging away to make sure each key is fully depressed. With the Northgate C/T 84, I find myself touching the key just to the point of contact marked by the detent, which I feel and hear. My touch therefore is lighter, faster, more accurate. The result is that I am more productive.

Northgate calls the C/T 84 an "intelligent keyboard," perhaps because this one can be used on both XT and AT compatible computers. Recessed underneath the keyboard is a slide switch marked "A" and "X": slide the switch to "A" if your computer is an AT-compatible, such as a Z-248, to "X" if it's an XT-compatible, such as a Z-158. Here are all the other instructions you need to install Northgate's keyboard on one of those Zenith computers: power down the computer, unplug Zenith's keyboard, plug in the C/T 84, and power up the computer.

What I'm considering seriously is getting another Northgate C/T 84 to carry with me when I have to work with other computers. I am sick and tired of having to rediscover keys, such as the backspace, escape, and backslash when I move from one kind of computer to another. My idea is a reasonable solution.

Three things more. First, make sure you order the C/T 84 and not Northgate's C/T 101. There's nothing wrong with the C/T 101 keyboard, only it won't work with Heath or Zenith computers. The reason is that it is a 101 key Enhanced AT keyboard, which is not supported by Zenith's ROM. Second, I haven't found any software incompatibilities at all with the Northgate C/T 84 keyboard. It works like a charm with everything. Third, Northgate will allow you to return one of its keyboards within ten days if you don't like it or even if you just feel like being ornery. There are procedures to follow, but nothing onerous. Northgate obviously believes you'll fall in love at first touch. So do I.

If you're interested in an inexpensive, easy upgrade that will improve your computer system dramatically, get this keyboard. Bill Adney and I have burned up the long distance lines talking about it. See for yourself why we're both so excited.

Boca Research's EGA And MultiEGA Boards

Boca Research is the company in Boca Raton, Florida, that produces good, solid add-on boards for IBM compatible computers, provides good support for its products, and charges a low price for them. It's usually possible to buy some Brand X near-equivalents at lower prices, but when price is your major criterion, you risk getting near or total misses instead. In my experience, Boca Research's products are always good values, and many of my friends among computer dealers agree at least privately. So when a Boca Research product arrives here, I'm always interested in trying it out. This time I have two to mention.

If you're looking for a standard EGA board to support a standard EGA monitor, buy EGA by Boca. Its list price is \$199 and it works well. Nope, I'm not damning EGA by Boca with faint praise. EGA boards are now common. What you want is a quality product at a reasonable price: that's EGA by Boca. I think it would be foolish to risk

saving a few dollars on a Brand X version of the standard EGA board when Boca offers this one.

If you have an NEC Multisync or other multi-frequency EGA monitor, however, you absolutely must think instead in terms of MultiEGA by Boca. This board at \$299 is a little jewel. It has on-board a 24 MHz crystal (a jumper lets you disable it in case you want to plug an expansion module, with its own external oscillator, into the feature connector), 256 KB of RAM, and a jumper to select either autoswitching or manual switching.

MultiEGA by Boca, therefore, takes good advantage of your multi-frequency monitor's ability to do various useful tricks — the very reasons, presumably, for which you bought a monitor like the NEC Multisync in the first place. For example, because it supports video modes 0-7 and D-37, the MultiEGA allows you to run different application software packages in the video environment best suited to each. Many people like working with 43 lines of text (instead of 24) on the screen when they do word processing: you can do it with an EGA display, if your software can be installed to support it. WordStar 4.0 can be, but WordStar 3.0 can't. MultiEGA comes with software that includes a patch for WordStar 3.0.

In fact, EGA utility software is one important advantage offered by Boca. As I had about finished exploring the 1.0 software supplied with MultiEGA by Boca, Boca Research sent me its Version 1.1 software with additional support for Lotus 1-2-3 (Release 2.0 or 2.1) and Symphony (Release 1.1 or 1.2), to give a display of 132 columns by 43 lines. If you already own MultiEGA and got the 1.0 software, write Boca Research for the 1.1 version and you'll get it free.

I confess it took some time for me to stop playing with another set of programs supplied in the MultiEGA software. They allow one to design custom EGA display fonts, and I find them addictive. Because those programs work by allowing you to turn pixels on or off on a simple grid that represents the character box, they don't require art talent great or small. I am using them to work out *The Perfect EGA Font*. You'll probably find a better use for them.

Zoom Telephonics' HC 2400 Internal Modem

The company that made my favorite internal modem, the Zoom/Modem PC

2400 XL with all the bells and whistles, has introduced a standard 300/1200/2400 Baud internal modem for those who don't need all the bells and whistles.

But it's as if Zoom Telephonics has an irresistible urge to provide more than it has to, even when it seems to be saying it won't. This basic modem isn't really a stripped down modem. Just like its more advanced siblings, the HC 2400 supports four COM ports instead of the usual two; just like its more advanced siblings, it has a high speed UART for reliable use in 80286 and 80386 computers; just like its more advanced siblings, the HC 2400 has an integral speaker with volume control; just like its more advanced siblings, the HC 2400 has an audio output jack so you can plug in your own external speaker if you want better sound than you can get from the integral speaker; just like its more advanced siblings, the HC 2400 has a modular jack so you can plug in your phone; just like its more advanced siblings, the HC 2400 gives you dialtone detection and auto-dial/auto-answer capabilities; just like its more advanced siblings, the HC 2400 is completely Hayes compatible; and just like its more advanced siblings, the HC 2400 comes with

communications software. The software is ProComm, the shareware program so many of you like so well. The price for all this is \$199.

See you later.

Products

Northgate C/T 84 Intelligent Keyboard	\$ 79
Northgate Computer Systems Suite 250, 2905 Northwest Boulevard Plymouth, MN 55441 (800) 328-8907	
EGA by Boca	\$199
MultiEGA by Boca	\$299
Boca Research Inc. Boca Raton, FL 33487 (305) 997-6227	
Zoom/Modem HC-2400	\$199
207 South Street Boston, MA 02111 (617) 423-1072	

*

PowerStation™

A Complete VT220 / VT241 Work Station Upgrade for Zenith PC's, AT's and Compatibles



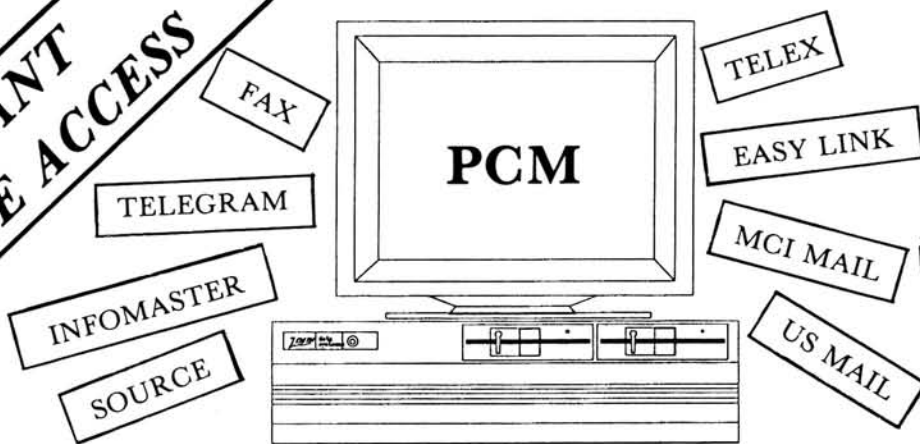
"Now true 800 pixel wide VT241 display and support for VMS Services for MS-DOS."

<p>PowerStation™ 240 VT240 style keyboard and ZSTEM VT240 Emulation Software. Optional WPS labelled keys (GOLD KEY MODEL) add \$30.</p> <p>ZSTEM pc™ - VT240 Emulator Emulation Software only. VT240/241 Emulation Software with all the features of ZSTEM VT220 plus ZSTEM 4014, sixel and ReGIS graphics.</p> <p>PowerStation™ 220 VT220 style keyboard and ZSTEM VT220 Emulation Software. Optional WPS labelled keys (GOLD KEY MODEL) add \$30.</p> <p>ZSTEMpc™-VT220 Emulator Emulation Software only. All the features of ZSTEM VT100 plus 8-bit mode, downloadable fonts, user defined keys, full national/multi-national modes. Extended macros/script language. True 132 columns on Hercules, VGAs, Super EGAs, and standard EGAs using the EGAmate option. 128 columns on CGAs. 43 line support on EGAs. Enhanced keyboard support. Ungermann Bass Net/One and VMS Services for MS-DOS support.</p>	<p>\$435</p> <p>\$295</p> <p>\$289</p> <p>\$150</p>	<p>EGAmate™ \$39 Daughterboard option for 132 columns and true 800 pixel wide ReGIS display on standard EGA adapters.</p> <p>PS220/2 \$19 Keyboard adapter cable for PS200 on PS/2 systems.</p> <p>ZSTEMpc™-4014 Emulator \$99 Use with ZSTEM VT100, VT220, or stand-alone. Interactive zoom and pan. Save/recall images from disk. Keypad, mouse, digitizer, printer, plotter, and TIFF support. 4100 color and line style color mapping. 640 x 400 and 640 x 480 on some adapter/monitors.</p> <p>ZSTEMpc™-VT100 Emulator \$99 High performance COLOR VT100. True double high/width, smooth scrolling. ISO and attribute mapped color. XMODEM and KERMIT, softkey/MACROS, DOS access.</p>
---	---	---

KEA Systems Ltd.
#412 - 2150 West Broadway, Vancouver, B.C. Canada V6K 4L9
Telephone (604) 732-7411 Telex 04-352848 VCR Fax (604) 732-0715
Order Toll Free (800) 663-8702
30 day money back guarantee AMEX/MC/VISA

SEE US AT
COMDEX
#2044

**INSTANT
ONLINE ACCESS**



NEED A HAND?

Imagine A Package With Communications, Word Processing, Mail Merge, Electronic Mail, Database, Desktop Management, On-Line Utilities And More.
PROFESSIONAL COMMUNICATIONS MANAGER HAS IT ALL

Features XMODEM Binary Transfer

Major Features:

- Electronic Mail Systems—drives all major E Mail services.
- On-Line Systems—drives all major electronic database systems.
- Administrative Assistant—extensive, dynamic personal database.
- Hot Wire—quick access to virtually everything!

Additional Features

- 411—our E-Mail "Information Please" finds subscriber addresses.
- Special Delivery—E-Mail auto-retrieval and down-load system.
- Pre-View—presents your letter for review of contents and appearance before transmission.
- Econo-Miser—quickly down-loads and economically captures large blocks of on-line database data for off-line review and editing.
- C.C.—"courtesy copy to" feature works with all E-Mail services.
- Protect—prevents unauthorized use.
- CONNEX port—for text transfer among micros, minis and mainframes.
- 611—an on-line technical trouble shooter, over and above our normal phone support.

- Cross-Connect—cross-references everything in your database.
- Echo-Check—proprietary error correction system that absorbs external glitches without losing data.
- CONNEXMODEM—uses XMODEM for error-free data transmission of binary files.
- Auto-insert—takes information from your database and automatically puts it in a letter with no intermediate steps.

Full Support For:

- Mail Services—AT&T Mail, EasyLink, Zipnet, Network, MCI Mail.
- All Mail Types—E-Mail, U.S. Mail, Telex, Telegram, Mailgram, Cablegram, instant message, overnight, 4-hour delivery, C.O.D., paper letter with reply, and Fax.
- On-Line Databases—CONNEX connects to over 600 in all. Everything on The Source, CompuServe, DIALOG, InfoMaster, Dialcom, GENie, Delphi and more.
- Word Processing—works with all the popular and most of the obscure packages including WordStar, Word Perfect, Microsoft Word, DisplayWrite 3, PC-Write, XyWrite and many more.
- Modems—perfect with Hayes and compatibles, Racal-Vadic, Prometheus, Novation, Kyocera, SmarTEAM, Capetronics, etc.

Connex is Not Copy Protected

With **PCM** by **CONNEX** anyone can send out cables, telexes, telegrams and electronic paper mail in top letter quality form. Also sends FAX.

So sit by your IBM PC or compatible, pull a tidbit from The Source, add it into your client letter, draw added facts from your database, then preview the results—complete with logo and signature—before it's sent. Forget the address? Get it from your contact file or use our built-in E-Mail directory assistance.

PCM by **CONNEX** combines the benefits of a dozen programs into an affordable, integrated package.

I'M READY FOR PCM! Send the 5-diskette package with 172-page manual to my attention at once. Limited time price of \$199.95 includes shipping, handling, tax, and one year of upgrades!

I need to see a demonstration! Send demo-disk with \$50.00 coupon for only \$5.00.

PCM is for IBM PC, AT, PS/2 and compatibles, hard disk required.
Mail to: **DESIGNLINE, P.O. 1198, Bethesda, MD 20817**
call: **800-USA-3360** or **800-372-6365**.

Name _____
Company/Agency _____
Address _____
City _____ State _____ Zip _____
Telephone (____) _____ (ext.) _____

REGISTERED TRADEMARKS IBM PC, AT, PS/2, DisplayWrite 3/International Business Machines Corp. • TheSource Telecomputing Corp. AT-T Mail/American Telephone and Telegraph, Inc. • EasyLink, InfoMaster/Western Union Telegraph Company • Zipnet Business Corporation of America Network/Network, Inc. • MCI Mail/MCI Communications Corp. • CompuServe/CompuServe, Inc. • DIALOG/DIALOG Information Services, Inc. • WordStar/MicroPro Intl Corp. • Microsoft Word/Microsoft Word/Microsoft Corp. • PC-Write Quicksoft • Hayes/Hayes Microcomputer Products • Racal-Vadic/Racal-Vadic Capetronics/Capetronic Computer Peripherals • DIALCOM/Dialcom, Inc. • GENie/General Electric Information Services Co. • Delphi/General Videotext, Inc. • Prometheus/Prometheus Products, Inc. • Novation/Novation, Inc. • Kyocera/Kyocera International Inc. • SmarTEAM/SmarTEAM, Inc. • XyWrite/XyQuest, Inc.

Getting Started With . . . Microsoft® Word

Jack W. Bazhaw

900 - 13 Street
Bellingham, WA 98225

© Copyright Jack W. Bazhaw, 1988. All rights reserved.

Let's face it, Microsoft Word is different. From its graphics display and mouse interface to its use of stylesheets, Word thinks about a document differently than most other word processors.

One might compare these differences to the difference between a standard or algebraic calculator and a Hewlett-Packard calculator that uses reverse Polish notation or RPN. At first, an RPN calculator seems strange, until after enough experience to reach the "ah ha" moment, when a new world of working with complicated calculations with ease opens up. So be prepared to open up to some new thinking about how you work with a document when you first jump into Word.

To avoid bloodshed, let's first cover a point about word processors in general. Many experience the same emotional attachment to their word processor as they do their spouse; perhaps they used common criteria for the selection. The capability of Word to easily manipulate the printer has been a strong reason for me to continue with the program since starting with Version 1.1. I confess to having never used WordStar, but that does not mean you cannot be happy with WordStar (or WordPerfect, MultiMate, etc.).

My experience has been that most of us have not had much of a problem learning

how to manipulate text with Word. The problems start when we want to format the text so it looks nice or meets certain criteria. The subtitle for this piece should be: Getting Started With Document Formatting With Microsoft Word StyleSheets.

What we will cover applies to Version 4.0, but can be used with previous versions, except for style by example, perhaps back to Version 2.

As we go through some examples, I'll use the same conventions the Word 4.0 manual does to represent certain keystrokes.

Italics indicates words and characters you will type on the keyboard. Capitals indicate keynames, such as ENTER, SHIFT, ESCAPE.

My machine has a fixed disk (or hard drive) and you may run into some situations using only floppies that require you to swap disks. If you do, just follow the screen prompt. So that we have a common starting point, before starting Word, change to the directory (or disk) where you have your Word files located and type:

```
dir normal.sty
```

If DOS shows you have such a file, re-name it by typing:

```
ren normal.sty normal$.sty
```

This will prevent any formatting you may already have in this stylesheet from keeping you from seeing the same thing on screen that I plan to describe.

Next, start up Word [while still in the directory (or disk) where the Word files are located] by typing:

```
word
```

The screen Word will present to you can be customized to a certain extent. To see more of the document, turn off the menu at the bottom. To do this, press ESCAPE, then O for Options. Use the RIGHT key (TAB prior to 4.0) to go to "menu", then press N and RETURN to turn the menu off. To bring back the menu at any time, just press ESCAPE.

Also on the Option menu is a selection called "visible". I find it very helpful to select "partial", as this allows the paragraph (¶) and new-line (↵) marks to show on screen. Because Word treats paragraphs as a unit, it is handy to know where the paragraph mark is located.

With Version 4.0, you can also turn the screen border characters off. However, you pay a price for the few extra characters you can see on screen — the mouse

cannot be used to open windows or scroll.

There are two other screen options not accessed from the Options menu, but from the Window Option menu. Press ESCAPE W O to bring up this menu. Later on, we will be working with stylesheets so turn on the style bar. The ruler at the top can be turned on or off as you see fit.

Such page oriented things as top and bottom margins, left and right margins and page numbering are controlled with Word as division formatting. If you make no specific assignments, Word will start you with left and right margins of 1.25 inches, top and bottom margins of 1 inch with a page 11 inches long and 8.5 inches wide. Certain printers, such as the Epson RX80 or the HP LaserJet series can use paper 8.5 inches wide, but only print an 8 inch line. For those types of printers, the paper width shown under Format Division Margins should be changed to 8 inches (press ESCAPE F D M, TAB 5 times, enter 8 for page width, press RETURN). Unless you use a stylesheet, any changes you make will effect only the current document.

Running heads, or "header" and "footer" are set to print .5 inch from the top or bottom. You may have as many lines as needed in a running head, but if they would print outside the allocated space and overlap the main text, Word will not print the running head until you change the margins to give enough room.

Assuming you have not pressed any keys, other than those for the menu selections, you should have on screen a small diamond mark with the cursor setting on top of it and an asterisk (*) to the left, over in the style bar. Press RETURN three times and you should see three paragraph marks (¶) appear, one above the other and each should have an asterisk alongside in the style bar. The asterisk means that you have not selected a style for the paragraph. Patience, we will get there later.

Word is normally in the insert mode rather than overtype, so typing errors can be corrected by pressing BACKSPACE to delete the character behind the cursor or press DEL to remove the character under the cursor. The diamond mark (called the end-mark) cannot be deleted or formatted.

Use the UP key to return to the middle paragraph mark. Above and below this

mark should be another one with no blank space between them. In other word processors, this is how you create blank space between paragraphs, by pressing RETURN. Word, however, stores information in the mark (¶) that tells it how much space to put between paragraphs. From the menu, select the Format Paragraph menu (press ESCAPE F P). Note the selections for "space before:" and "space after:". TAB to the "space after" area, type a 1 and press RETURN. You have told Word to follow the middle paragraph with one line of space. Now you should have something like this on screen:

¶
¶
¶

Well, you say, "Big deal, I could do that a lot easier with the RETURN key." Hold on. Now press SHIFT+F10 (the function key) to select the entire document, such as it is. Now go to the Format Paragraph menu and put in a 2 for "space after". Your paragraphs will now be spaced two lines apart. Imagine you had an entire document you used RETURN to put a blank line after each paragraph and then decided to change it. Somehow you will have to find a way to remove all the unwanted marks or add to them. The more complicated the document formatting, the more difficult or time-consuming it can become to make the change. Using the Format Paragraph command instead of the RETURN key will speed the process. And throw in style-sheets and such changes become even easier.

Press SHIFT+F10 to select everything and then press DEL. Again, we have a blank (for Word, anyway) screen. Type in enough text to give you two or three lines, then press RETURN. Press SHIFT+F10 again, press DEL again, then press INS three times. The same text should now appear in three different paragraphs, single-spaced with no spacing between paragraphs. You can tell they are separate paragraphs by the paragraph mark.

With the entire document still highlighted (press SHIFT+F10 if it is not), select Format Paragraph from the menu. For "alignment", enter L for left; for "left indent" and "right indent" enter 0, enter .5 for "first line", enter a 1 for "space after" and "line spacing", then press RETURN. Each paragraph will have the first line indented ½ inch with a blank line following.

Now repeat the preceding paragraph, but use 0 for "first line" and 2 for "line spacing". You should now have block paragraphs that are double-spaced. The idea behind using the Format Paragraph command to change the appearance of the paragraphs rather than the TAB and RETURN keys is it is much simpler to make document-wide changes, particularly when a stylesheet is used. Like learning an RPN calculator, it takes a little more effort up front, but the payoff comes as the job gets tougher.

Using the paragraph mark to store formatting information means strange things may happen when a mark is deleted if you don't plan ahead. Here is an example to demonstrate. You should still have on screen the three paragraphs of text that are double-spaced. Move the cursor into the middle paragraph. Select Format Paragraph from the menu, change the first line indent to .5 inch and the line spacing to 1. This makes the middle paragraph different than the first and last which are block and double-spaced. Put the cursor on the paragraph mark that ends the middle paragraph and press DEL.

Bingo. The middle paragraph is now blocked and double-spaced and joined with the last paragraph. Press RETURN to restore a paragraph mark. You will have three paragraphs again, but they still look the same. When you deleted the mark, you deleted all the formatting information about the middle paragraph and it assumed the formatting of the next mark. Press BACKSPACE (don't use DEL) to delete the mark you just typed, then press INS to re-insert the deleted mark which was placed in the scrap (and still is there, in fact) by the DEL key.

With the formatting information restored, the middle paragraph is again indented and single-spaced. It is easy in editing without the visible option set to partial to delete a paragraph mark and experience sudden formatting changes. Recovery is a matter of getting back the missing mark by using the scrap contents or the Undo command.

Now let's look at a special kind of paragraph, running heads, or "header" and "footer". A common use for running heads is to print chapter titles, revision numbers, page numbers and the like on each page. For running heads to print on a page, they must be the first paragraph(s) on the page. To tell Word a paragraph is to be a running head, place the cursor in

the paragraph and press ESCAPE F R for the Format Running Head menu. Select "top" for a header or "bottom" for a footer. The next three choices determine which pages (odd, even, first) the running head will print on and are used to tell Word to no longer treat the paragraph as a running head by setting all three choices to "no".

The style bar will display a one or two letter code to tell you where and on what pages a running head will print.

Running head paragraphs do not share the same margin settings as normal paragraphs. To have a running head print with the same margin setting as the rest of the paragraphs, select the running head, press ESCAPE F P for the Format Paragraph menu and set a left and right indent equal to your left and right margins. The default settings of 0 will print your running heads at the paper edge.

Here is a representative running head and how it is formatted. We want a footer with the chapter title flush with the left margin, the page number flush with the right margin and these two elements connected with a thin solid line. When printed, the footer should look like this:

Chapter 1 _____ Page 10

To create this footer, go to the beginning of the document (CTRL PGUP) and type:

Chapter 1 TAB Page page F3 RETURN

Pressing F3 following the second typing of the word "page" will insert the Word glossary entry to give you automatic page numbering [it will show on screen as "Page (page)"]. Now move your cursor back into the running head paragraph and press ESCAPE F P for the Format Paragraph menu. Press L for left alignment, and enter your left and right margin settings for the left and right indent entries (make sure "first line" is 0), then press RETURN. Make it a running head with ESCAPE F R, selecting where you want it to print (at least one choice must be yes or it will not be a running head), press RETURN, then bring up the tab menu with ALT+F1 (ESCAPE F T prior to 4.0). For the tab measurement, type in your right margin setting. For alignment select right and select the underscore () for the leader character. Press RETURN and you should see (if "partial" is set in Options Visible):

Chapter 1 _____ Page (page)

If the page number does not print (on paper, it will not show on screen), make sure you have "yes" shown in the Format Division Page Numbers menu.

The glossary entry "page" we used in the running head is one of several that are pre-defined by Word. The others are footnote, date, dateprint, time and timeprint. (Plus clipboard which is used only when running Word under Windows.) The glossary is a place to store frequently used text, such as your senator's address or contract boilerplate.

Each entry is stored under a name of your choosing. The names of all current glossary entries can be seen by pressing ESCAPE I, then F1 (RIGHT prior to 4.0). If you have made no entries, the only ones you will see will be the ones pre-defined by Word; they cannot be deleted.

The next formatting element Word uses is "character". Not the Boy Scout kind, but the marks you put on paper with your printer. Character formatting is the most printer-dependent feature in Word. If you are using a laser printer or one of the newer 24-pin dot matrix printers, you may have many choices on how to format the printed character. Something like an MX-80 will be much more limited.

It is very important that you have the proper printer driver installed. Press ESCAPE P O to bring up the Print Options menu and check the name of the print driver shown against what the Word manual says should be used with your printer. If the correct one is not showing, press F1 (RIGHT prior to 4.0) for the list of drivers, highlight the correct one and press RETURN. If the correct driver is not shown on the list, you will need to check your printer disks and copy the file to your working disk or directory.

With the correct driver installed, the list of available fonts can be seen by pressing ALT+F8 and then F1. Prior to 4.0, press ESCAPE F C, then TAB to "font name" and press RIGHT.

Highlight the desired font name, press TAB, then F1 (RIGHT prior to 4.0) to see the list of sizes available in that typeface. Highlight the desired size, press RETURN and that font will be used for all new text.

Font name is the name by which the typeface is known. For example, with the driv-

er EPSONFXE loaded, press ALT+F8 and then F1. Six font names will be listed — Pica, PicaD, Elite, EliteD, NLQ and PS. With Pica highlighted, press TAB, then F1 and the four available sizes for Pica, 8, 12, 14 and 16 points, will show in the list.

From the same menu that you select font name and size, you can also select other character formatting, such as italic or bold. However, there is a shortcut using the ALT key and certain "speed formatting" keys. To underline a word with the menu, after selecting the word, requires several keystrokes: ESCAPE F C TAB TAB Y. With the speed keys, just press: ALT U. Once you attach a stylesheet to your document, use ALT X instead of ALT. Our sample would become ALT XU when the stylesheet is attached.

If you are going back to text you previously typed to add character formatting, such as underlining, be aware of how Word reacts when only a single character is selected. When you have two or more characters selected (highlighted) Word assumes you are ready to apply whatever style you enter immediately to those characters. With only one character selected, Word is not sure if you want to start typing with the format you selected or to just apply it to the single character. You solve Word's confusion by entering the formatting keystrokes twice.

Formatting with the menu or speed keys overrides any formatting applied by the stylesheet. I believe it is best to use the stylesheet to apply the character formatting to the entire paragraph and use the menu or speed keys only for isolated words or letters within the paragraph.

Let's try something with a stylesheet.

To use a stylesheet we first must attach it to the document. Select Format Stylesheet Attach (Format Stylesheet for 3.0 and 3.1) and type in *playwith*, press RETURN and answer Y when asked to create the stylesheet. Since we have no styles defined, nothing has happened to our document except we have set aside an area to record how we want the document to look.

Now let's go to the blank stylesheet and create our first style. We will first name the style and then select the characteristics that apply to the style.

To name the style, select the Gallery (ESCAPE G) menu and you will see a blank

screen. Notice that the file name displayed in the lower right is now PLAY-WITH.STY and the menu shows "GALLERY" in the lower left corner. Now, to insert our first style, type *I* for Insert. Word is now asking for the two-letter key code we will use to refer to this style. Type *GT* for general text. You may select any two characters you desire, but avoid starting with an "X" or you will interfere with the use of the speed formatting keys. Then TAB to "usage:" and type *P* for paragraph, then TAB to "variant:". Press F1 (RIGHT for 3.0 and 3.1) and you will see the list of what Word calls the variant. Highlight the word "Standard", press TAB, type in *General Text* under "remarks:" and press RETURN. By selecting "Standard" for the variant, we have told Word to use this style for the document's paragraphs unless we select another.

Now to select the characteristics that apply to the style, select *Format Paragraph*. For "alignment", enter *L*; for "left indent" and "right indent", enter *0*, enter *.5* for "first line", enter a *1* for "space after" and press RETURN.

Normally, you would also select the character style (*Format Character*) for your standard (or normal) paragraph. I am using a LaserJet with Softcraft downloaded fonts and have selected TimesRoman in 12 point size. For a printer like an Epson FX85 you might select Pica in 12 point.

The first style has been completed so *Exit the Gallery* (press *E*).

Your document will now return to the screen with a few changes. The style bar should have "GT" in place of the asterisk with each paragraph having the first line indented 1/2 inch with a blank line following.

Why did Word change the asterisk to "GT" and apply the style? Because when we selected "Standard" as the variant, we told Word this is the default paragraph format for normal text and Word will use it for all paragraphs until told otherwise.

Go back to the Gallery (ESCAPE *G*). We're going to make another new style, so type *C* then *INS* to copy the existing style to scrap and insert it back into the stylesheet. Right now you have two identical styles on screen with the new one highlighted. Type *N* to name the new style and enter *BP* for the key code. For variant, type *1*. Under remarks, enter *block paragraph* and press RETURN. Then, *Format*

Paragraph and change "first line" to 0 (left and right indent should remain at 0). *Exit the Gallery*.

Back at our test document, place the highlight in the first paragraph. Press ALT *BP*. The style bar will show "BP" alongside the first paragraph and the first line will no longer be indented.

By now you are an expert on creating styles in your stylesheet and applying them to the document. But it is still a lot of work — can it be simplified? There are two ways.

Word was furnished with several sample stylesheets. It is easier and faster to modify an existing stylesheet than to create one from scratch. There is at least one book available with a disk of several sample stylesheets (Microsoft Word Style Sheets, by Rinearson and Woodcock).

Word 4.0 users have a second option for creating styles, called style by example. Instead of going to the stylesheet, creating the style and then applying it to the document, the process is reversed. While in the document you format the material as you desire, seeing the effects on screen as they occur, then you record the style with ALT+F10 (or ESCAPE *Format Style-sheet Record*).

Recall that when selecting the "variant" for our first paragraph style we used "standard" which was the first from a long list. The selection "footnote" would attach to your footnotes (once the style is defined) just as "standard" applied to the normal paragraph. Note that once you have defined a style, the key code will appear in parentheses after the variant to show which variants have already been assigned.

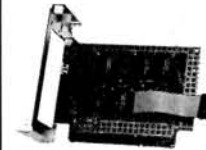
Word is a powerful program and the stylesheet is one of its power features. Hopefully, some of the mystery has been removed and you will find it easier using this power.

Oh, yes, don't forget to rename your stylesheet, if you did so at the start, by typing:

ren normal\$.sty normal.sty



HEATH/ZENITH 88, 89, 90 PERIPHERALS

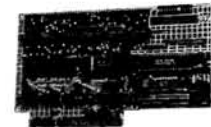
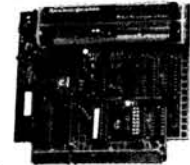


**16K RAM
EXPANSION
CARD**

Only \$65.00
Shipping &
Handling \$5.00

REAL TIME CLOCK

Price \$130.00 with
Batteries
Shipping &
Handling \$5.00
\$114.00 without
Batteries



**2 PORT
SERIAL
3 PORT
PARALLEL
I / O CARD**

Price \$199.00
Second Operating System Driver \$25.00
Shipping & Handling \$10.00



PRICES ARE LESS SHIPPING & TAX
IF RESIDENT OF CALIFORNIA
MAIL ORDER: 12011 ACLARE STREET,
CERRITOS, CA 90701
(213) 924-6741
TECHNICAL INFO / HELP:
8575 KNOTT AVENUE, SUITE D
BUENA PARK, CA 90620
(714) 952-3930

TERMS & SPECIFICATIONS SUBJECT TO CHANGE WITHOUT NOTICE
VISA & MASTER CARD GLADLY ACCEPTED

G1: COMM
 Execute View-Options Sort-Range Column Row 1=Primary key 2=Secondary key
 Perform sort and return to READY mode

	A	B	C	D	E	F	G	H
1	ENABLE's Sort							
2								
3	FFF	234		THESE ARE LINES				
4	HHH	5645		MORE LINES-				
5	WWW	123		DIFFERENT LINES				
6	NNN	67		THESE ARE IN THE SORT				
7	SSS	87685		EXAMPLE OF SORT				
8	YYY	1234		ENABLE'S SORT				
9	DDD	756		BEFORE SORT				
10	OOO	3246		DIFFERENT LINES				
11	WWW	2345		ALL THESE WILL BE SORTED				
12	III	8765		LINES				
13	DDD	2345		TEXT				
14	AAA	1243		LINES AND TEXT				
15								
16								
17								
18								
19								

Figure 2
SORT MENU

errors can occur or you may want to change a part. These two activities can cause problems as was pointed out when a person misquoted a bid because he forgot to change a range. With the ENABLE spreadsheet, you can HILITE all cells that are affected by the selected cell. To use this capability, press (F10) or SLASH to display the top line menu. Select (W)orksheet and (H)ILITE. You can then move around the spreadsheet with the cursor

keys to the cell you want to check. Press <RETURN> and all cells that would be affected by the cell will be highlighted. Displayed on the 25th line (STATUS LINE) will be the total number of cells that are affect by the selected cell. You can continue to move around the spreadsheet to check on cells that are outside the visible range until you press (ESC). You can not select another cell until you leave the option, which clears the choice, and return

again.

ENABLE's spreadsheet permits you to split the screen either horizontally or vertically. This feature allows you to view parts of the spreadsheet that would normally be off the screen. To invoke this capability, move the cursor to the cell where you want the split to occur. If you plan on splitting horizontally, the cursor must be on the cell that will be the top cell of the

G1: COMM
 Execute View-Options Sort-Range Column Ro

	A	B	C	D	E
1	ENABLE's Sort				
2					
3	FFF	234		THESE ARE LINES	
4	HHH	5645		MORE LINES-	
5	WWW	123		DIFFERENT LINES	
6	NNN	67		THESE ARE IN TH	
7	SSS	87685		EXAMPLE OF SORT	
8	YYY	1234		ENABLE'S SORT	
9	DDD	756		BEFORE SORT	
10	OOO	3246		DIFFERENT LINES	
11	WWW	2345		ALL THESE WILL	
12	III	8765		LINES	
13	DDD	2345		TEXT	
14	AAA	1243		LINES AND TEXT	
15					
16					
17					
18					
19					

```

Range to sort : A3..D14
Primary key   : A3..A14
Secondary key : B3..B14
Primary is    : ASCENDING
Secondary is  : ASCENDING
Order for SORT : COLUMN
  
```

20SCREEN DISPLAY OF HILITE FUNCTION IN ENABL
 #1 SM009 Use up/down cursor keys to view list; use any other key to exit

Figure 3
View-Options Window


```

G1:                                     READY
      A          B          C          D          E          F          G          H
1  ENABLE's Sort
2
3  FFF          234          THESE ARE LINES
4  HHH          5645         MORE LINES-
5  WWW          123          DIFFERENT LINES
6  NNN          67           THESE ARE IN THE SORT
7  SSS          87685        EXAMPLE OF SORT
8  YYY          1234         ENABLE'S SORT
9  DDD          756          BEFORE SORT
10 OOO          3246         DIFFERENT LINES
11 WWW          2345         ALL THESE WILL BE SORTED
12 III          8765         LINES
13 DDD          2345         TEXT
14 AAA          1243         LINES AND TEXT
15
16
17
18
19
20 SCREEN DISPLAY OF HILITE FUNCTION IN ENABLE
#1  A:\HUGSIX.SSF                               Cap    W52

```

sired command with first letters. (F9) is the expert command key in all modules in ENABLE. Function key (F6) will permit you to move between the two windows on the screen. Pressing F6 will immediately move you to the other window. If you do something in either window, it will show up in both if the cells are displayed. Splitting the screen is similar to using the WORKSHEET TITLE option. In the TITLE option, the top or left section of the screen is locked and scrolling is accomplished in the other section. TITLE will not display the two windows with borders while both windows have full top and side borders displayed in SPLIT.

Use of the SPLIT screen is done through the MCM or Master Control Module. The other MCM options available are: The Window, Other Window, File, and Screen. The SPLIT screen is done through (T)his Screen. This screen handles the function of what its label says, THIS SCREEN. The Close option does what it says, closes the current window. One function of ENABLE that has not been discussed in detail in this series is window size and location. This capability exists in all modules of ENABLE and makes it very handy to use. Modify Size or Location

```

G1:                                     READY
      A          B          C          D          E          F          G          H
1  ENABLE's Sort
2
3  AAA          1243         LINES AND TEXT
4  DDD          756          BEFORE SORT
5  DDD          2345         TEXT
6  FFF          234          THESE ARE LINES
7  HHH          5645         MORE LINES-
8  III          8765         LINES
9  NNN          67           THESE ARE IN THE SORT
10 OOO          3246         DIFFERENT LINES
11 SSS          87685        EXAMPLE OF SORT
12 WWW          123          DIFFERENT LINES
13 WWW          2345         ALL THESE WILL BE SORTED
14 YYY          1234         ENABLE'S SORT
15
16
17
18
19

```

Figure 4
Before And After Sort Display

bottom display. If you plan to split vertically, this will occur to the left of the cursor. The cells below/to the left of the cursor will be fixed, while those above or to the right can scroll by using the cursor key or SHIFT/9 or SHIFT/3 for the Z-100 or PGUP/PGDN for the PC. Side movement using the TAB or SHIFT/HELP (TAB or SHIFT/TAB for the PC) will move the entire spreadsheet, both top and bottom sections. Note that the size of the split sections is determined by the position of the cursor when first selected. After positioning the cursor, press (F10) or SLASH (/) and then MCM for the top line menu. Select (T)his Window and (S)plit. To keep the cells in both windows from scrolling, you can select (U)nsync. This selection will permit only one section to scroll while moving about in the other section. (C)lear will return the spreadsheet to normal operation. As you become more familiar with ENABLE, you can invoke the split screen capability by using the expert commands. The Split Window commands are (F9 W H) to split the window horizontally, (F9 W V) for vertical split, (F9 W S) for a synchronized split or (F9 W U) for unsynchronized split. (F9 W X) will clear the split window. These commands are easy to learn as they indicate the de-

```

C14:                                     POINT
Enter range to sort: A3..C14
      A          B          C          D          E          F          G          H
1  ENABLE's Sort
2
3  FFF          234          THESE ARE LINES
4  HHH          5645         MORE LINES-
5  WWW          123          DIFFERENT LINES
6  NNN          67           THESE ARE IN THE SORT
7  SSS          87685        EXAMPLE OF SORT
8  YYY          1234         ENABLE'S SORT
9  DDD          756          BEFORE SORT
10 OOO          3246         DIFFERENT LINES
11 WWW          2345         ALL THESE WILL BE SORTED
12 III          8765         LINES
13 DDD          2345         TEXT
14 AAA          1243         LINES AND TEXT
15
16
17
18
19
20 SCREEN DISPLAY OF HILITE FUNCTION IN ENABLE
#1  A:\HUGSIX.SSF                               Cap    W52

```

```

G1:                                     READY
      A          B          C          D          E          F          G          H
1  ENABLE's Sort
2
3  AAA          1243         THESE ARE LINES
4  DDD          2345         MORE LINES-
5  DDD          756          DIFFERENT LINES
6  FFF          234          THESE ARE IN THE SORT
7  HHH          5645         EXAMPLE OF SORT
8  III          8765         ENABLE'S SORT
9  NNN          67           BEFORE SORT
10 OOO          3246         DIFFERENT LINES
11 SSS          87685        ALL THESE WILL BE SORTED
12 WWW          123          LINES
13 WWW          2345         TEXT
14 YYY          1234         LINES AND TEXT
15
16
17
18
19
20 SCREEN DISPLAY OF HILITE FUNCTION IN ENABLE
#1  A:\HUGSIX.SSF                               Cap    W52

```

Figure 5
Before and after sort display
where total area was not selected.

B24: @SUM(A23..C23) COMM
 Highlight all cells with formulas that refer to a given cell
 Enter cell whose related cells are to be highlighted: B24

	A	B	C	D	E	F	G	H
10	OOO	3246		DIFFERENT LINES				
11	SSS	87685		ALL THESE WILL BE SORTED				
12	WWW	123		LINES				
13	WWW	2345		TEXT				
14	YYY	1234		LINES AND TEXT				
15								
16								
17								
18								
19								
20	SCREEN DISPLAY OF HILITE FUNCTION IN ENABLE							
21								
22								
23	456	34	542	123				
24	490	1032	579	1155				
25								
26	946	1066	1121	1278				
27								
28								
29								

#1 SM010 Number of related cells: 3 Cap W52

Figure 6
Screen Display of HILITE

permits you to change the size of the current window and to move it to any location on the screen. By down-sizing the window, you can have several windows (parts of them) on the screen at one time. Up to eight windows can be displayed at

a time. The Modify Size or Location commands are (S)hrink, (E)xpand, or (M)ove the window. You select one of these options and then use the cursor keys to affect the requested action. The window must be reduced before it can be moved.

(ESC) will return to normal operation. You can continue to work in the down-sized window as before. If this is the second window opened, window one will be displayed around the edge of the down-sized window. You cannot move into the

B1: Col 2 READY

	B	C	D	D	E	F	G	H
1	Col 2	Col 3	Col 4	Col 4				
2								
3								
4								
5	234		THESE ARE	5THESE ARE LINES				
6	5645		MORE LINE	6MORE LINES-				
7	123		DIFFERENT	7DIFFERENT LINES				
8	67		THESE ARE	8THESE ARE IN THE SORT				
9	87685		EXAMPLE O	9EXAMPLE OF SORT				
10	1234		ENABLE'S	10ENABLE'S SORT				
11	756		BEFORE SO	11BEFORE SORT				
12	3246		DIFFERENT	12DIFFERENT LINES				
13	2345		ALL THESE	13ALL THESE WILL BE SORTED				
14	8765		LINES	14 LINES				
15	2345		TEXT	15 TEXT				
16	1243		LINES AND	16LINES AND TEXT				
17								
18								
19								
20								

#1 A:\HUGSIX.SSF Cap W54

Figure 7
Split Screen Vertical


```

D1: Col 4
Split: Close Modify Size or Location Previous-Size Zoom
Set split-screen and synchronized scrolling

```

	A	B	C	D	E	F	G	H
1	Col 1	Col 2	Col 3	Col 4				
2								

```

D1: Col 4
Horizontal: Vertical Sync Unsync Clear
Split the screen horizontally at the current row

```

	A	B	C	D	E	F	G	H
1	Col 1	Col 2	Col 3	Col 4				
2								
3	ENABLE's Sort							
4								
5	FFF	234		THESE ARE LINES				
6	HHH	5645		MORE LINES-				
7	WWW	123		DIFFERENT LINES				
8	NNN	67		THESE ARE IN THE SORT				

A10: YYY READY

	A	B	C	D	E	F	G	H
2								
3	ENABLE's Sort							
4								
5	FFF	234		THESE ARE LINES				
6	HHH	5645		MORE LINES-				
7	WWW	123		DIFFERENT LINES				
8	NNN	67		THESE ARE IN THE SORT				
9	SSS	87685		EXAMPLE OF SORT				
10	YYY	1234		ENABLE'S SORT				
11	DDD	756		BEFORE SORT				
12	OOO	3246		DIFFERENT LINES				
13	WWW	2345		ALL THESE WILL BE SORTED				
14	III	8765		LINES				
15	DDD	2345		TEXT				
16	AAA	1243		LINES AND TEXT				

Figure 8
Split Screen Horizontal

other window with cursor commands. (Z)oom will return the window to full size.

ENABLE uses the function keys in a manner similar to LOTUS. Because ENABLE is an integrated package, some function keys perform the same operation in several modules. HELP or F1 is the HELP key. Pressing this key will provide a help screen for the application you are in. The basic HELP screen can be used to select further help screens. These screens also provide a reference where you can find more information. (F2) is the pointer control key. Using this key and a cell address or range name will move the cursor directly to that location. Using the F2 and a right or left arrow key will move the cursor to the last or first non-blank cell in a row. (F3) will insert a row at the cursor. (F4) is the edit key and will permit the contents of the cell to be edited using the arrow keys to position the cursor. (F5) will force a manual recalculation of the spreadsheet. (F6) permits moving between halves of a split window. (F7) makes cell references absolute if they are pointed out. To do this, place the first operator in the new cell, then move the cursor to the cell that you want in the new cell. Press (F7) and the value in the new cell will be an absolute value of the pointed cell. (F8) will copy the contents of a range of cells to a

A26: +A25+B25

READY

	A	B	C	D	E	F	G	H
1	Col 1	Col 2	Col 3	Col 4				
2								
3	ENABLE's Sort							
10	YYY	1234		ENABLE'S SORT				
11	DDD	756		BEFORE SORT				
12	OOO	3246		DIFFERENT LINES				
13	WWW	2345		ALL THESE WILL BE SORTED				
14	III	8765		LINES				
15	DDD	2345		TEXT				
16	AAA	1243		LINES AND TEXT				
17								
18								
19								
20								
21								
22	SCREEN DISPLAY OF HILITE FUNCTION IN ENABLE							
23								
24								
25	456	34	542	123				
26	490	1032	579	1155				
#1	A:\HUGSIX.SSF						Cap	W54

Figure 9
TITLE Screen

	A	B	C	D	E	F	G	H
10	YYY	1234		ENABLE'S SORT				
11	DDD	756		BEFORE SORT				
12	OOO	3246		DIFFERENT LINES				
13	WWW	2345		ALL THESE WILL BE SORTED				
14	III	8765		LINES				
15	DDD	2345		TEXT				
16	AAA	1243		LINES AND TEXT				
17								
18								
19								
20								
21								
22	SCREEN DISPLAY OF HILITE FUNCTION IN ENABLE							
23								
24								
25	456	34	542	123				
26	490	1032	579	1155				
27								
28	946	1066	1121	1278				
29								

Press M(move) or S(shrink) or E(expand). Then use eCap hen done.

Figure 10
Modify Windows Command Screen

new location. This is the expert command for F10 (W)orksheet (C)opy command explained in the introduction to spreadsheets. (F9) is the expert key and is used with other keys as explained above. (F10) or SLASH (/) brings up the Top Line menu structure.

The function keys are doubled by using the ALT key, which is the F0 key on the Z-100. On the Z-100, you press F0 and then the second function key; while on the PC, you must press both keys together. F0 F1 will display the function key chart at the top of the screen. This is a toggle

key function, so pressing the F0 F1 again will remove this display. F0 F2 is the print command and will use the default setting. This will use the default printer and all of the spreadsheet. F0 F3 will delete the row at the cursor. F0 F4 will center the contents of the cell under the cursor. F0 F5

B18:

READY

Other MCM options available
The SPLIT screen has done th
of what its label says, TH
closes the current window.
in detail in this series is
in all modules of ENABLE and
Location permits you to chan
to any location on the scree
window (parts of them) on th
displayed at time. The Modi
(E)xpand, or (M)ove the wind
the cursor keys to effect th
before it can be moved. {ES
continue to work in the down
window opened, window one wi
window. You can not move in
will return the window to fu

	B	C	D	E	F
3					
4					
5	234		THESE ARE LINES		
6	5645		MORE LINES-		
7	123		DIFFERENT LINES		
8	67		THESE ARE IN THE SORT		
9	87685		EXAMPLE OF SORT		
10	1234		ENABLE'S SORT		
11	756		BEFORE SORT		
12	3246		DIFFERENT LINES		
13	2345		ALL THESE WILL BE SORTED		
14	8765		LINES		
15	2345		TEXT		
16	1243		LINES AND TEXT		
17					
18					

FIGURE TEN - Modify windows

FIGURE ELEVEN - Down sized window overlay on top of Word Processing.

#1 A:\HUGSIX.SSF

Cap W54

Figure 11
Down-sized Window Overlay
on Top of Word Processing

will copy data between windows. This is not the windows of the split screen, but windows of different spreadsheets or modules. F0 F6 will permit you to skip protected cells. This feature will cause the cursor to skip over cells that have been protected. To have cells protected from overwrite, you must turn on global protection using F10 W G P (D) changes. You must then protect the cells using F10 W R P, and then name the cells. Using this procedure, ENABLE will BEEP and not permit changes to the protected cells. By using F0 F6, ENABLE will automatically skip over the protected cells. F0 F7 will bring up the graphics options. This is the same as F10 G. F0 F8 will permit you to MOVE a range of cells to another location in the spreadsheet. This is the same as F10 W M. F0 F9 applies to macros. ENABLE's macros are quite extensive. This capability will be explored in detail in the advanced series modules. The last of the function key options is F0 F10 which will save your work. This is the same as F10 S.

In addition to the function keys and expert command discussed, ENABLE's spreadsheet has a few more expert commands. The F9 key used in conjunction with other keys provide the same capability as using the Top Line menu, but with fewer keystrokes. Other expert commands are F9 D CHR + (F9.DEL + for the

PC) to center the contents of a cell. This set of keystrokes is not typical. The use of D CHR to center does not fall in the usual thought pattern. F9 D CHR L will delete the row at the cursor. F9 D CHR B will erase a marked range. F9 I CHR L will insert a row(s) above the cursor. You are prompted for the number of rows you want to add. A function that is unique to ENABLE is the capability of listing formulas and only formulas on screen. F9 L F will display a listing of ALL formulas in the spreadsheet. This listing includes single cell references. A similar listing is available for range names. This listing is in a window on the right side of the screen.

Along with the expert commands, keystroke and pointer commands are also available. These keystrokes, along with the expert command, are designed to save time and keystrokes to provide increased capability. SHIFT/F0 [(CTRL/[for the PC) will align the contents of a cell on the left. SHIFT/F0] (CTRL/] for the PC) will align it on the right. Remember, SHIFT/F0 is the Z-100 equivalent of CTRL. Another unique capability in ENABLE is the ability to display both spreadsheet and graph on the same screen. Using this feature, you can immediately see the results of changes in the spreadsheet graphically. To get this display, press SHIFT/F7. Two windows will

be created, the spreadsheet will appear on the left and the graph on the right. The graph displayed will be the last one created. Because of the amount of data being presented on the screen, the resolution will not be great, but it is good enough to use. The Z-100 display is very good when compared to others with less than EGA capability. In addition to normal ENABLE macros, LOTUS-like macros or those read in from a LOTUS spreadsheet can be used. LOTUS writes macros into a cell and then runs the cell. To use this capability in ENABLE, you must press SHIFT/F9 [x], where x is the cell containing the macros.

The pointer commands provide a means to move quickly around the spreadsheet. To move one screen up, press SHIFT/9 on the keypad or SHIFT/3 to go down one screen. Use the PGUP or PGDN keys on the PC. TAB will move one screen right, while SHIFT/HELP will move one screen left. TAB or SHIFT/TAB will do the same on the PC. Arrow keys will move the cursor one cell in the direction of the depressed arrow. HOME will move the cursor to cell A1. F2 and the cell address will move the cursor to that cell. SHIFT/1 HOME (END/HOME for the PC) will move the cursor to the bottom right cell. F2 -> (right arrow) will move the cursor to the last non-blank cell in the selected row,

044 (\$,2) @hlookup(\$C\$34,\$H\$47..\$W\$52,2)

COMM

```
A24: +A23+B23
A26: @SUM(A23..A24)
B24: @SUM(A23..C23)
B26: @SUM(B23..B24)
B36: @HLOOKUP(C32,H30..W45,C34)
B43: @hlookup($C$34,$H$47..$W$52,1)
B44: @hlookup($C$34,$H$47..$W$52,2)
B45: @hlookup($C$34,$H$47..$W$52,3)
B46: @hlookup($C$34,$H$47..$W$52,4)
B47: @hlookup($C$34,$H$47..$W$52,5)
C24: +A23+D23
C26: @SUM(C23..C24)
C43: +B43*$C$41
C44: +B44*$C$41
C45: +B45*$C$41
C46: +B46*$C$41
C47: +B47*$C$41
C49: +B36+@SUM(C43..C47)
C50: +C49*B50
C52: @SUM(C49..C50)
```

#1 SM009 Use up/down cursor keys to view list; use any other key to exit

Figure 12
Screen Dump of Formula Listing

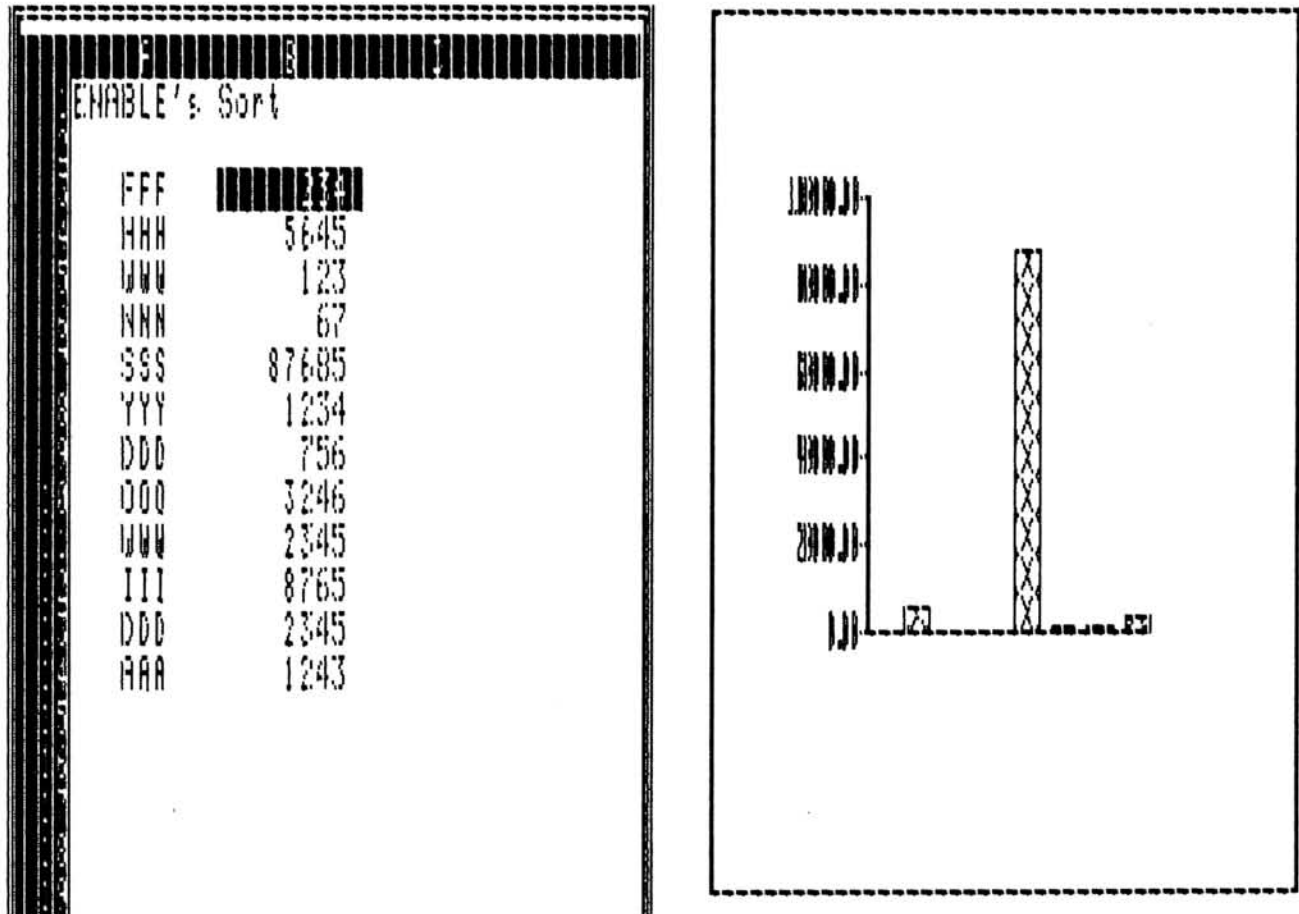


Figure 13
Screen Dump of Spreadsheet
And Graph

while F2 <- (left arrow) will move the cursor to the first non-blank cell in the row. SHIFT/1 and an arrow key will move the cursor to the next or last non-blank cell in the direction indicated. This is the end of the expert/keystroke/pointer commands. All of these commands are listed along with the function keys on the blue plastic overlay provided with the package.

Like other spreadsheets, ENABLE has many functions which make complex formulas easy to enter. These functions are added by moving the cursor to the cell where it is needed and inserting it like any other formula. These functions start with the "@" sign. The simplest of these functions is the @sum command. This command will sum a row or column of cells. In LOTUS, it is possible to write this as @SUM(B1..B7) even though the documen-

tation says that two periods are required between the cells. ENABLE requires both periods to work. This function will work with a single row or column, or multiple rows and columns, or a range. It is also possible to sum individual cells throughout the spreadsheet by indicating all cells, i.e., @SUM(A1..B4..F2..C5). You can also mix row/columns and individual cells, i.e., @SUM(A1..A6..F3..G8). The same numbers could have been added by typing in "+A1+A2+A3+A4+A5+A6+F3+G8." Functions can also be used as part of a formula, i.e., (B7-F8)*@sum(a1..a8) or even several functions in the same formula, i.e., @sum(A1..b7)+@sum(f7..f20). Functions can also use range names, i.e., @sum(cashflow) to perform the request operation.

Two similar functions are @HLOOKUP or

horizontal lookup, and @VLOOKUP or vertical lookup. These functions permit you to look up a value in a table based on a specified value. This capability requires a table be input with values. A comparison set of values is also required. The cell that requires the result of the lookup would contain the following formula as an example, @VLOOKUP(A5..F2..F12,1). This formula would request ENABLE to use the value in cell A5, the value being looked up, against the values in the range of cells F2..F12. The selected value would be located in a cell next to the selected cell in the range, but in the range G2..G12. @HLOOKUP works the same, except the table is horizontal.

ENABLE will also generate a Julian date in a spreadsheet with the function @JULIAN(expression) and is available only in

	G	H	I	J	K	L	M	N
1		Lookup table						
2		Dayton	SAC	WASH				
3			1	2	3			
4		Dayton-1	na	950	350			
5		SAC - 2	950	na	1150			
6		WASH - 3	350	1150	na			
7								
8								
9								
10		Airfare						
11		From	2					
12		To	3					
13								
14			1150					
15								
16								
17								
18								
19								
20								
#1	A:\HUGSIX.SSF						Cap	W54

Figure 14
Vertical Lookup Table

version 2.0. This is a convenient function and can be used in any of the ENABLE modules. By using @TODAY as the expression in the Julian date function, the correct Julian date can be obtained, i.e., @JULIAN(@TODAY). Other date functions are @DATE(YY,MM,DD), which will change the date to an integer from the base date used by ENABLE which is 31 December 1899. @DATE(integer) will provide the date in the form YY/MM/DD. @DATE(string) will convert the string "YY/MM/DD" to an integer based on the ENABLE base date. @DATE\$(integer) uses the integer and outputs the date in a string, i.e., November 10, 1942. With 0 equalling Sunday, @WEEKDAY(integer) will output a number equal to the day of the week. @WEEKDAY\$(integer) will do the same, except the output will be a day, information for the year.

Along with the date functions, there are several time functions. @ETIME will provide an elapsed time between the arguments. The complete function is @ETIME(begindate, begintime, enddate, endtime). Other functions can be used for these arguments, i.e., @TODAY, @NOW, etc. The result is given in seconds. @HOUR(integer) will convert the integer, which is seconds to whole hours. @NOW will return the number of seconds since midnight. @TIME(HM,MM,SS) will return the number of seconds in the expression. @TIME

(integer) will return HM:MM:SS for the integer in the expression. @TIME\$(string) will provide a number consisting of the number of seconds in the string in text time format.

The above functions are in addition to the normal @ABS (absolute value), @ACOS (arc cosine), @ASIN (arc sine), @ATAN (arc tangent), @COS (cosine), @EXP (exponentiation), @LOG (logarithm base 10), @PI (PI to the value of 3.141592653589794), @SIN (sine), and @TAN (tangent) which are used in mathematical formulas. ENABLE also supports @AVG (average), @CUMPCT (cumulative percentages), @CUMSUM (cumulative sum), @FV (future value), @IRR (internal rate of return), @NPV (net present value), @PMT (payment per period), and @PV (present value of an annuity) which are used in financial formulas. These are not all of the functions available in ENABLE, but will give you a feel for the completeness of the package. To give examples and explain in detail all of the above functions would take a complete article in itself. If you need additional information, the reference material provided with ENABLE will help. Books on LOTUS or other spreadsheets will also help as most functions are common between packages. Applications written to run on LOTUS will, for the most part, run on ENABLE. Because of the integration, the size and

available memory will limit large spreadsheets. The Z-100 with 768K of main memory will permit larger applications than on PCs, which are limited to 640K.

The advanced spreadsheet article will cover graphics. I will provide a discussion on the advanced graphics capabilities of ENABLE called Perspective in an upcoming article. This super graphics package will not run on the Z-100, as it requires EGA capability even though the Z-100 could handle the display in an interlace mode. Also included will be transfer of data between windows and the already mentioned macro discussion. *

Did you know that HUG has a small business accounting package? Its unique name is **Accounting System**. As with most HUG software, it is user-friendly, double entry, can handle up to 999 separate accounts during any calendar year, and is available for ANY Heath/Zenith computer with a double density disk drive. The different versions available are as follows: **CP/M — P/N 885-8047-37, Z-DOS/MS-DOS — P/N 885-8048-37, MS-DOS — P/N 885-8049-37.**

Continued from Page 20

```
while(ptr->next != 0 && row < 25) /* clear screen from current row dn */
{
    locate(row,1);print(ptr->string,row,1); /* print new line */
    ptr->ptr->next;row++; /* get pointer to next line */
    if(row < 25) {locate(row,1);print(ptr->string,row,1);}
    ptr->tmpr,row=temp;locate(row,col); /* restore all values */
}

update_lineno(val) /* gets line num, increments it, */
register val; /* & stores new value */
{
    tmpr=ptr; /* save pointer */
    while(ptr->next != 0)
    {
        ptr=ptr->next;ptr->lineno=val; /* store new line no */
        val++;
    } ptr=tmpr; /* restore pointer */
}

del_line() /* deletes a line by removing */
{
    int val; /* its link to others & reestablishing link */
    /* see text */
    if(ptr->next==0 && ptr != base)
    {fptr=ptr;ptr->prev->next=0;lines--; /* deleting last line */
    ptr->ptr->prev;if(row > 1) row--;refresh_screen();
    rumptr=ptr;ptr->next=0;return;}
    if(lines <= 2) { create();first();create();check(ptr);
    rest();ptr=base;refresh_screen();return;}
    if(base == ptr && lines > 2) {fptr=ptr;
    ptr->ptr->next;base=ptr; /* deleting first line */
    refresh_screen();val=ptr->lineno-1;ptr->prev=0;
    update_lineno(val+1);lines--;return;}
    if(base==ptr && lines ==1)
    { create();first();ptr->lineno=1;refresh_screen();check(ptr);}
    fptr=ptr;
    ptr->prev->next=ptr->next;ptr->next->prev=ptr->prev; /* deleting inter- */
    ptr->ptr->prev;row-=1;refresh_screen();val=ptr->lineno; /* mediate lines */
    update_lineno(val+1);lines--;free(fptr); /* free memory */
}

clean_it(string,pos,num)
char string[];
int pos, num;
{
    int i;pos-=1;
    for(i=pos; i < num; ++i)
        string[i]=0;
}

del_ch(stng,pos) /* deletes a character at cursor pos */
char stng[];
int pos;
{
    register i, j;
    int len, len1;
    char string1[160];clean_it(string,1,160);
    if(stng[0]==VACANT) return;
    if(stng[0]==SPACE && stng[1]=='\0' && ptr->next != 0)
{
    string1[0]=SPACE;string1[1]='\0';
    join(string1,pos);goto joint1;}
    if(ptr->next==0 && ptr->string[0]==VACANT) return;
    for(j=0; j <= pos; ++j) /* string1 = ptr->string up to cursor */
        string1[j]=stng[j]; /* is cursor at end of line? */
    return; else if(stng[j-1]==VACANT && ptr->next != 0)
    {join(string1,pos);goto joint1;} /* if so, join it */
    for(i=pos; i <= 79; ++i) /* if not, get string to right of */
        string1[i]=stng[i+1];joint1; /* cursor & append it to that to the left */
    if(flag_long==1) add_ch(string1);
    clean_it(stng,1,80);
    strcpy(stng,string1,80); /* copy new string into storage */
    print(stng,row,1); /* print new string & erase end of old */
    locate(row,col);
}

join(addon,pos) /* joins the next line with the current */
unsigned char addon[]; /* line at the cursor position */
int pos;
{
    int len, j;
    char buff2[1602];clean_it(buff2,1,162);
    if(ptr->next==0) return;
    len=strlen(addon); /* find the length of current line */
    for(j=len; j <= pos; ++j) /* pad with spaces (to cursor pos) if needed */
        addon[j]=SPACE;addon[j-1]='\0';
    strcpy(buff2,addon);
    strcat(buff2,ptr->next->string,80); /* concatenate old line with new */
    len=strlen(buff2);
    if(len > 79) { for(j=0; j <= len; ++j)
        buff2[j]=buff2[j+78];
        buff2[78]='\0';buff2[j]='\0';
        flag_long=1;strcpy(addon,buff2,80);
        tmpr=ptr;ptr->ptr->next;row++; /* get pointer to next line & move there */
        del_line();ptr=tmpr;locate(row,col); /* delete (old) existing line */
    }
    add_ch(s)
    unsigned char s[];
    {
        int len, j;len=strlen(buff);if(len < 2) flag_long=0;
        strcat(s,buff,1);for(j=0; j <= len;++j) buff[j]=buff[j+1];
        buff[j]='\0';
    }
    ins_ch(stng,pos) /* inserts character from keyboard */
    unsigned char stng[]; /* into current line at cursor pos */
    int pos;
    {
        char string1[80], string[80], c;
        clean_it(string1,1,80);clean_it(string,1,80);
        fix_line();
        while(in==1)
        {
            strcpy(string1,stng); /* copy existing string into string1 */
            for(i=0; i < pos; ++i)
                string[i]=string1[i]; /* truncate string at cursor pos */
        }
    }
}

```

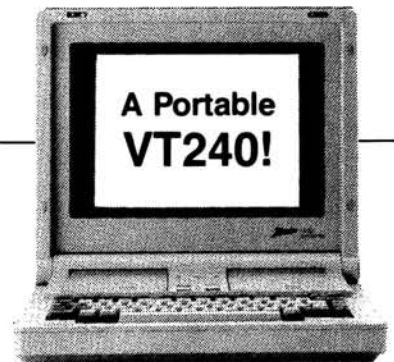


```

first();          /* link as first */
else rest();     /* link as any other line */
skipl:
for(i=col-1;i < 80 && ((c=get_ch())!= 68 || flag != 1); ++i); /* exit\F10 */
/* above line of code is input loop as well as a trap for exit char */
fix_line();
if(i==80) {col=1;i=0;if(chr != 13) row++;locate(row,col);} /* fix */
if(flag == 1 && c == 68) /* F10 key to act as exit signal */
end=TRUE; /* exit condition, flag main */
}

main()
{
int lev;
char ch, dummy[2];dummy[0]=""; /* initialize variables */
while(dummy[0] != '4')
{
cls();lines=0;row=lev=col=1;end=init=FALSE;
locate(2,37);print_string(BOLD,"Edtxt");drawbox(2,3,78,24);
drawbox(4,4,76,23);locate(6,22);
printf("Please enter...");locate(9,28);printf("1. Create new document.");
locate(11,28);printf("2. Edit existing document.");
locate(13,28);printf("3. Print document (hard copy).");
locate(15,28);printf("4. Exit to operating system.");
locate(18,22);printf("the number corresponding to your choice ");
gets(dummy);
switch (dummy[0])
{
case '1': { cls();
while( end != TRUE)
{
build_line();
}
fix_line();cls();
open_file();write_file();
break; } /* end of case 1 */
case '2':{ cls();open_file();read_file();ptr=base;cls();lev=1;
do
{
locate(lev,1);prnt(ptr->strng,lev,1);ptr=ptr->next;lev++;
} while(ptr->next != 0 && lev < 25);
if(lev < 25) {locate(lev,1);prnt(ptr->strng,lev,1);}
ptr=base;if(ptr->next !=0) skip=TRUE;locate(1,1);row=col=1;update_screen();
while(end != TRUE)
build_line();
write_file();
break; } /* end of case 2 */
case '3': { cls();
open_file();read_file();ptr=base;fclose(fd);
f2=fopen("prn","w");
putc(27,f2);putc(21,f2);
do
{
for(i=0; ptr->strng[i] != '\0'; ++i)
putc(ptr->strng[i],f2);putc('\n',f2);ptr=ptr->next;
} while(ptr->next != 0);
break; } /* end of case 3 */
}
} /* end of while */
}

```



Use your Z181/183

as a portable VT220/240 workstation with:

- True double-high / double-wide
- 132 column mode via 128 column display
- True smooth scrolling
- KERMIT & XMODEM file transfers
- VMS Services for MS-DOS Support

ZSTEMpc™ — VT220 Emulator \$150

ZSTEMpc™ — VT240 Emulator \$295

including 4014 and ReGIS graphics.

also available: VT100, D400 & 4014 emulators and the *PowerStation™* VT200 style keyboard

KEA Systems Ltd.

#412 - 2150 West Broadway
Vancouver, B.C. Canada V6K 4L9

Telephone (604) 732-7411

Telex 04-352848 VCR Fax (604) 732-0715

Order Toll Free (800) 663-8702

30 day money back guarantee AMEX/MC/VISA

SPEED MODS - Assembled and Tested - \$34.95

- H/Z 150/160 Software Select 6.67 or 4.77 MHz
- Hardware Reset included. Satisfaction Guaranteed
- H/Z 89 Software Select 2 or 4 MHz. No Trace Cuts

H/Z 89 20 Meg Hard Disk only \$475!

Boots from Hard Disk. Supports CDR and Magnolia.
All Software and Hardware except Hard Disk \$175

100/148/150 Hardware Smart Watch \$ 35

20 Meg ST-225 \$279 (drive only \$249) 30 Meg \$309

MT PAL 640/704k RAM \$19.95 Super PAL 1.2 Meg \$45

148 Expansion bus \$129 EGA \$145 AT 3 Meg RAM \$139

MS-DOS and CP/M Software:

Perfect Funds \$29 MT Accountant \$19 Paycheck \$39

Perfect Money \$19 Perfect Printer \$19

Micronics Technology (205) 244-1597

We Moved to: 410 Bellehurst, Montgomery, AL 36109

Checks, VISA, MC. Shipping \$2, Hard Disk \$10

Call or Write for our FREE complete Catalog

Continued from Page 32

The solution I have applied is to place a patch in the portion of the program which checks the character count beyond the root directory character "\ " in the path statement. This causes it not to go to these lookup tables. When this routine is entered, a change to the program's command buffer, as I will call it for lack of a better name, would result in the path statement.

Testing thus far allows the program to work in this area without any path error. Sometimes it will occur and show two root directory characters. If this happens, just enter the path statement without the preceding \. If an error is made on entry resulting from fat fingering, the program does return an error, but with whatever was entered by the Human.

The program will crash if a Copy, Delete or Rename function is tried from the Maintenance Menu, but the above changes will allow you to use the database functions. I have not tested the program to its fullest capabilities, and there may still be other problems beyond this, but this I feel, was a major one. If anyone has any other patch information, I would appreciate hearing from them.

I have included both the patch data to be added to the PATCHER.DAT file, along with the sections of code before and after the patches.

Ronald J. Vince
Springfield, Ohio

```
PC-FILE + ver. 1.0 (Dated 3-01-87)
Insert the disk containing PCF.EXE (w/byte count of 284691)
PCF.EXE
2ABB,CD,90
2B24,CD,91
2B2A,90,90
2B2E,90,90
2B32,90,90
9C76,C4,5E,F0,90,90
14C21,90,90,90,90,90
14C27,90,90,90,90,90
14C36,90,90,90,90,90
14C3C,90,90,90,90,90
Z
```

***** Keyboard Port Patches *****

PCF.EXE (ORIGINAL CODE)

```
401F:2BB0 00B09C1E  ADD      [BX+SI+1E9C],DH
401F:2BB4 06        PUSH     ES
401F:2BB5 51        PUSH     CX
401F:2BB6 53        PUSH     BX
401F:2BB7 50        PUSH     AX
401F:2BB8 57        PUSH     DI
401F:2BB9 56        PUSH     SI
401F:2BBA FB        STI
401F:2BBB E460     IN      AL,60 <-- Keyboard Port
401F:2BBD BB4000   MOV     BX,0040
401F:2BC0 8EDB     MOV     DS,BX
401F:2BC2 BE1700   MOV     SI,0017
401F:2BC5 8A1C     MOV     BL,[SI]
401F:2BC7 80E30C   AND     BL,0C
401F:2BCA 7433     JZ      2BFF
401F:2BCC 80FB08   CMP     BL,08
401F:2C20 8A05     MOV     AL,[DI]
401F:2C22 F8        CLC
401F:2C23 C3        RET
401F:2C24 E461     IN      AL,61 <-- Keyboard Port
401F:2C26 8AE0     MOV     AH,AL
401F:2C28 0C80     OR      AL,80
401F:2C2A E661     OUT     61,AL
401F:2C2C 86C4     XCHG   AL,AH
401F:2C2E E661     OUT     61,AL
401F:2C30 B020     MOV     AL,20
401F:2C32 E620     OUT     20,AL
401F:2C34 C3        RET
```

PCF.EXE (PATCHED CODE)

```
401F:2BB0 00B09C1E  ADD      [BX+SI+1E9C],DH
401F:2BB4 06        PUSH     ES
401F:2BB5 51        PUSH     CX
401F:2BB6 53        PUSH     BX
401F:2BB7 50        PUSH     AX
401F:2BB8 57        PUSH     DI
401F:2BB9 56        PUSH     SI
401F:2BBA FB        STI
401F:2BBB CD90     INT     90 <-- Patched ZPC keyboard
401F:2BBD BB4000   MOV     BX,0040 interrupt 90
401F:2BC0 8EDB     MOV     DS,BX
401F:2BC2 BE1700   MOV     SI,0017
401F:2BC5 8A1C     MOV     BL,[SI]
401F:2BC7 80E30C   AND     BL,0C
401F:2BCA 7433     JZ      2BFF
401F:2BCC 80FB08   CMP     BL,08
401F:2C20 8A05     MOV     AL,[DI]
401F:2C22 F8        CLC
401F:2C23 C3        RET
401F:2C24 CD91     INT     91 <-- Patched ZPC keyboard
401F:2C26 8AE0     MOV     AH,AL interrupt 91
401F:2C28 0C80     OR      AL,80
401F:2C2A 90        NOP     <-- Patched NOP inst.
401F:2C2B 90        NOP     <--
401F:2C2C 86C4     XCHG   AL,AH
401F:2C2E 90        NOP     <-- Patched NOP inst.
401F:2C2F 90        NOP     <--
```



```

401F:2030 B020 MOV AL,20
401F:2032 90 NOP
401F:2033 90 NOP
401F:2034 C3 RET

```

```

<--- Patched NOP inst.
<---

```

```

***** Lookup Table Routine *****
NOTE: This routine is also used during the Report Gen.
portion of the program. The changes cannot be made
here.

```

```

PCF_EXE (ORIGINAL CODE)
401F:944B 55 PUSH BP
401F:944C 8BEC MOV BP,SP
401F:944E 83EC04 SUB SP,+04
401F:9451 57 PUSH DI
401F:9452 56 PUSH SI
401F:9453 2BFF SUB DI,DI
401F:9455 C45E06 LES BX,[BP+06]
401F:9458 26803900 CMP Byte Ptr ES:[BX+DI],00 <--- RESULT: ES=62C5, BX=0690
401F:945C 7428 JZ 9486
401F:945E 2BF6 SUB SI,SI
401F:9460 C45E0A LES BX,[BP+0A]
401F:9463 26803800 CMP Byte Ptr ES:[BX+SI],00 <--- AT START SI=0000
401F:9467 7417 JZ 9480
401F:9469 268A00 MOV AL,ES:[BX+SI]
401F:946C C45E06 LES BX,[BP+06]
401F:946F 263801 CMP ES:[BX+DI],AL
401F:9472 750F JNZ 9483
401F:9474 C45E0E LES BX,[BP+0E]
401F:9477 268A00 MOV AL,ES:[BX+SI]
401F:947A C45E06 LES BX,[BP+06]
401F:947D 268801 MOV ES:[BX+DI],AL
401F:9480 47 INC DI
401F:9481 EBD2 JMP 9455
401F:9483 46 INC SI
401F:9484 EBD4 JMP 9460
401F:9486 5E POP SI
401F:9487 5F POP DI
401F:9488 8BE5 MOV SP,BP
401F:948A 5D POP BP
401F:948B CB RETF

```

```

NOTE: The command buffer which
contains the Drive letter
only at time of entry into
this routine is
located at 62C5:0690

```

```

PCF_EXE (PATCHED CODE)
401F:9D50 088BF80B OR [BP+DI+0BF8],CL
401F:9D54 FF7508 PUSH [DI+08]
401F:9D57 FF760C PUSH [BP+0C]
401F:9D5A FF760A PUSH [BP+0A]
401F:9D5D EB73 JMP 9DD2
401F:9D5F 0BF6 OR SI,SI
401F:9D61 7505 JNZ 9D68
401F:9D63 8BF868 MOV AX,68F8
401F:9D66 EB68 JMP 9DD0
401F:9D68 B85C00 MOV AX,005C <--- ES Segment value for first
401F:9D6B 50 PUSH AX table
401F:9D6F 50 MOV AX,002F <--- BX Register value for first
401F:9D70 FF7614 PUSH AX table pointer
401F:9D73 FF7612 PUSH [BP+12]
401F:9D76 C45E0F LES BX,[BP-10] <--- Added patch to all the ES &
401F:9D79 90 NOP BX registers to contain the
401F:9D7B 83C408 ADD SP,+08 same values as if call was
made.

```

```

***** Video Port Patches *****
PCF_EXE (ORIGINAL CODE)
501F:4D1C BADA03 MOV DX,03DA <--- Video Port Address
501F:4D1F B401 MOV AH,01
501F:4D21 EC IN AL,DX
501F:4D22 84C4 TEST AH,AL
501F:4D24 75FB JNZ 4D21
501F:4D26 FA CLI
501F:4D27 EC IN AL,DX
501F:4D28 84C4 TEST AH,AL
501F:4D2A 74FB JZ 4D27
501F:4D2C AD LODSW
501F:4D2D FB STI
501F:4D2E C3 RET
501F:4D2F 8BD8 MOV BX,AX
501F:4D31 BADA03 MOV DX,03DA <--- Video Port Address
501F:4D34 B401 MOV AH,01
501F:4D36 EC IN AL,DX
501F:4D37 84C4 TEST AH,AL
501F:4D39 75FB JNZ 4D36
501F:4D3B FA CLI
501F:4D3C EC IN AL,DX
501F:4D3D 84C4 TEST AH,AL
501F:4D3F 74FB JZ 4D3C
501F:4D41 8BC3 MOV AX,BX

```

```

PCF_EXE (PATCHED CODE)
401F:9D50 088BF80B OR [BP+DI+0BF8],CL
401F:9D54 FF7508 PUSH [DI+08]
401F:9D57 FF760C PUSH [BP+0C]
401F:9D5A FF760A PUSH [BP+0A]
401F:9D5D EB73 JMP 9DD2
401F:9D5F 0BF6 OR SI,SI
401F:9D61 7505 JNZ 9D68
401F:9D63 8BF868 MOV AX,68F8
401F:9D66 EB68 JMP 9DD0
401F:9D68 B85C00 MOV AX,005C <--- ES Segment value for first
401F:9D6B 50 PUSH AX table
401F:9D6F 50 MOV AX,002F <--- BX Register value for first
401F:9D70 FF7614 PUSH AX table pointer
401F:9D73 FF7612 PUSH [BP+12]
401F:9D76 C45E0F LES BX,[BP-10] <--- Added patch to all the ES &
401F:9D79 90 NOP BX registers to contain the
401F:9D7B 83C408 ADD SP,+08 same values as if call was
made.

```

```

NOTE: After exiting this
routine, the path
statement is placed in
the command buffer
along with the new drive
letter character.

```

```

PCF_EXE (ORIGINAL CODE)
401F:9D50 088BF80B OR [BP+DI+0BF8],CL
401F:9D54 FF7508 PUSH [DI+08]
401F:9D57 FF760C PUSH [BP+0C]
401F:9D5A FF760A PUSH [BP+0A]
401F:9D5D EB73 JMP 9DD2
401F:9D5F 0BF6 OR SI,SI
401F:9D61 7505 JNZ 9D68
401F:9D63 8BF868 MOV AX,68F8

```

```

PCF_EXE (PATCHED CODE)
401F:9D50 088BF80B OR [BP+DI+0BF8],CL
401F:9D54 FF7508 PUSH [DI+08]
401F:9D57 FF760C PUSH [BP+0C]
401F:9D5A FF760A PUSH [BP+0A]
401F:9D5D EB73 JMP 9DD2
401F:9D5F 0BF6 OR SI,SI
401F:9D61 7505 JNZ 9D68
401F:9D63 8BF868 MOV AX,68F8

```



```

501F:4D43 AB      STOSW
501F:4D44 FB      STI
501F:4D45 C3      RET

```

PCF.EXE (PATCHED CODE)

```

501F:4D1C BADA03  MOV     DX,03DA    <-- Video Port Address
501F:4D1F B401    MOV     AH,01
501F:4D21 90      NOP
501F:4D22 90      NOP
501F:4D23 90      NOP
501F:4D24 90      NOP
501F:4D25 90      NOP
501F:4D26 FA      CLI
501F:4D27 90      NOP
501F:4D28 90      NOP
501F:4D29 90      NOP
501F:4D2A 90      NOP
501F:4D2B 90      NOP
501F:4D2C AD      LODSW
501F:4D2D FB      STI
501F:4D2E C3      RET
501F:4D2F 8BD8    MOV     BX,AX
501F:4D31 BADA03  MOV     DX,03DA    <-- Video Port Address
501F:4D34 B401    MOV     AH,01
501F:4D36 90      NOP
501F:4D37 90      NOP
501F:4D38 90      NOP
501F:4D39 90      NOP
501F:4D3A 90      NOP
501F:4D3B FA      CLI
501F:4D3C 90      NOP
501F:4D3D 90      NOP
501F:4D3E 90      NOP
501F:4D3F 90      NOP
501F:4D40 90      NOP
501F:4D41 8BC3    MOV     AX,BX
501F:4D43 AB      STOSW
501F:4D44 FB      STI
501F:4D45 C3      RET

```

*

HFS - III

The New Home Finance System from Jay Gold Software

You don't have to be an accountant to keep your accounts in order.

- Use HFS-III to track cash in up to 100 Asset Accounts and 100 Credit Accounts. Number of transactions limited only by disk space.

- Prints checks in any of 3 formats that you design using any standard business-sized check. Prints payee address (if desired) for use with window envelopes.

- Asset Accounts include checking, regular asset accounts such as savings, IRAs, CDs, or cash, parent accounts that keep track of the total balance of their "children" and reserve accounts which let you hide funds in your checking account until you need them.

- Model transactions allow commonly used transactions to be recalled for later entry with a few keystrokes (number of models limited only by disk space)

- Easy selection of accounts and codes from pop-up menus or by entry of name or number.

- Help boxes available at a keystroke.

- Fast. 100% assembly language code for the fastest operation possible.

- Up to 100 user defined expense codes and 15 deposit codes. Assign up to 14 expense or deposit codes to each transaction (including separate tax flag).

- Accepts HFS-II data.

Hardware: HZ-100, all Heath/Zenith PCs or any other PC/XT/AT compatible (256K), 2 disk drives, any printer.

Software: MS-DOS 2 or higher.
Price: \$99.00 (includes shipping).

MasterCard/Visa accepted, please include phone number.

Order from:



Jay Gold Software, Inc.
Box 2024
Des Moines, IA 50310
(515) 279-9821

FORM FILL-R

DISCOVER COMPUTER FORM PROCESSING

FORM FILL-R: Use with pre-printed or user created forms up to 132 columns and 132 lines

Create: Custom forms, charts, work sheets, data entry screens for pre-printed forms

Enter: Data in any desired order relative to printed form

Print: Directly or save processed forms in disk files


Modify: Easily adaptable to new forms

Examples Included: AMA form 1500, Invoices, Statements, and calendars

Requires: IBM Compatible or Z-100 computer
MS-DOS 2.0 or greater
Text editor that makes an ASCII file
Printer that handles desired forms
Line drawing requires IBM characters

\$39.95 postpaid

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

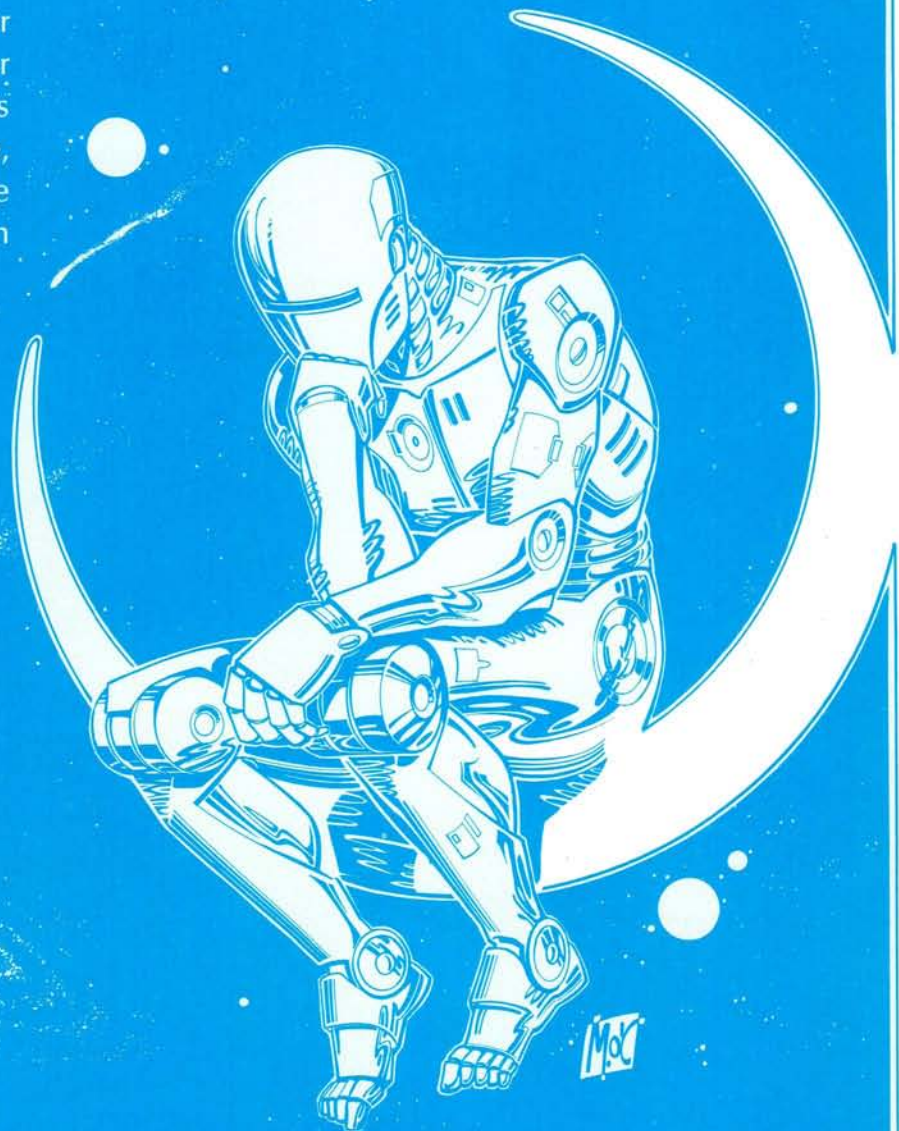


"Why Haven't You Called?"

"I've stayed up 24 hours a day waiting to hear from you. I know that for most of you, this may be your first time, but I promise, I'm very gentle! If you're still unsure, let me tell you what I have to offer. First, there's my message base. Here's where my users exchange ideas, receive assistance, and sell things. Next, is my database. Most of my users will tell you, it's second to none! With over 30 megabytes of hand-picked software it caters to all Heath/Zenith computers, but mostly PC compatibles. Finally, I have something new for you, my 'Bargain Centre'. Here you can buy surplus software and hardware, at unheard of prices. Interested? I hope so! Set your modem to either 300, 1200, or 2400 baud, and call me right now at (616) 982-3956, and register today. If you're still a bit shy, you can still call my human at (616) 982-3837 and register with him. Although he talks at 150 baud, he's gentle, too!"

MOC

Don't Get HYPER or
CROSS because your
modem software is
too complex to use,
get HUGMCP for the
no-hassle modem
connection.



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 885-2100