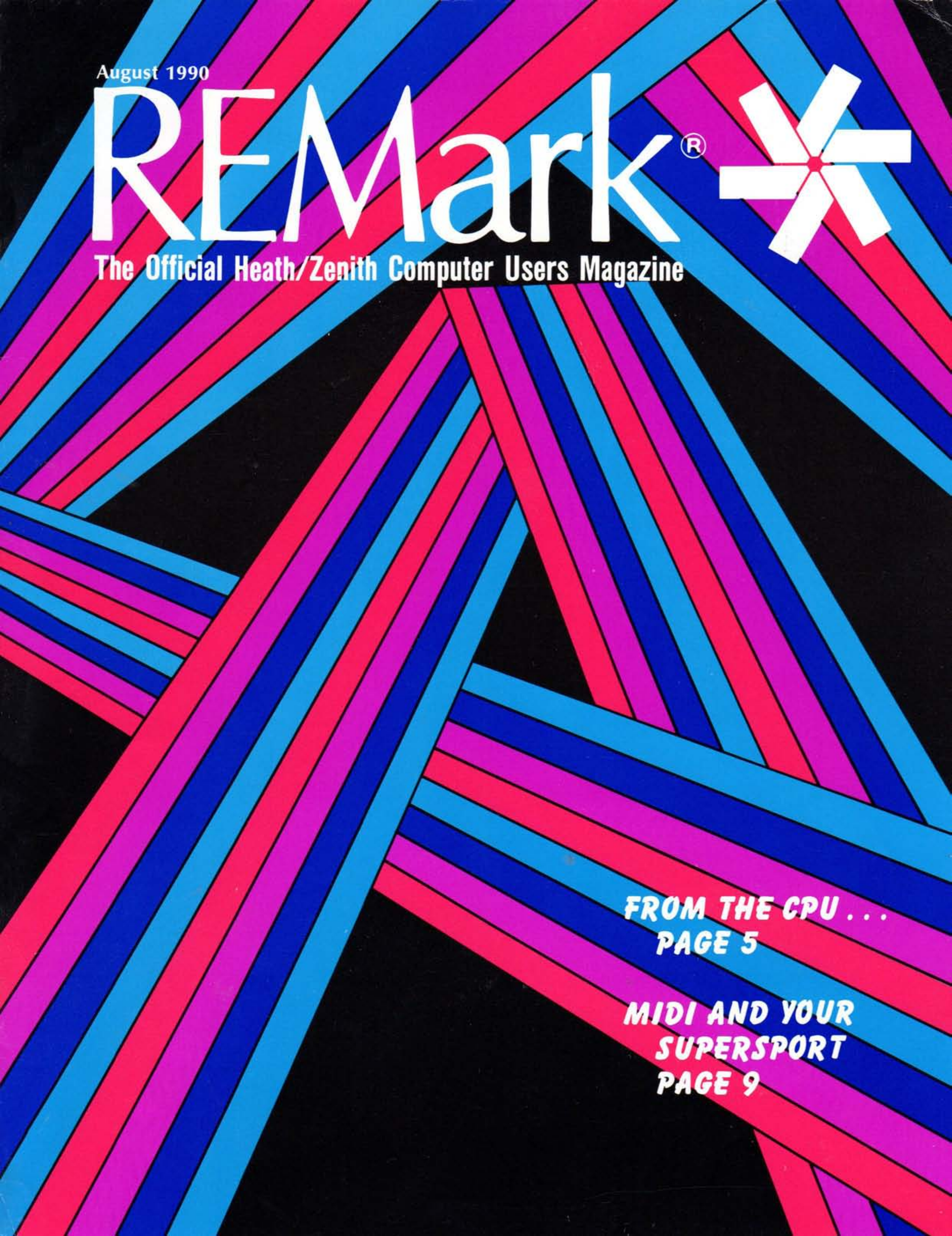
August 1990

REMark®

The Official Heath/Zenith Computer Users Magazine



FROM THE CPU . . .
PAGE 5

MIDI AND YOUR
SUPERSPORT
PAGE 9

HADES II

It's HOTTER than ever! Jam-packed with new features, HADES II still remains the easiest-to-use disk editor ever! Just look at some of the features:

- Sector Display/Editing
- Sector HEX/ASCII String Search
- File Display/Editing
- Physical and Logical Cluster Display
- File HEX/ASCII String Search
- Drive Parameter Display
- 512 MegaByte Drive Size Limit
- File Attribute Display/Edit
- Automatic Erased File Recovery
- Manual Rebuild File Recovery
- Works with Headerless MS-DOS Disks
- PC-Compatible or H/Z-100

HADES II is still only $40, and original HADES owners can upgrade their distribution disk for only $15. Call HUG today at: (616) 982-3463.

The Official Heath/Zenith Computer Users Magazine

REMark®

Volume 11, Issue 8 • August 1990

Resources

Classified Ads 1
HUG Price List 2
Buggin' HUG 4
From the CPU 5

Reader Service No.		Page No.
104	FBE Research Co., Inc.	33
175	Key Computer Publications	36
136	Lindley Systems	11
117	Payload	20
193	QuikData, Inc.	12
149	WS Electronics	4

CLASSIFIED ADS

DIABLO D-36 Daisy Wheel printer including tractor, sheet feeder, accessories $325. **Hayes 1200** Smartmodem External $125. **SK-203** buffer 512k, cables $180. **HD-4040** Packet Node Controller $100. **HD-3030** RTTY Terminal interface with HD-3006 indicator, all filters $200. **REMark Magazine Issues** 1-1978 through December 1989 $150. (708) 246-7856.

PC Compatibles

All models include the following series of computers: H/Z-130, 140, 150, 160, 170, 180, H/Z-200 and 300.

PC Compatible

WordPerfect Macros — Part 1
Edwin G. Wiggins 7

Add MIDI to Your SupersPort
Terry Perdue 9

An Introduction to Postscript
Alan Neibauer 13

Using Novell Netware on Zenith Data Systems Computers
Harold C. Ogg 16

Unix on an 8088 (Well, Almost!)
Jack W. Bazhaw 21

WordPerfect Executive
Edwin G. Wiggins 37

Using the Apple LaserWriter II With a Heath/Zenith System
Pat Swayne 39

Using an NEC Multisync Monitor on a VGA Equipped Z-150
Robert Maskasky 45

Z-100 and PC Compatible

Single-Byte Integers in BASIC
Kenneth D. Granzow 31

Assembly Language — Part 7
Pat Swayne 34

MS-DOS File Recovery
Gale Horst 41

General

A Versatile Mailing List Program in dBase
Pat Swayne 25

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
(Modem Only)

HUG Parts Ordering .... (616) 982-3463

Hardware Questions ...... (616) 982-3309

Contributing Editor...... William M. Adney

Contributing Editor...... Robert C. Brenner

Advertising.... Rupley's Advertising Service
Dept. REM, 240 Ward Avenue
P.O. Box 348
St. Joseph, MI 49085-0348
(616) 983-4550

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) 982-3463

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 Center 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/Zenith 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.

HUG

H8 - H/Z-89/90

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
ACCOUNTING SYSTEM	885-8047-37	CPM	BUSINESS	20.00
ACTION GAMES	885-1220-[37]	CPM	GAME	20.00
ADVENTURE	885-1010	HDOS	GAME	10.00
ASCIRITY	885-1238-[37]	CPM	AMATEUR RADIO	20.00
AUTOFILE (Z80 ONLY)	885-1110	HDOS	DBMS	30.00
BHBASIC SUPPORT PACKAGE	885-1119-[37]	HDOS	UTILITY	20.00
CASTLE	885-8032-[37]	HDOS	ENTERTAINMENT	20.00
CHEAPCALC	885-1131-[37]	HDOS	SPREADSHEET	20.00
CHECKOFF	885-8010	HDOS	CHECKBOOK SOFTWARE	25.00
DEVICE DRIVERS	885-1105	HDOS	UTILITY	20.00
DISK UTILITIES	885-1213-[37]	CPM	UTILITY	20.00
DUNGEONS & DRAGONS	885-1093-[37]	HDOS	GAME	20.00
FLOATING POINT PACKAGE	885-1063	HDOS	UTILITY	18.00
GALACTIC WARRIORS	885-8009-[37]	HDOS	GAME	20.00
GALACTIC WARRIORS	885-8009-[37]	CPM	GAME	20.00
GAMES 1	885-1029-[37]	HDOS	GAMES	18.00
HARD SECTOR SUPPORT PACKAGE	885-1121	HDOS	UTILITY	30.00
HDOS PROGRAMMERS HELPER	885-8017	HDOS	UTILITY	16.00
HOME FINANCE	885-1070	HDOS	BUSINESS	18.00
HUG DISK DUPLICATION UTILITIES	885-1217-[37]	CPM	UTILITY	20.00
HUG SOFTWARE CATALOG	885-4500	VARIOUS	PRODUCTS THRU 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
MAGBASE	885-1249-[37]	CPM	MAGAZINE DATABASE	25.00
MAPLE	885-8005	HDOS	COMMUNICATION	35.00
MAPLE	885-8012-[37]	CPM	COMMUNICATION	35.00
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
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

H8 - H/Z-89/90 - H/Z-100 (Not PC)

PRODUCT NAME	PART NUMBER	OPERATING SYSTEM	DESCRIPTION	PRICE
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
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 7 ISSUES 72-83	885-4007	N/A	1986	25.00
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

Price List

		OPERATING		
PRODUCT NAME	**PART NUMBER**	**SYSTEM**	**DESCRIPTION**	**PRICE**
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
DUNGEONS & DRAGONS (ZBASIC)	885-3009-37	MSDOS	GAME	20.00
ETCHDUMP	885-3005-37	MSDOS	UTILITY	20.00
EZPLOT II	885-3049-37	MSODS	PRINTER PLOTTING UTILITY	25.00
GAMES (ZBASIC)	885-3011-37	MSDOS	GAMES	20.00
GAMES CONTEST PACKAGE	885-3017-37	MSDOS	GAMES	25.00
GAMES PACKAGE II	885-3044-37	MSDOS	GAMES	25.00
GRAPHIC GAMES (ZBASIC)	885-3004-37	MSDOS	GAMES	20.00
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	885-3018-37	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
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	MSDOS	GAME	10.00
ASSEMBLY LANGUAGE UTILITIES	885-8046	MSDOS	UTILITY	20.00
BACKGROUND PRINT SPOOLER	885-3029	MSDOS	UTILITY	20.00
BOTH SIDES PRINTER UTILITY	885-3048	MSDOS	UTILITY	20.00
CXREF	885-3051	MSDOS	UTILITY	17.00
DEBUG SUPPORT UTILITIES	885-3038	MSDOS	UTILITY	20.00
DPATH	885-8039	MSDOS	UTILITY	20.00
HADES II	885-3040	MSDOS	UTILITY	40.00
HELP	885-8040	MSDOS	CAI	25.00
HEPCAT	885-3045	MSDOS	UTILITY	35.00
HUG EDITOR	885-3012	MSDOS	TEXT PROCESSOR	20.00
HUG MENU SYSTEM	885-3020	MSDOS	UTILITY	20.00
HUG SOFTWARE CATALOG UPDATE #1	885-4501	VARIOUS	PROD 1983 THRU 1985	9.75
HUGMCP	885-3033	MSDOS	COMMUNICATION	40.00
ICT 8080 TO 8088 TRANSLATOR	885-3024	MSDOS	UTILITY	20.00
MAGBASE	885-3050	VARIOUS	MAGAZINE DATABASE	25.00
MATT	885-8045	MSDOS	MATRIX UTILITY	20.00
MISCELLANEOUS UTILITIES	885-3025	MSDOS	UTILITIES	20.00
PS's PC & Z100 UTILITIES	885-3052	MSDOS	UTILITY	20.00
REMARK VOL 5 ISSUES 48-59	885-4005	N/A	1984	25.00
REMARK VOL 7 ISSUES 72-83	885-4007	N/A	1986	25.00
REMARK VOL 8 ISSUES 84-95	885-4008	N/A	1987	25.00
REMARK VOL 9 ISSUES 96-107	885-4009	N/A	1988	25.00
REMARK VOL 10 ISSUES 108-119	885-4010	N/A	1989	25.00
SCREEN DUMP	885-3043	MSDOS	UTILITY	30.00
UTILITIES II	885-3014	MSDOS	UTILITY	20.00
Z100 WORDSTAR CONNECTION	885-3047	MSDOS	UTILITY	20.00
	PC Compatibles			
ACCOUNTING SYSTEM	885-8049	MSDOS	BUSINESS	20.00
CARDCAT	885-6006	MSDOS	CATALOGING SYSTEM	20.00
CHEAPCALC	885-6004	MSDOS	SPREADSHEET	20.00
CP/EMULATOR II & ZEMULATOR	885-6002	MSDOS	CPM & Z100 EMULATORS	20.00
DUNGEONS & DRAGONS	885-6007	MSDOS	GAME	20.00
EZPLOT II	885-6013	MSDOS	PRINTER PLOTTING UTILITY	25.00
GRADE	885-8037	MSDOS	GRADE BOOK	20.00
HAM HELP	885-6010	MSDOS	AMATEUR RADIO	20.00
KEYMAP	885-6001	MSDOS	UTILITY	20.00
LAPTOP UTILITIES	885-6014	MSDOS	UTILITY	20.00
PS's PC UTILITIES	885-6011	MSDOS	UTILITIES	20.00
POWERING UP	885-4604	N/A	GUIDE TO USING PCS	12.00
SCREEN SAVER PLUS	885-6009	MSDOS	UTILITIES	20.00
SKYVIEWS	885-6005	MSDOS	ASTRONOMY UTILITY	20.00
TCSPELL	885-8044	MSDOS	SPELLING CHECKER	20.00
ULTRA RTTY	885-6012	MSDOS	AMATEUR RADIO	20.00

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.

Now Available!

HUG software is now available on 2" disks. Just put a "-90" at the end of the part number (i.e., 885-6014-90). Also add $3.00 to the purchase price of the software (i.e., $20.00 + $3.00 = $23.00).

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

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. They say it's so easy to use, they didn't even need to look at the manual. "It's the only modem software that I use, and I'm in charge of the HUG bulletin board!" says Jim Buszkiewicz. *HUGMCP* runs on ANY Heath/Zenith computer that's capable of running MS-DOS!

ORDERING INFORMATION

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

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

BUGGIN' HUG

Disk Technician Advanced

Dear HUG:

Our thanks to Robert C. Brenner for his review of our software, Disk Technician Advanced (June issue). We thoroughly enjoyed his comment that our marketing "hype" made Disk Technician Advanced seem like "the US Air Force's best discovery since Congressional porkbarrels." While he obviously had his tongue planted firmly in-cheek when he made that statement, in fact, he wasn't far off the mark.

Mr. Brenner was referring to a 27-month case study during which the Air Force tested our software on 210 new Zenith Data Systems Z-248s. Before Disk Technician Advanced was installed, hard disk-related problems had accounted for 33 percent of the service calls on these machines. After installation, there was a 100 percent drop in these service calls. The Air Force found that, by simply using Disk Technician Advanced, they could save tens of thousands of dollars in repairs, data recovery and other associated costs each year. Not enough, perhaps, to fill a porkbarrel, but a substantial amount, nevertheless.

Unfortunately, when the time came for Mr. Brenner to try Disk Technician Advanced for himself, he ran into several problems. All of which could have been avoided or easily solved.

First, when he attempted to test Disk Technician Advanced on his Heath/Zenith 386, 40 mb machine, Disk Technician Advanced properly refused to run. After contacting our technical support department, he found the reason was Disk Technician Advanced's requirement that all partitions on hard disks be 32 mb or less.

Apparently, he had inadvertently overlooked the technical specifications section in the manual where this requirement appears three times. It's also clearly listed on the packaging, and in all our marketing brochures.

We didn't include this info in the README.DOC because we reserve this file for new items or ones which were accidentally omitted from the manual.

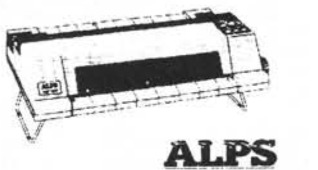
Our tech support department told Mr. Brenner that he could solve the problem and run Disk Technician Advanced on his Heath/Zenith 386 by repartitioning his drive into two partitions of less than 32 mb each. He chose instead to test Disk Technician Advanced on a rented PC XT with a 20 mb drive.

All went well until near the end of the test, when Disk Technician Advanced, as it's designed to, stopped testing after encountering an apparent hardware problem. He again contacted our tech support department, who let him know that the symptoms he described were typical of a controller malfunction. Unfortunately, because of time and financial constraints, Mr. Brenner was unable to follow-through on their suggestions for a solution and complete his testing.

This was too bad, because he was forced to end his review with a rather ambivalent conclusion. One which, I think, neither he nor your readers found very satisfying.

Given the circumstances, Mr. Brenner's assessment of Disk Technician Advanced as "quite capable" was more than fair. However, I must take issue with his limiting the program to use with "older systems." There is no reason why Disk Technician Advanced cannot work on brand new 286 or 386 machiens, as long as they don't use hard disk partitions of over 32 mb each. In fact, several of us here at Prime Solutions own 386s and, naturally, run Disk Technician Advanced regularly.

Continued on Page 33



FROM THE CPU ...

Henry Fale
2618 Pen Circle
Sheboygan, WI 53081-4250
Reprinted with permission from H-SCOOP

CIS — GO ZENITH Problems

I really didn't want to have to get into this subject, but at this time I feel it is imperative to do so. This editorial in part was written late last year and I decided not to print it without good reason. I have been provoked by injustice and snobbishness to the point that I will not keep silent about it.

The whole idea of bulletin boards, I feel, is support, cooperation, exchange of information and data, and building a comradeship between users. Most boards do very well in this respect, but I have a major problem with CIS, especially with GO ZENITH on CIS.

I hardly get on the GO ZENITH area of CIS anymore, mainly because I don't have the time, and I don't care for the attitude on GO ZENITH. CIS is a very powerful board and it has a lot going for it. The ZENITH forum in CompuServe appears to be nice on the outside, but their general attitude is "you can have what you want, but pay me first, and don't you dare share it with anybody."

Since I don't get on CIS very often (I visit dozens of other technical based BBs every month, including the Zenith Data Systems Service Tech. bulletin board and the HUG BBS), customers of mine and H-SCOOP subscribers often send me files. I have found some good stuff on CIS, but I have also found lots of worthless junk. CIS is the slowest board I have ever been on and their file transfer is the slowest I have ever experienced. Our QD.BB will transfer files at the same CIS baud rate at least twice as fast. So what's the idea? To offer support, or be continually holding out your hand for the almighty buck?

Most boards give you files, and some have requirements of so many downloads for so many uploads. CIS stops the clock (not the long distance phone bills) when you upload files, but you do not get credit, say free download time. So what if the clock is stopped. What are you going to get out of it?

So many of the files (and many important Zenith Data Systems related files) I have had sent to me or have seen specifically state that they are for CIS only and they cannot be placed on another BB, given away, sold, etc. That for one, leaves out support for any unfortunate individual who does not have a modem, does not have a local phone line they can access so must pay CIS charges, plus long distance phone charges, and many others. Since one cannot make the disks available or place the files on another BB, these folks are screwed.

I could understand if there was much development time spent on some of the "proprietary files" and CIS needed to recoup their investment. This has not been the case on the files I have examined that stated they are not to be given out in any way to anyone else. **Most have been direct uploads from ZDS sources,** and as far as I can see, since they are copyrighted by Zenith Data Systems, if they are on the board in the first place (probably illegally), then they should be allowed to be shared and be placed on other boards. ZDS owns the copyright, **not CIS and not the Sysop!!!** Most of the files have come from the Zenith Data Systems Tech bulletin board, the HUG bulletin board, or the ZDS Service Technical bulletin board. Many of the copyrighted ones are from the Service board, and the Sysop is not a subscriber. (I am and at around $1,000/year!) ZDS has those files for support of their products. What in the world good would those files do if they were hidden away where nobody could get them, as the Sysop wants?

I was talking to a ZDS attorney the other day and I learned that ZDS is up in arms over this whole CIS business and has been investigating it and the Sysop's antics for some time. I learned that he has absolutely no business having most of those ZDS files on CIS in the first place and he has absolutley no business tacking on his "notice" that CIS owns the files. Most of the files, if examined, have a ZDS copyright right in them. So how does he get off placing them on GO ZENITH and then adding *his* "don't place these files on other boards or give them out" notice. My attitude is ignore the Sysop's notices. If he has them on CIS, then I see no problem with sharing them!

Just before press time, I learned, to my total astonishment, that the Sysop and CIS had the unbelievable audacity to place Microsoft's (ZDS version) MS-DOS 3.3 (looks like all the files minus the BIOS and COMMAND.COM) on the GO ZENITH forum of CIS. As usual, the "don't copy, give out, place on other bulletin board" disclaimer is there. Well Sysop, how come you can violate copyright code and then tell everyone else not to do it? The files illegally placed on CIS are copyrighted by Microsoft and by Zenith Data Systems, not by the Sysop or CompuServe, thank you! Those files have no business being on CompuServe. Here's a sample of what I saw when I visited GO ZENITH on May 16th on the advice of a hot tip about DOS 3.3 being up there:

```
Newest DOS 3.3 Plus utilities in Library 1
The most recent versions of all the MS-DOS 3.3 Plus utility
programs (comprising Version 3.30.14) are in the files DOS33A.ZIP,
DOS33B.ZIP, and DOS 33C.ZIP in Library 1. See DOS33.DIR in
Library 1 for contents of those three distribution files.
```

So I went to New Files libraries and did a BROWSE and here's what I saw:

```
[70007,1401]
DOS33A.ZIP/binary   12-May-90 162808    Accesses: 32
Title   : ZDS MS-DOS 3.3 Plus utilities: Disk 1 (Part 1)
Keywords: ZDS DOS MSDOS 3.3 PLUS UTILITIES UPDATES

Zenith Data Systems' MS-DOS 3.3 Plus utilities (Version 3.30.14),
complete in three files:


DOS33A.ZIP  Disk 1 (Part 1) <--
DOS33B.ZIP  Disk 1 (Part 2)
DOS33C.ZIP  Disk 2

?For the contents of each part, including filenames, dates, and times, see:

DOS33.DIR

From Zenith Data Systems. Not for redistribution. UPL Bill Stuebe
[ZDS].
```

So I chose DOWNLOAD a file option for DOS33.DIR, and here's what I saw:

```
FILE: MS330.DIR
Not for redistribution.

These three ZIP archive files in the Zenith Forum libraries contain
the utility programs for Version 3.30.14 of MS-DOS 3.3 Plus from
Zenith Data Systems:

MS330A.ZIP     Disk 1, Part 1 (A-J)
MS330B.ZIP     Disk 1, Part 2 (K-Z)
MS330C.ZIP     Disk 2
Although together these files are the utilities in the Version 3.30.14
distribution of Zenith Data Systems  MS-DOS 3.3 Plus, each utility
program has its own version sequence. You may check Neither the
ZIP archive files nor any of their contents may be redistributed, on
bulletin board systems or by any other means, even if no fee is
charged."
```

So there we go again, the "don't violate my (Sysop's) Copyright" for stuff he has no right to in the first place bit.

I'll list a few lines of what the file contained, and you can easily see these comprise DOS 3.3:

Contents of File: MS330A.ZIP:

```
Filename     Date       Time
----------   --------   -----
NOTICE
ANSI.SYS     02-02-88   00:00
APPEND.EXE   02-02-88   00:00
APPLY.COM    04-24-89   11:14
ASSIGN.COM   02-02-88   00:00
```

Need I say more? The idea started, I hear, because some files change or have bugs, and upgrades are made available. **Some** files. This is a good idea, as a handful are on my bulletin board, the HUG, the Zenith Data Systems Tech, and others. A good way for folks to get files that are not bug ridden. But to place almost every file comprising DOS there all at once, even though many never had bugs, is another story. And the ones on the boards that have been placed there to solve bug problems, neither the Sysop nor CIS have any control over. Give them away, share them, place them on the boards, and then leave CIS a message that "Henry sent you!"

[A day before press time, I heard that CIS was forced to take those files off. I understand they did, and placed them back on again under another name (I have not verified this) from another person who had uploaded them for CIS. When I checked on the 23rd of May, they were not there, so I guess they got the message, and ZDS is not through with them yet.]

One thing I will say, is our QDBBS will never have a policy like that. We have started placing more files on our board (sorry, you can't see them yet) and I have megabytes more to sift through. We will soon be charging for accessing the files on a yearly rate, not a time rate. We are here to make a living supporting H/Z computers. However, any files, especially files to correct bugs, that you get off of our board are to do with whatever you want. Give them out, share them, copy them . . . whatever. We will eventually compile a large H/Z support file section and when that happens the only restriction that I will have is not to place **my** support files on other boards since I don't want somebody simply to copy the files on a board and make money off of hundreds of hours I will have spent putting it all together. Sound fair?

We are not going to be operating our board to make lots of bucks. I am planning real soon now to add about $5,000 worth of equipment to the board in the area of a new and faster computer with a 300 MB ESDI drive. I also plan to be adding more phone lines in the near future. My software is already set up to support up to 16 callers so I'm all set in that department.

And speaking of boards, I spoke to Jim Buszkiewicz and the female robot over there recently and they have some very exciting plans for their board. They are basically using the same software I am, eSOFT's TBBS which is a very versatile, fast and powerful system. I understand they now have 8 lines coming in on a 386 computer, soon to be expanded to 16 lines,

Continued on Page 36

WordPerfect Macros

Edwin G. Wiggins
13 Clare Drive
E. Northport, NY 11731

Introduction

WordPerfect version 5 offers a rich array of macro tools. These range from simple keystroke macros to command language macro programs. Individual macro files can be collected into a keyboard redefinition file. This saves disk space, and it allows you to have twice as many macros. You can also create temporary macros that are wiped out when you exit WordPerfect.

These features are not thoroughly covered by the various books on using WordPerfect. The WordPerfect manual does cover everything, but the coverage is rather brief, and not many examples are given.

This article covers four topics: simple keystroke macros, keyboard redefinition files, command language macro programs and temporary macros. It should help you get the full benefit of WordPerfect's macro capability.

Keystroke Macros

If you type the same words over and over in various documents, keystroke macros can speed up the process. If you step through the same series of menus repeatedly, keystroke macros can save time and aggravation. Writing a keystroke macro is easy; it involves little more than what you already type.

At the end of every business letter, I type the same complimentary close: Yours truly, (four blank lines), Edwin G. Wiggins. Actually, I don't type that anymore; a macro does it for me. Here's how the macro was created:

<CTRL-F10>	turns on the macro recorder.

The prompt at the bottom of the screen says "Define macro:". That's a little bit cryptic, but it means "Tell me what key combination should execute the macro." I respond <ALT-C>, which means that I want the macro I am creating to execute when I hold down the ALT key and press the C key.

The prompt at the bottom of the screen changes to "Description:". Entering a description is optional. You can simply press <ENTER> to bypass it, but I recommend that you enter one. It will help you remember what the macro does. I type "COMPLIMENTARY CLOSE" and press <ENTER>.

Now the prompt at the bottom of the screen flashes "Macro Def". From now until I press <CTRL-F10> again, the macro recorder will remember every key I press. I type "Yours truly," followed by <ENTER> five times, followed by "Edwin G. Wiggins". Pressing <CTRL-F10> the second time turns off the macro recorder and stores the keystrokes. From now on I can put the complimentary close on my letters by pressing just <ALT-C>.

Since I'm an engineer, I have occasion to type superscripts and subscripts frequently. Apparently, WordPerfect Corp. found that few users do this, because you must move through several menu levels to turn super- or subscripting on and off. The following keystroke macro subscripts the character to the left of the cursor:

<CTRL-F10>	turns the macro recorder on.
<ALT-D>	selects the macro execution key.
SUBSCRIPT	is the description of the macro.
<Left>	left arrow key moves cursor back one character.
<ALT-F4>	turns the block marker on.
<Right>	right arrow key moves cursor ahead one character, marking a one character block.
<CTRL-F8>	brings up the font menu.
1	selects "size."
2	selects "subscript."
<CTRL-F10>	turns the macro recorder off and stores the keystrokes.

If I want to type the chemical formula for water, I type: H2<ALT-D>O. The <ALT-D> makes the 2 a subscript.

The macro for superscript is assigned to key <ALT-U>. It is the same as the subscript macro except that the 2 above is replaced by 1.

Macros that are assigned to a combination of <ALT> and a single letter will execute immediately when that key combination is pressed. With this system, you can create 26 macros. This may seem like a lot, but once you're into macros it may not be enough. You can also assign a macro to a word. These macros take more keystrokes to execute, but it increases the number of macros you can create, and word macros are easier to remember.

Since I teach at Webb Institute of Naval Architecture, I used to type that name a lot. Now I have a macro assigned to the word WEBB that types the whole name for me. Creation of the macro goes like this:

<CTRL-F10>	turns the macro recorder on.
WEBB	assigns the macro to the word WEBB.
Webb Institute	describes the macro.
Webb Institute of Naval Architecture	is the macro itself.
<CTRL-F10>	turns the macro recorder off and stores the keystrokes.

In order to execute this macro, I press <ALT-F10>. The prompt at the bottom of the screen says "Macro:". I respond WEBB and press <ENTER>. The full name of the school appears in my document.

These examples illustrate a few of the things you can do with keystroke macros. Examine your work patterns for ways that macros can save you time.

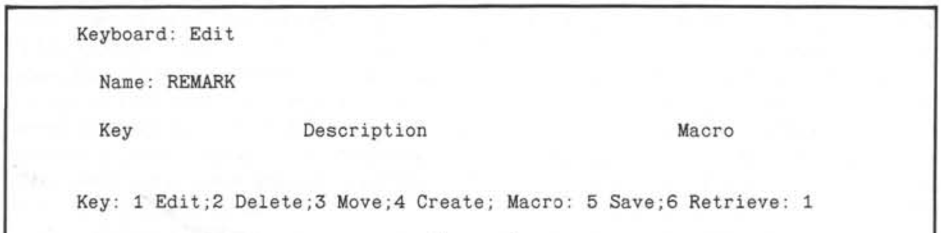
Keyboard Redefinition

Each keystroke macro that you create is a separate file. Even though each file may be only a few bytes long, it occupies a certain minimum number of clusters on your hard disk. Therefore, macro files can eat up a lot of disk space. This problem can be reduced through the use of a keyboard redefinition file. In addition, keyboard redefinition files allow you to assign immediate execute macros to key combinations other than <ALT> plus a letter.

Plain and Fancy

You can create a keyboard redefinition file from scratch, or you can assemble it from existing macros. You'll find a description of the latter approach below. If you assemble the file from existing macros, the macro files themselves remain on the disk, and the information they contain is copied into the keyboard redefinition file. When you're done, you probably want to delete the macro files.

To create a keyboard redefinition file press <SHIFT-F1> to call up the SETUP menu. Select 6 for keyboard layout. Select 4 for create. The prompt at the bottom of the screen says "Keyboard filename:". Type the name you want to give to your redefinition file and press <RETURN>. The following screen appears:



```
Keyboard: Edit

  Name: REMARK

  Key                 Description                          Macro


Key: 1 Edit;2 Delete;3 Move;4 Create; Macro: 5 Save;6 Retrieve: 1
```

Figure 1
Keyboard Edit Screen

The cursor will appear near the top of the column under the "Key" heading. To retrieve existing macros into the keyboard redefinition file named REMARK, press 6. At the bottom of the screen the prompt will say "KEY:". Type the key whose action you want to redefine. With a few exceptions, you can redefine the action of any key on the keyboard. You could redefine the A key to produce the letter Z if you wished. Normally though, you want to redefine the action of <ALT-key> or <CTRL-key> combinations.

In this example, I will redefine the action of the <ALT-Q> combination, so I respond to the "KEY:" prompt by pressing <ALT-Q>. The prompt changes to "MACRO:". Here I type the name of the existing macro that is to be copied into the keyboard redefinition file. In this case, the macro is also named <ALT-Q>, so I type <ALT-Q> again. Information from the macro is automatically copied into the keyboard redefinition file, and the appropriate entries appear in the "KEY", "DESCRIPTION" and "MACRO" columns in Figure 1. This process may be repeated to copy other macros into the redefinition file. When I'm done, I press F7 twice to return to the main editing screen.

Although the macro <ALT-Q> will remain on disk, I can not execute or edit it if the keystroke redefinition file REMARK is active. Anytime I press <ALT-Q> the keystrokes are intercepted by the keystroke redefinition file. To activate or deactivate a redefinition file, press <SHIFT-F1> to display the SETUP screen, and 6 to display the keyboard redefinition selection screen. Highlight the desired redefinition file name and press 1 to activate it. Press 6 to deactivate all keyboard redefinitions.

Command Language Macros

WordPerfect comes with a powerful, but not very friendly, macro programming language. There's virtually no limit to what you can do with this language, but it takes some time to master it. Complete coverage of this feature would take a series of articles. I'll just provide an introduction and a simple example here.

The macro programming language includes IF...ELSE, CASE and GO statements. There are variables that can take on numerical or string values. Custom prompts can be displayed on the screen, and the computer can be made to BEEP to alert the user.

There are two major sources of confusion and programming errors. First, where a tilde (~) appears in a command syntax, make sure to type it in. Failing to include a tilde produces all sorts of strange behavior when the macro runs.

The second source of confusion is between the name of a variable and its value. Variables are named {VAR 0} through {VAR 9}, but they are invoked by typing <ALT-number>. Thus, if you hold down the ALT key and press the 1 key, the contents of {VAR 1} will be typed onto the screen. If you're on the editing screen, the variable contents will be typed as text. If you're in a menu, the variable contents will be treated as menu selections.

If you're writing a macro program, and you want to insert the name (not the value) of a variable in the program, type <CTRL-V> followed by <ALT-number>.

I normally keep document files on a floppy disk in drive A:, so when I start WordPerfect, I want to change the default drive to A: and display its root directory. Since I sometimes forget to insert the disk, I want the computer to remind me. I wrote the following macro to accomplish these things:

```
{PROMPT}INSERT DISK IN DRIVE A:~
{BELL}
{PAUSE}
{List Files}=A:{Enter}{Enter}
```

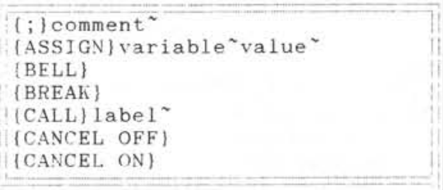
The first command displays a prompt in the lower-left corner of the screen. The second sounds a beep, and the third causes macro execution to pause until I press <ENTER>. The last command changes the default drive to A: and displays its root directory.

The macro editor must be used to create the above macro; the commands in braces can not be typed from the keyboard. WordPerfect's built-in macro editor can edit existing macros, but it can't create new ones, so first a dummy keystroke macro must be created. The initial contents of the macro do not matter since they will be replaced. A blank space will do. Once the dummy macro exists, proceed as follows: Press <CTRL-F10>. Respond to the "Define macro:" prompt by typing the name of the dummy macro. The following prompt appears:

```
ALTQ.WPM is Already Defined.
    1 Replace; 2 Edit: Ø
```

Type 2. The macro editing screen will appear with the dummy macro displayed on it. Press 2 to edit the action of the dummy macro, and delete the original contents.

Now we're ready to begin creating the command macro listed above. Press <CTRL-PgUp> to display the command menu displayed below.



```
{;}comment~
{ASSIGN}variable~value~
{BELL}
{BREAK}
{CALL}label~
{CANCEL OFF}
{CANCEL ON}
```

Figure 2
Command Menu

There are more commands than the window displays. Pressing the down arrow key will move the cursor down the window, and when it reaches the bottom, additional commands will scroll into view. To move more quickly, begin typing the letters that make up the command. The cursor will automatically seek a match to what you type.

Move the cursor to the {PROMPT} command and press <ENTER>. The prompt command will automatically be transferred to the macro action window. (It does not work to simply type "{PROMPT}" in the macro action window!) With {PROMPT} displayed in the macro action window and the usual flashing cursor immediately to the right of the command, type "INSERT DISK IN DRIVE A:~". Raised dots appear when you press the space bar. Don't forget the tilde! Press <ENTER>. Press <CTRL-PgUp> to shift the cursor back to the command window, move it to the command {BELL} and press <ENTER>. Repeat the process for the command {PAUSE}. Repeat the process again for the {List Files} command. When this command is displayed on the macro action window with the flashing cursor next to it, type "=A:". The two

Continued on Page 36

Add MIDI to Your SupersPort

Terry Perdue
3646 Crestview Drive
St. Joseph, MI 49085

This article is for anyone who has an interest in using their SupersPort laptop with MIDI-equipped musical instruments.

MIDI, the Musical Instrument Digital Interface, was described by T. E. Thompson in the January and February 1988 issues of REMark, and magazines such as *Keyboard* and *Electronic Musician* often feature articles on the subject. It is an opto-isolated 31.25 kilobaud serial 5 mA current loop used to allow all kinds of musical instruments to communicate with each other and with sequencers, computers, and other devices. The music industry uses MIDI extensively, and many of the electronic musical instruments sold today come with this interface built in.

To interface a MIDI compatible instrument with a PC compatible computer that has an unused card slot, you can purchase a Roland MPU-401 interface card, or one of several clones that use the Roland chip. These half-wide cards feature both 'dumb' and 'smart' modes of operation. In the dumb mode, the card simply passes data back and forth between the host computer and the MIDI port. The smart mode allows the card to take over the timing functions of sequencer software, reducing overhead on the PC. Most of the popular MIDI programs on the market take advantage of this mode.

If you own a keyboard instrument with MIDI, or a MIDI 'expander', which is the instrument without the keyboard, you may find yourself in my situation. My musical instruments (a Kurzweil expander and a MIDI-compatible organ) are in the living room, while my desktop computer is in a spare bedroom. It's not at all convenient to carry the computer back and forth. However, I also have a SupersPort 286, and I realized that its size would make it ideal for this application, since it's small enough to sit next to me on the organ bench.

I decided that I wanted to develop my own MIDI software, and run it on the laptop. I also decided to try to avoid the extra expense of an MPU-401 card and the expansion chassis that would be required to use it with the laptop.

The simplest solution seemed at first to be some kind of outboard box that would plug into both the RS-232 serial port on the laptop and the MIDI input and output ports on the musical instrument. The rub is MIDI's 31.25 kilobaud rate. The MIDI Association apparently chose this rate because it is easily derived from a 1 MHz clock, but it can't be derived from the standard baud rate clock used in the SupersPort. So this approach would end up requiring two UARTs and two baud rate oscillators, in addition to the opto-isolation and EIA level conversion — a bit more than I had counted on.

It finally dawned on me that a MIDI interface card could be easily designed to plug into the unused modem slot in the side of my SupersPort, and made operational by using the laptop's SETUP menu to enable the 'modem'.

The interface was laid out as a single-sided board to allow easy duplication. You can use any of the usual methods to fabricate the board, but be careful of the few narrow foils, which may get etched off if you don't do everything just right. If you wish, you can leave those foils off, and just replace them with point-to-point wiring, using magnet wire or wirewrap wire, after the board is assembled.

Before assembling the DIN connectors to the circuit board, they must be modified by removing the front metal shield plate, and filing or sanding the square tops to conform to the rounded body of the connector. Otherwise, the connectors will be too tall to fit into the modem compartment. Also file off the step on the bottom of both connectors so that they will sit as low on the circuit board as possible. Remove the ground spring on the connector used at P1 **only**.

The modem compartment's cover plate also must be modified. Disassemble the cover plate and remove the plugs that fill the two rectangular holes in the plate. Using a rat-tail file, carefully enlarge the holes to accept the standard 5-pin DIN plugs that are used for making MIDI connections. The board was laid out so that the DIN connectors will line up with the original openings. I suggest that you not modify the cover plate until you have assembled and installed the interface so that you don't enlarge the holes more

C1	0.1 uF ceramic
C2	56 pF ceramic or mica
C3	27 pF ceramic or mica
D1	1N4149 general purpose silicon diode
R1-R7	1/4 Watt, 10%
P1	Hirose PCN13-32S-2.54DS (Digi-Key H3032)
P2, P3	AMP 212044-1 (Digi-Key A1663)
U1	8250B Asynchronous Communications Element
U2	74HCT00 quad NAND gate (Digi-Key CD74HCT00E)
U3	Sharp PC-900 optoisolator (or equiv.)
Y1	2.0 MHz crystal (Digi-Key X068)

Figure 1
Parts List

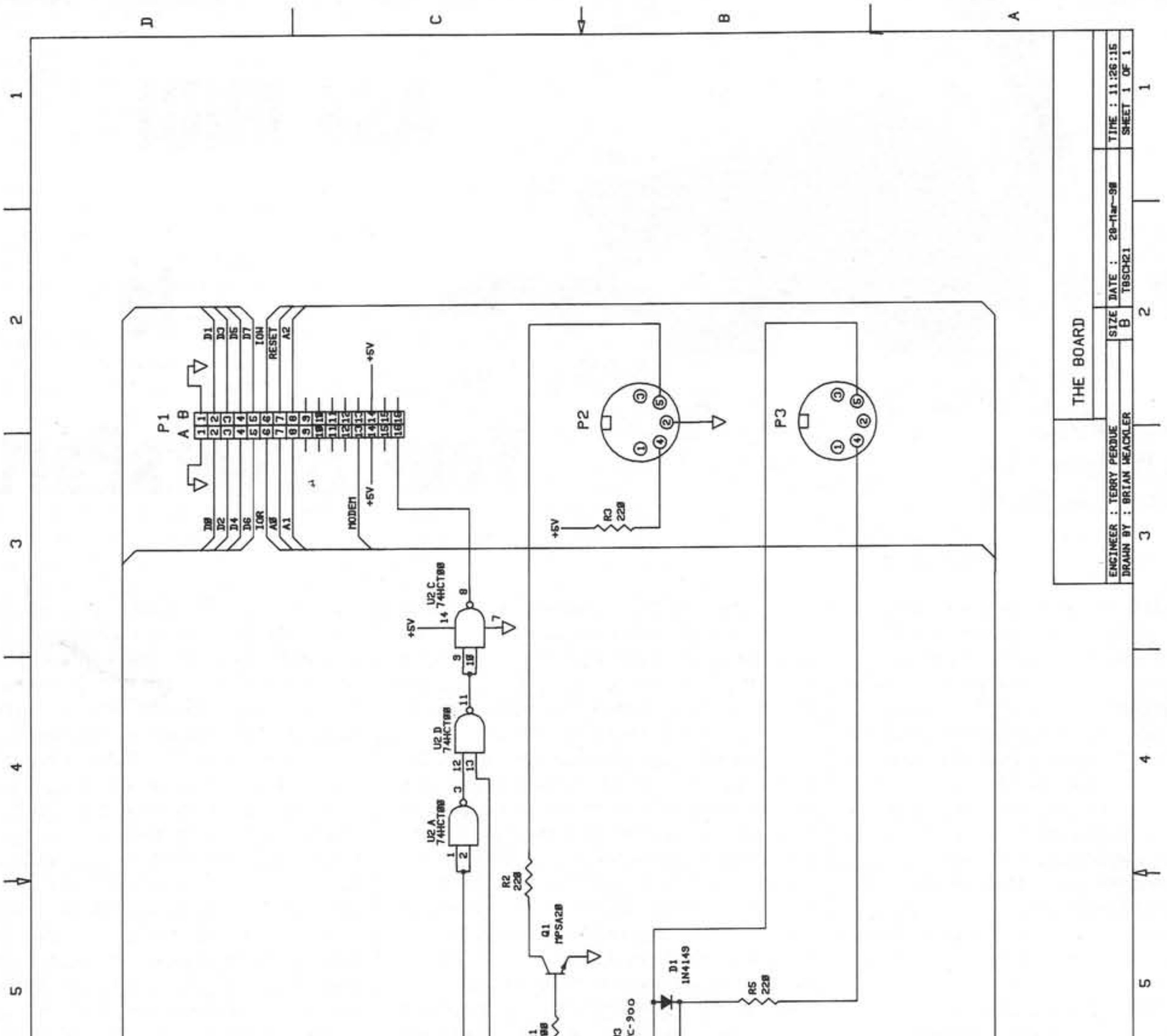
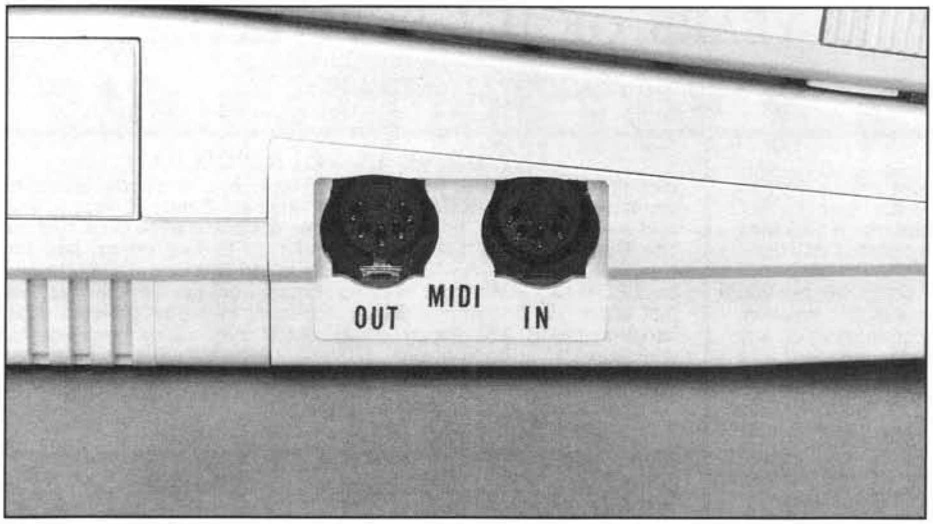


Figure 2
Schematic for the Laptop MIDI Interface



than necessary. You will also need to remove about 3/16" off the top of the vertical rib on the inside of the cover to make room for the board. You can just clip it off with a pair of diagonal cutters.

Dry-transfer lettering can be applied to identify the input and output ports, although accidentally reversing the cables cannot damage anything. First, with the tip of a sharp knife, carefully scrape off the old legends. Then press on the new letters, and apply a light coating of clear fingernail polish to protect them.

Make sure that the cables you use are intended for MIDI. Purchase preassembled MIDI cables at a music store, or make your own from 5-pin DIN plugs sold at Radio Shack, but don't assume that a cable with a 5-pin DIN plug on each end is a MIDI cable. If it's not, it probably won't damage anything, but it probably won't work either, or if it does, it may cause hum.

Since this interface is accessed differently than the MPU-401, and only operates in the dumb UART mode, commercial programs won't work with it. However, it is a relatively easy job to write load/dump and control programs for it. Even record/playback and sequencer programs may be written that will work satisfactorily when used with a single instrument.

Note that your program must program U1 to divide the 2 MHz clock by 4. This produces an internal sampling clock of 500 kHz, which is 16 times the required baud rate.

For the popular Kurzweil 1000 series expanders, complete control and programming functions are possible. Kurzweil sells a program called Object Mover™, that allows the expanders to be programmed much more easily than they can be from the front panel. I believe it would be reasonably easy to patch this program to operate with the laptop interface described here, and I hope to try that soon.

In the meantime, I've written a load/dump/control program for my Kurzweil 1000PX+ expander. While written specifically for this model, it should work with any expander in the series, and would provide a basis for similar programs for other manufacturers' models. A copy of the program (assembly source, .COM file, and documentation) is available from the author for $10. Please specify whether you prefer a 5.25" or 3.5" disk.

My thanks to Brian Weackler for laying out the circuit board. You may be interested in reading his article on circuit board layout techniques in the April 1990 issue of **Heath Electronics**.

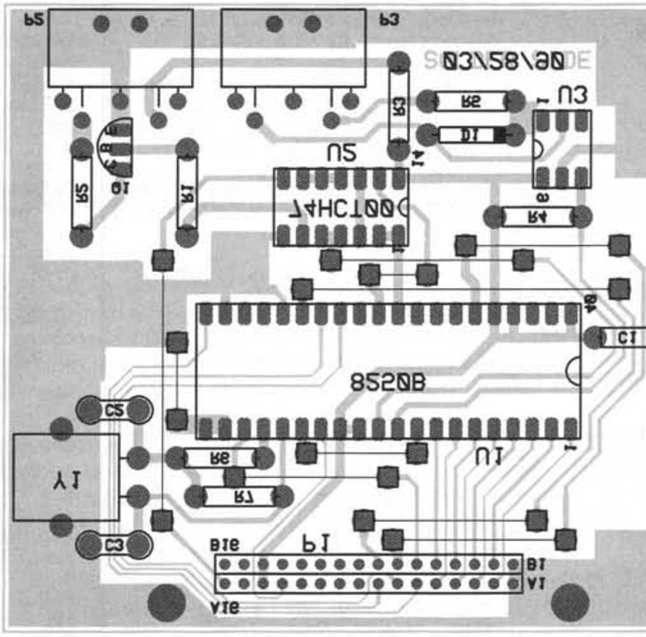


Figure 3
Foil pattern from the solder side of the board (1:1). (The circuit board measures 3.25" × 3.60".)

QUIKDATA - 13 YEARS OF H/Z SUPPORT!

For all your H/Z 8-bit and PC/XT/AT needs

ACCELERATE YOUR PC!

From Sota Technologies, Inc., the fastest and most proven way to breath new life in your Heath/Zenith PC/XT computer, giving it -AT compatible speeds! Turn your turtle into a -286 rabbit with a 12.5 Mhz 80286 accelerator board. Complete with 16K on-board CACHE.

There are other ways to speed up your H/Z PC/XT computer, but you spend too much and get too little. The **286i** is the effective solution, making your H/Z150/160/150/158/159 series of computers, or any general PC/XT computer faster, in many cases, than a standard IBM AT type computer! Simple half-card plug-in installation with switchable speed control. **You won't believe your stop watch!**
EXP-12 - **$295**
EXP386 - **$395** Much faster 16Mhz 80386 SX version

MEMORY UPGRADES

Note: All memory upgrades come without memory chips. Call for current chip pricing. Memory chip prices are down. Example: 256K 150ns DRAM is $2.25 as of this printing.
Z150MP - $17 Will allow you to upgrade your H/Z150/160 to up to 704K on the main memory board, using up to 18 256K DRAM chips.

MEGARAM - $43 Upgrades your H/Z150160 series with up to 704K of main memory, and about 512K for RAMDRIVE memory. Includes documentation, software RAMDRIVE disk, PAL and jumper wire. For the full 1.2 megs total memory, 45 256K DRAM chips are required.

ZMF100 - $53 Will allow you to upgrade your H/Z110/120 (old motherboards; with p/n less than 181-4918) to 768K system RAM. Requires 27 256K DRAM chips.

Z100MP - $76 Similar to ZMF100 above, but for new motherboards with p/n 181-4918 or greater.

Z159 EMS LOGIC UPGRADE includes the PAL chip, logic chips and one bank of 256K (9) chips. Up to two banks (18) can be installed for a total system RAM of 1.2 meg on the Z159 main board.
Z315-1 - $79!!

WINCHESTER UPGRADE KITS

PCW20 - $269 Complete winchester setup for a H/Z150, 148, 158, 159, 160, PC etc. Includes 21 meg formatted half-height Segate ST-225 65ms drive, WX1 controller, cable set, doc.

PCW30 - $339 32 meg with 38ms Segate ST-138.
PCW40 - $379 42 meg with 28ms Segate ST-251-1.
PCW80 - $629 80 meg with Segate ST-4096 full size drive.

We also have the DTC controllers ($69) and daughter board expansions ($69) to place a hard drive in the **H/Z148** computers

BARE WINCHESTER DRIVES

ST-125 - $239 21 meg 37ms autopark 3.5" drive in 5" frame
ST-225 - $219 21 meg 65ms HH 5" drive
ST-138 - $275 32 meg 38ms autopark 3.5" drive in 5" frame
ST-251-1 - $319 42 meg 28ms autopark HH drive
ST-4096 - $579 80 meg 28ms autopark FH drive
ESDI - $949 **160MB ESDI** drives - several types - call or write

FLOPPY DRIVES

MF501 - $75 5" 360K DS/DD drive
MF504 - $83 96 TPI 1.2 meg AT/Z100 drive
MF353 - $85 720K 3.5" drive in 5" frame
MF355 - $95 1.4 meg 3.5" AT drive in 5" frame
SIEM-R - $39 40 track SS refurb (H8/H89 type)
TM100-4R - $75 90 trk DS refurb (H8/H89 type)

DOES YOUR COMPUTER KNOW THE TIME?

We carry the SMART CLOCK for any PC/XT computer, including the Z100 series. Simply plug it in the system ROM socket, and plug the ROM into the clock module. Self contained sealed lithium battery for 5-10 year life. Includes software for the clock/calendar.
SMARTCLK - $33 (add $2 for Z100 installation for spacers)

ANY DRIVE IN YOUR PC/XT/AT

With the CompatiCard, you can install up to four additional drives, of any type in your PC/XT/AT computer. Add a 1.2 meg 5" floppy, or a 1.44 meg 3.5" floppy, or any other drive, including 8" to your system. The CompatiCard (CCARD) will handle up to four drives, and the CompatiCard II (CCARD2) will handle up to 2 drives. CCARD4 has boot ROM to allow it to be used as primary boot controller in systems that allow you to remove floppy controller in some systems. Also handles 2.8MB 3.5" floppy drives. Additional cables and external enclosures may be required.
CCARD2 - $79
CCARD - $99
CCARD4 - $125

OUR QD LINE OF COMPUTERS

Excellent quality bare bones PC/XT/AT 0-wait state clone systems with 101 keyboard and one year warranty. Buy the basic system and choose from our product line to add what you need for a complete system.

QD88/12 is our high-powered entry level system with 640K RAM, keyboard, 360K floppy drive, 12Mhz V20 CPU, co-processor socket, clock/calendar, 4 bay cabinet with turbo/reset button and keylock, 150 watt switching power supply, 8 card expansion slot, parallel port, serial port, game port. Add TTL video monitor and go, or add hard drive, additional floppy drives, video, etc. $495

QD286/12 is a solid 12Mhz 80286 AT compatible work-horse with a 200 watt switching supply, 1 meg RAM expandable to 4 megs on motherboard, 80287 socket, clock/calendar, 5 bay cabinet with turbo/reset button and keylock, 8 card expansion slot. Add controller, ports, video, drives and go. $649

QD386/25 is a fast top of the line 25Mhz 80386 system with the same specs as 286/12 except expandable to 16 megs on main board, 80387 or Weitek socket. $1295

8-BIT/Z100

We carry a full line of replacement boards, parts and power supplies for the H/Z89/90 and Z100. We also have some H8 boards available. We continue to fully support and carry a full line of hardware and software products for the H8/H89/90 and Z100 computers. This includes diskettes, printers, modems, hard drives, etc. Here's a sample.

H37 SOFT SECTOR CONTROLLER	$169
H8 SOFT SECTOR/WINNIE CONTROLLER PACKAGE	$195
DISKPACK for Z100 Allows reading and writing of PC formats including 1.2 and 1.4 meg on Z100.	$39
MSDOS 3.1 FOR Z100	$89

OTHER STUFF

Quikdata also carries spike protection filters, backup power supplies, modems, printers, disk drives, drive enclosures, cables and connectors, laptop batteries, video monitors and video cards, memory cards, memory chips and ICs, joysticks, accessory cards, serial and bus mouse, all parts for H/Z89/90 and H/Z100 computers, some H8 parts, accessories, a variety of useful and most popular software and much more!

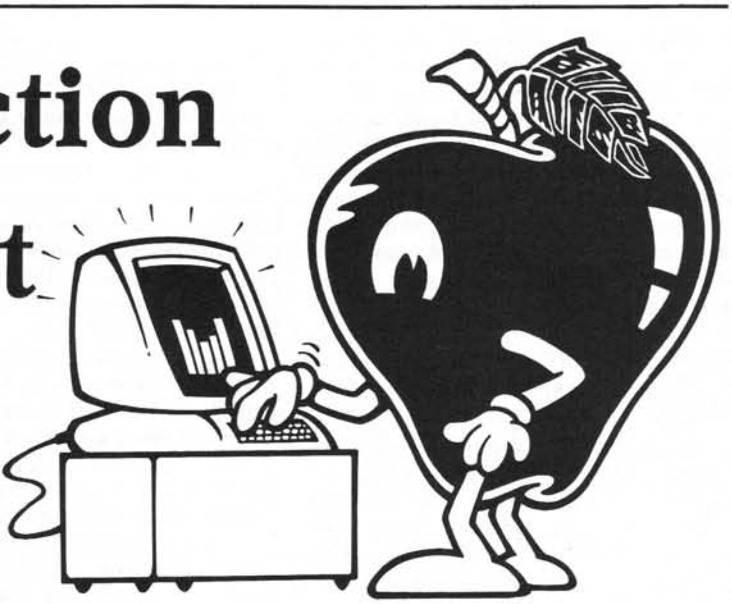
Call or write in to place your order, inquire about any products, or request a free no obligation catalog. VISA and Master Card accepted, pick up 2% S&H. We also ship UPS COD and accept purchase orders to rated firms (add 5% to all items for POs). All orders under $100 add $4 S&H. Phone hours: 9AM-4:30PM Mon-Thu, 9AM-3PM Friday. Visit our bulletin board: (414) 452-4345. FAX: (414) 452-4344.

QUIKDATA, INC.

2618 PENN CIRCLE
SHEBOYGAN, WI 53081-4250
(414) 452-4172

Reader Service #193

An Introduction to Postscript



Alan Neibauer
11138 Hendrix Street
Philadelphia, PA 19116

For the most part, laser printers are divided into two types — the Hewlett-Packard LaserJet (and those compatible with it) and Postscript.

(To get the legal business out of the way, Postscript is a registered trademark of Adobe Systems, Inc.).

Just a few years ago, this division strictly paralleled the two major camps in the PC world. If you were using a laser printer with an IBM compatible computer, you owned a LaserJet. If you had an Apple computer, you were using a Postscript printer.

DOS users tended to know nothing about Postscript. If you look through some old PC magazines, you'll probably find very few articles on how to use Postscript. The same was true with Apple users. To them, the LaserJet was alien, a supposedly less sophisticated, less expensive, form of laser printing.

Not only was it rare for the two worlds to meet, but proponents off each side formed into vocal, often hostile, camps. It was common to see letters to the editor and magazine articles denigrating the other point-of-view. It seemed that comparing the two standards feature-by-feature, or through benchmarks on printing speed, was pulling the two sides further apart. No summit meetings for these cold war foes.

Well, times have certainly changed. Thanks to improved printer drivers, utility programs, and add-on hardware, either type of laser printer can be connected to any computer.

The two camps are not only talking, but sharing hardware and software, ideas and technology.

There are now many ways to use or implement Postscript in the DOS world. In addition to actually having a Postscript printer, you can access the benefits of Postscript through software or hardware on a LaserJet, and even on some dot matrix printers, as well.

So now that it is no longer an act of treason to talk about Postscript in a magazine devoted largely to DOS, let's examine what Postscript is and why so many people are talking about it. (In a future article, I'll outline specific ways to implement PostScript on non-Postscript printers.)

In this article, I'll first describe the features of Postscript that make is so desirable. I'll also describe the Postscript language and give examples of its commands and uses. I'll then explain the three ways to create Postscript programs.

Postscript Features

Two standard sets of fonts are usually supplied with Postscript. Printers compatible with the original Apple LaserWriter have 13 fonts: a symbol set of Greek characters, and Courier, Times, and Helvetica in normal, bold, italic, and bold italic. LaserWriter Plus compatible printers have 35 fonts, those 13 plus Zapf Chancery Medium Italic, Zapf Dingbats, and four fonts each of Palatine, ITC Avant Garde, ITC Bookman, Helvetica ·Narrow, and New Century Schoolbook.

The fonts are supplied as outlines, basic patterns that can be enlarged to any size and shaped as needed. When you want to print in a particular font and size, your software's printer driver (or your own program) transmits to the printer the commands necessary to create the characters "on the fly." This means that specifically sized and formatted characters are made at the time of printing. The use of outline fonts is what gives Postscript printers their traditional "advantage" over the LaserJet models.

I said "traditional" because LaserJet-compatible printers can now be outfitted with many of the same features that were once only claimed by Postscript. But let's look at the three benefits that Postscript supporters always point to with pride.

First is fonts on the fly. Using a Postscript printer, you only need one font outline for each style of a typeface. The printer can create characters of any size from the basic outline. Using a LaserJet, you must create all of the fonts you want to use ahead of time — and find disk storage for them. If you create 6, 10, 12, 14, 18, and 24 point fonts, those are the only sizes you can print.

Second, Postscript outline fonts can not only be sized, but rotated, stretched, compressed, or filled, creating a wide variety of special graphic and text effects.

Third, you can include portrait and landscape orientations on the same page.

The only downside to these benefits is speed. Creating the fonts on the fly, called rasterizing, can take a long time. Slow speed, in fact, is the most common complaint against Postscript.

The PostScript Language

Postscript is actually a set of commands that control the actions of the printer, much like a programming language controls the actions of the computer. The commands are transmitted to the printer as an ASCII text file.

The Postscript printer interprets these commands, translating the actions into a printed image. In many ways, Postscript is similar to Forth. It is a stack oriented language utilizing postfix notation. And like all programming languages, it offers a full complement of structures, including loops, conditionals, arrays, procedures, and even the ability to create library files of subroutines.

Postscript programs create a path, a

set of curved or straight lines that make up graphic and character shapes. The lines of the path can then be stroked on the page, drawn as with a plotter pen, or used to define an area to be filled in.

Paths are drawn using an X-Y coordinate system with movements in 1/72's of an inch, or points.

For example, this simple program draws a 4 inch horizontal line across the center of the page:

```
newpath
162 396 moveto
396 45Ø lineto
stroke
showpage
```

The *newpath* command tells the printer to start a new path or image. The second line, *162 396 moveto*, places two numbers on top of the stack, then executes the moveto command. Moveto sets the path's point of origin, using the two numbers at the top of the stack as the X and Y coordinates relative to the lower-left corner of the page. In this case, the path starts 2-1/4 inches (162/72) from the right, and 5-1/2 inches (396/72) from the bottom of the pages.

The next line, *396 450 lineto*, defines a line from the current point to the coordinates now on top of the stack, 396 and 450. This is a point 6-1/4 inches along the X axis, or a line 4 inches long. *Stroke* "paints" the path, drawing the line defined by the path. Finally, *showpage* prints the page. The program started a new path, defined the lines that make up the path, then stroked or drew the lines.

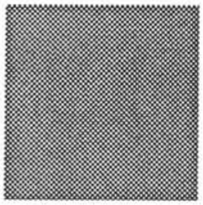


Figure 1

Now look at this program that prints a one-inch square with a 50% dot, or fill pattern, as shown in Figure 1:

```
newpath
288 36Ø moveto
Ø 72 rlineto
72 Ø rlineto
Ø -72 rlineto
closepath
.5 setgray
fill
showpage
```

The program starts by setting a new path, then moving the point of origin to 4 inches from the right and 5 inches from the bottom of the page. The next three program lines form the left, top, and right sides of the area to be filled. They use the rlineto command that sets the lines of the path relative from the current point, using the numbers on the top of the stack as the X and Y distances.

The line *0 72 rlineto* places the numbers 0 and 72 on the stack, then defines a line from the current point (288 and 360) to a point 0 increments to the right on the X axis and 72 increments up on the Y axis. This is the left side of the filled area. A second path line is created with the command *72 0 rlineto*. It places 72 and 0 on the stack, then sets a line to a point 72 points to the right on the X axis and 0 points on the Y axis, forming the top of the filled area.

The right side of the filled area is defined by the next line, 72 points down on the Y axis. Closepath defines the final line of the part, the bottom of the filled area. This command defines a line from the current point to the point of origin, the position set with the moveto command.

Now that the four boundaries are defined, *.5 setgray* establishes a 50% fill pattern and *fill* "paints" the area defined by the path. The lines of the path, however, are not drawn by the stroke command, the region is just filled.

Printing Characters

You print characters in Postscript by first placing the name of the font on top of the stack, then setting the size factor. For example, here is a program that prints REMARK in 32 point, Times Roman:

```
/Times Roman findfont
32 scalefont
setfont
162 396 moveto
(REMARK) show
showpage
```

The first three lines place the name of the font on top of the stack, sets the size at 32 points, then makes the font the current font for characters. The point of origin is set at 162 points from the right and 396 points from the bottom of the page.

The text to be printed — REMARK — is enclosed in parentheses, designating a string on top of the stack. The show command prints the string on top of the stack starting at the current point.

Remember that only an outline of Times Roman must be available, not completing created fonts. With fonts on the fly, the final characters can be shaped in many ways. For example, this program prints lines in portrait and landscape:

```
/Times-Italic findfont
32 scalefont setfont
3ØØ 4ØØ moveto
(Portrait) show
27Ø rotate
(Landscape) show
showpage
```

After the first line is printed, the orientation is rotated 270 degrees. Then, a second string is placed in the stack and printed. Figure 2 shows the program's output on a LaserJet Plus.

This next program goes a little further by printing four characters at different rotations (Figure 3):

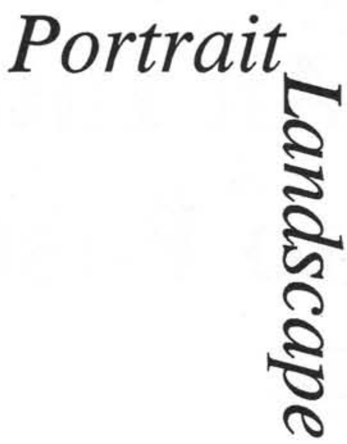


Figure 2

```
/inch {72 mul} def
/Times-Roman findfont 1.Ø inch scalefont
                                  setfont
2.5 inch 5.5 inch moveto Ø rotate (P) show
45 rotate (O) show
18Ø rotate (S) show
27Ø rotate (T) show
showpage
```

The first line creates a Postscript procedure called Inch. The procedure multiplies the number on top of the stack by 72, a convenient way of using measurements in inches. After the procedure is defined, the command *2 inch 2 inch moveto* is equivalent to *144 144 moveto*. After the point of origin is set, each of the four characters is printed at increasing rotations.

In addition to rotating characters, you can scale them, as well. The scale command uses the top two numbers on the stack as the X and Y scale factors. Factors of 2 and .5, for instance, would print characters half as high, but twice as wide as normal, as shown in Figure 4. The word on top was printed using 32 point, Times Roman characters at their normal scale. The word on the bottom was printed with these lines:

```
2 .5 scale
(Remark) show
```

Scaling also works with graphics, as well. For instance, the two shapes in Figure 5 were created with this program:

```
/square
{newpath
Ø,Ø,moveto
Ø,72,rlineto
72,Ø,rlineto
Ø;-72,rlineto
closepath
fill} def
square
15Ø 15Ø translate
square
showpage
```

The procedure *Square* is basically the program you saw previously. The procedure is called once to print the first square, then the coordinate system shifted with the translate command. This changes the system so some position oth-

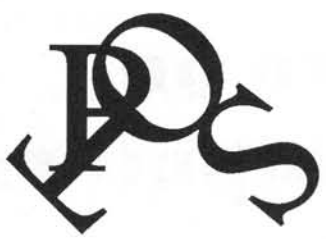


Figure 3

er than the lower-left corner is used as the point of reference. The scale factor is set at .5 and 2.5, then the square procedure called a second time.

Program Structures

Like any programming language, Postscript has a full range of structures. Unfortunately, we'd need an entire issue of REMark to do justice to the these structures and the other Postscript commands necessary for using them. Instead, let's look at two examples of how they can be used.

The following program uses the For repetition command to create the three dimensional effect shown in Figure 6:

```
/Times-Italic findfont 30 scalefont setfont
/printRemark
{ 0 0 moveto (Remark) show} def
320 400 translate
{setgray printRemark -1 .5 translate} for
1 setgray printRemark
showpage
```

After setting the current font to 30 point, Times Roman Italic, this program defines a procedure called *printRemark* in the lines:

```
/printRemark
{ 0 0 moveto (Remark) show} def
```

This procedure leaves the point of origin at the current location, then prints the word Remark.

The For loop itself is controlled by these two lines:

```
{setgray printRemark -1 .5 translate} for
```

The three numbers placed on the stack represent the loops starting value, the step value (amount of increment), and ending value. So this loop starts with a setgray fill value of .95, and reduces the fill value by .05 with each repetition until it reaches a 0 fill level.

With each repetition, the point of origin is shifted with the translate command slightly to the left and up the page, creating the three-dimensional effect.

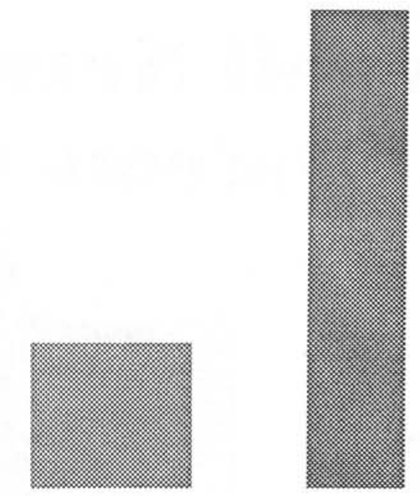


Figure 5

Finally, Figure 7 shows the effects of clipping. Clipping defines the path of one image based on another. In this case, a series of radiated lines have been clipped to conform to the path of the word Remark in 32 point, Times Roman Bold outline characters. The fill region of the lines is limited to the path occupied by the characters. Clipping is one method in which special graphic and text effects can be created.

The Preamble

Postscript programs may contain two parts — the preamble (sometimes called the prologue) and the script. The preamble is transmitted to the printer first and it contains definitions and procedures used by the application. It is similar to constant, variable, function, and procedure definitions that precede a Pascal program, or a library file of functions merged into a C language program.

The script contains the program lines for the specific document or publication being printed.

Every application program has its own unique preamble. The preamble used by Xerox Ventura Publishers, for example, would not work with Microsoft Word or Aldus Pagemaker. But the application's preamble is the same for every document or publication that it prints.

In some cases, the preamble is a separate disk file. For instance, Microsoft Word includes a file called POSTSCRP.INI that is transmitted to the printer before every script. The definitions in that file are stored by the Postscript interpreter and used to convert the script for a specific document into bitmaps.

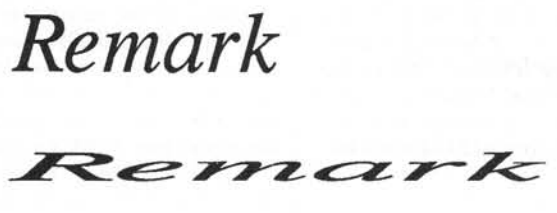


Figure 4

With these types of programs, the preamble file must be present on the disk for any document to print on a Postscript device. The application will display a warning message if the preamble file isn't available, since the definitions and procedures used in the script will be meaningless to the Postscript interpreter.

Other programs, such as WordPerfect, combine the preamble with every script. So the Postscript program to produce one line of text with WordPerfect could be 7000 bytes, while the same Word file is less than 200.

You don't need a preamble if you're not defining functions and procedures ahead of time. However, a preamble can serve as a useful set of basic procedures. For instance, you could include this definition for a one-inch square in your preamble:

```
/square
{0 72 rlineto
72 0 rlineto
0 -72 rlineto
closepath} def
```

Now whenever you want to print a square, you specify the position on the page and the fill pattern, such as:

```
newpath
270 360 moveto square
fill
showpage
```

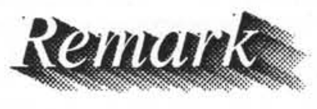


Figure 6

Using PostScript

There are three ways to use Postscript. You can use an application with a Postscript driver, write and download your own Postscript programs, or write the Postscript program interactively.

The easiest way to use a Postscript printer is with an application program that has a Postscript driver. You use the application's commands for selecting and sizing fonts, setting page formats, drawing lines, and merging graphics. The driver will then convert your document or drawing into the Postscript program and transfer it to the printer. A Postscript interpreter built into the printer then converts the commands into a graphic bit-map that's placed on the laser's drum and then on

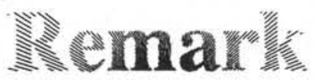
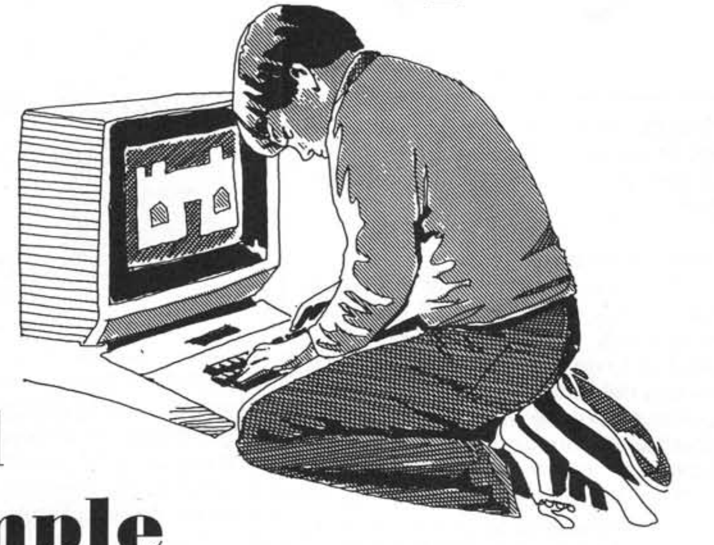


Figure 7

Continued on Page 36

Using Novell Netware on Zenith Data Systems Computers:



Harold C. Ogg
357 W. Diversey Avenue
Addison, IL 60101-3508

An Educational Example

Background

Zenith Data Systems' (ZDS) formal cultivation of the higher educational market makes the purchase of multiple computers practical and economically desirable. The company's technical support and price structuring (discount) incentives for post-secondary institutions render ZDS' equipment attractive where installations will be used for instructional purposes. It is, therefore, possible for an educational institution to benefit from the ruggedness and versatility of ZDS computers which the Military and commercial sectors have enjoyed for some time.

Recently, the Novell Corporation has made available its SFT Advanced NetWare as a grant to qualified educational institutions[1]. The company's intent, aside from obvious philanthropy and support of higher education, is to generate some advertising goodwill for its product and introduce students to the benefits of local area networks (LANs). In short, ZDS' pursuit of the educational market and Novell's active involvement in instructional settings provide an ideal marriage to proceed with installation of LANs on a demonstration level.

Choice of Software and Hardware

Novell's grant to our library was the champagne version of its software — version 2.15 SFT (System Fault Tolerant) NetWare. Our shop needed to connect a variety of Z-151s, Z-158s, Z-159s, and a couple of Z-248s. We also needed to connect an IBM-PC and a no-name PC clone. Although the NetWare required a minimally configured 80286-based machine for its file server, we decided that the computing power of an 80386 wasn't needed at this time. In fact, we determined that the 8 MHz Z-248 would do nicely for a file server since a great deal of traffic over the network was not anticipated.

The use of such high-powered software on (relatively) slower machines seems, at first examination, to compare with sending a cannon to do a popgun's job. Novell does vend abridged versions of its NetWare (see Alternatives section below), but the SFT version 2.15 offered to us actually proved ideal for our purposes. We needed the flexibility of being able to continue the use of a variety of third-party peripheral devices (such as CD-ROM drives) that have proven reliable with our Zenith Data Systems computers. And, we needed the capability of mix-and-match provisions to be able to use a wide variety of machines.

We also wanted ultimately to be able to interface with CD Net[2] and use the Z-159s as on-line work stations for data base (CD-ROM) searching. In the long run, we wanted to be able to bridge to our University's Academic Computing Department for resource sharing, a time which would involve interaction with that department's laboratory of Z-148s for student use.

Planning Out Particular Installation

The first consideration before committing to any new software configuration, operating system, or hardware upgrade should be "what is it going to be used for?" We knew that the primary functions of our LAN would be resource sharing (i.e., of high speed and laser printers), file transfer (word processing and desktop publishing), and CD-ROM searching. None of these tasks were time critical, and the word processing (document creation) events would be sporadic. We extrapolated the following characteristics for the initial phase of our LAN:

1. Since deadlines would not be relevant to document delivery, data throughput and speed of printer queuing were not major factors in our LAN environment.
2. Occasional multi-paged documents (as for desktop publishing) would not unduly tie up the network, since this activity was not pursued at the majority of work stations.
3. Data base searches would, for some time, be performed on local CD-ROM drives. The results of the searches would be printed over the network, usually in the evenings, and spooling (queuing) would not be a factor in tying up the server's resources.

Our local ZDS distributor recommended that we use Standard Microsystems Corporation's (SMC) ARCNET network cards. We needed only three varieties, the PC550FS for the Z-248 file server, a PC500WS 16-bit card for the one Z-248 12 MHz work station, and the PC260 for the Z-151s, Z-158s, and Z-159s. SMC's ARCNET cards can be purchased either in coaxial or two-wire twisted pair models, and we decided to go with the latter.

The twisted pair configuration allows a throughput of two megabits per second, which we felt was an adequate speed for our system. Twisted pair wires are easily run, and we felt that this medium would lend itself better to frequent reconfigurations and/or rewirings. Since our LAN would be used somewhat for instructional purposes, mobility was a concern where cables would have to be portable. Cables are easily built with twisted pair wires, and we required only an $8.95 wire crimper from Radio Shack to build our own RJ-11 terminated lines.

We also studied our floor layout diagram (the creation of which is essential to any successful installation) and determined that we would have no work station more than 100 feet apart from another. ARCNET employs a daisy chain/star topology, requiring that no node (chain of computer work stations) be more than 400 feet in length for twisted pair wires. It was necessary to plan a few years into the future, and we decided that we could easily radiate from the Z-248 within the maximum range for the foreseeable life of the LAN.

The Installation Process: Hardware Problems and Solutions

Reality begins to set in at the time all the paraphernalia are collected and you face the one and one-half foot shelf of NetWare manuals all together. Since the Novell grants are made directly from the vendor, you're not going through a reseller or dealer who may have performed some initial preparations for you. In short, you're the hardware and software installer, supervisor, maintenance technician, work station organizer, security liaison, and chief cook and bottle washer.

It is a time such as this that I face Benton Harbor and praise the salesperson who, many years ago, persuaded me to plunge into the realm of the H-8 and its multitude of assembly manuals. Although not cryptic, the Novell setup process requires a certain amount of intuition and a bit of experience for which a hobbyist or kitbasher has a decided edge. Novell's manuals are a bit "wordy," a necessity for walking a neophyte through the various processes. There is some excellent supplemental literature on NetWare installation and use, and appropriate titles will be given at the end of this article.

A parenthetical note: Installers relatively new to computer architecture, and persons who, when facing Novell's intricacies of installation for the first time, feel a need to bridge a gap in their computer educations, will find ZDS' Data Communications and Networks[3] individual learning program invaluable. A considerable amount of the material therein deals directly with PC networks and their related terminologies. ZDS recommends that you preface the course with a regimen of digital techniques (also offered in tutorial form by ZDS) and that you employ the accompanying trainer and an oscilloscope. Necessary parts for the twelve experiments are included with the course. I have taken several of Zenith Data Systems' individual learning courses, and I have found that use of the relevant hardware with your study makes the learning process considerably more meaningful. However, for a familiarization of the basic data communications vocabulary, of which much reiterates in the Novell manuals, a reading study of the course will suffice.

The first thing you should do, unless your ZDS 80286 machine is relatively new, is replace the system ROMs. As of this writing, these are part numbers 444-423-10 and 444-424-10 and sell for $13.05 apiece. Check with Zenith Data Systems' Parts Department before you order — there may be a price change and a newer release of the firmware.

The other preliminary preparation is to install a new hard disk controller board. Our ZDS/Novell distributor advised that he had "seen some problems" with the cards in the older Z-248s and recommended a DTC 5250 to reduce the possibility of software conflicts. This recommendation is confirmed by Novell as well, for at one of the first troubleshooting junctures in the installation manual is the caveat "check compatibility of your hard disk controller" in the event of an abend when the system software examines the computer's internal configuration. I took the dealer's and Novell's advice on faith, and the new controller board worked flawlessly.

Some choices which are irrevocable must be made from the beginning. Although Novell's other flagship product, the Advanced NetWare, allows the server to perform double duty as a standalone DOS work station, the more sophisticated SFT NetWare does not. If you ever want to use the machine as a work station, you must tell the setup program that you want a DOS partition on the hard drive. And you cannot boot the SFT NetWare from DOS; to access the DOS partition would require booting from a floppy disk and switching to the hard drive. Since the inclusion of a DOS partition obviously reduces the storage space for the network, and setup of a DOS partition would require some running of the customization module of the NetWare, we opted (as stated above) to let the Z-248 run as a fully dedicated server.

With the SFT NetWare, you must use at least an 80286-based machine for the server. Full-fledged NetWare is truly not a program to attempt to run using a PC or XT as a pivot. Further, you need at least one megabyte of memory above the 1 MB address boundary. Novell, in fact, recommends a minimum of two megabytes of such memory if a hard drive(s) of more than seventy megabytes capacity is installed. And the configuration must be extended (linear) and not expanded RAM, since SFT NetWare accesses higher memory addresses directly via the 80286's selector:offset mechanisms in much the same manner as OS/2. Two examples of extended RAM boards are the AST RamVantage and the Everex 3000. Each allows 2.5 megabytes of additional RAM and also provides a full 640K roundout to the older 512K machines.

Since we used ARCNET boards, only one PC550FS file server board was necessary. You can have multiple file servers if desired, but software bridges (from another LAN) require the purchase of an additional set of NetWare. ARCNET cards are relatively inexpensive, and they work with Zenith Data Systems computers right out of the box, with the default settings. You need only verify that the interrupt setting is IRQ2, and that the I/O address is 2E0h (Novell's standard). The only setting for which you must make a choice is the numeric identifier for each work station. This was a simple matter of establishing our file server as station #1, the first connected Z-159 as station #2, etc. The only caution you should take is to double-check the dipswitch settings: ARCNET designates "on" as 0 and "off" as 1. But the documentation with each card gives switch diagrams for any possible combination of settings, and the documentation is clear and to the point. In fact, ARCNET includes the default settings for Novell NetWare in its illustrations.

We decided to substitute a Seagate 4096 80 MB hard disk drive for the 40 MB model already in place. The 4096 is a full-height drive which has been available for some time, and has proven rugged and reliable. The Z-248 will accept two full-height drives without any sacrifice of space for other purposes. For a small LAN, Novell recommends at least a 40 MB drive. However, a prereading of the literature told us that the hard drive was used as a storage area for printer queues, suggesting that some free space was needed for scratchpad work. We also wanted to allow some room on the server for special user accounts. Also, we substituted a 360K floppy disk drive (reasons below) for the 1.2 MB unit already in place. The drives installed without problems, and it was necessary only to note the changes in the Z-248's setup ROM. The Seagate 4096 is type 35 in the ROM's list of drives, so no special considerations were needed for the hard drive's configuration. No other changes or modifications were necessary to the Z-248.

We installed two printers on the LAN — a high speed Epson FX-850 for drafts and literature search printouts, and a

Hewlett-Packard LaserJet II for desktop publishing documents. The latter was connected to the Z-248's COM1: port to save a slot from an additional I/O card. From a work station, NetWare allows you to redirect a local LPT: to a server's COM: port transparently. Be careful that you connect a serial printer to the server with a null modem cable (pins 2 & 3, 4/5 & 8, and 6 & 20 switched); otherwise, the COM: port cannot communicate with the printer.

It was necessary at this point only to run the cabling. We used shielded twisted pair wire to prevent interference from fluorescent lights in the ceilings. NetWare provides a COMCHECK utility to check cabling and hookups. Essentially, the program allows each connected PC to "see" all others; any station not reporting can be assumed incorrectly wired or bearing a faulty network card. This utility helped us to identify an overly long cable (ARCNET provides a list of maximum runs in a specifications guide), into which we had to splice a signal repeater box. Aside from this one glitch, the system tested perfectly the first time it was powered on.

The Installation Process: Software Problems and Solutions

Interestingly, Novell furnishes the 5-1/4" disk version of the NetWare on 360K floppies, and not the 1.2 MB versions found in most AT-class machines. There have been some reports of difficulties writing to 360K disks in a 1.2 MB drive, and, although the Z-248 never has caused me any problems in this regard, I decided to take out a bit of insurance by substituting the 360K drive. The installer is urged to work from backup diskettes, for the obvious reasons, and also because some writebacks are made to the installation disks. In fact, two of the master diskettes are blank, placed in the set as placeholders to ensure that the resultant installation disks have the same number of floppies as the shipped masters.

It is advisable to purchase high quality diskettes for the backup copies. NetWare requires that you use the DOS DISKCOPY command to make the backup disks, since DISKCOPY formats, copies subdirectories, and copies volume labels (NetWare uses volume labels to check appropriate disk insertion during the installation process). The DOS XCOPY program would solve the problem of bad sectors on disks, since that utility has the ability to lock out disk imperfections. However, on some of the NetWare disks there is only one or two kilobytes of space left — not enough to prevent an "insufficient disk space" error if a sector is locked out. None of the disks are copy protected (Novell uses a serial number encryption to prevent unauthorized duplicate usage), so the copy process is straightforward.

To install, you must use DOS version 3.1 or greater; The ZDS supplied version works fine. Once installed, the individual work stations can run using DOS versions as old as 2.1. However, Novell recommends that all stations use the same, newer versions for uniformity. We installed and configured work station shells from the server, which already contained the 3.3 version of DOS. What the installation manual doesn't tell you is that you should initially boot the server from a floppy disk. Seagate (and many other) hard disk drives use a device driver (such as Ontrack's DMDRVR.BIN) to access the hard drive's partitions. The aforementioned conflicts with NetWare's attempt to install its own disk controller driver, and the installation abends in DMDRVR.BIN's presence. The installation should be run with no device drivers or memory resident programs present, just to be on the safe side.

There are fifty 360K floppy disks in the shipped version of SFT NetWare, but you won't be using them all. Some are for special interfaces, such as to an IBM 3270 mainframe. We used the following procedure for installation:

1. Generate the work station shells. The first activity is the creation of installation disks for the various PCs to be attached to the LAN. All but one of our PCs were Z-151/Z-158/Z-159s with ARCNET PC260 cards installed, running version 3.3 MS-DOS. The resultant main files were the configured versions of IPX.COM (the internetwork packet exchange) and NET3. COM (the DOS version 3.x communications program) and could be copied repeatedly from the same working disk. NET2.COM would be created for DOS version 2.x.
2. Generate the work station software. Standard Microsystems Corporation supplies its own device driver for the 500-series ARCNET boards (server and 16-bit work station). Because of some considerations for multiple servers, the driver diskette contains subdirectories that the NetWare installation module cannot access. Some experimenting proved that the directories could be eliminated, with all driver .OBJ files placed in the root. Thus, when the NetWare prompts "Do you have other drivers to install?," the LAN_DRV_ 500 volume will be fully visible to the NetWare program. Other than this special preparation, the NetWare allows you to accept the default system parameters very easily, since the installation program examines your hardware to determine what is installed. Of course, if you have some special considerations (such as a disk co-processor), the custom configuration module is menu driven and not difficult to run in lieu of the defaults.
3. Prep(are) the hard disk drive. At this point, you must finish the installation from the floppy disk drive. Your preparations of the server programs will have resulted in eight to ten altered floppy disks (from the backup copies), and you'll need to use a few more of the backups which have not been altered. NetWare employs its own hard disk drive verifier (COMPSURF) to check for bad sectors, prep and partition the drive. Because of the thoroughness of the program, it is best to let it run overnight if you don't want to "babysit" the computer. Remember, the server software runs an operating system which is not DOS (although it will accept many DOS commands). If you haven't set the hard drive type from the ROM setup routine, the NETGEN routine will do it for you.

There is a paradox in the hardware/software installation processes. Once the installation sequences are complete, all you have to do to add more work stations to the original setup is run the appropriate cable, install the LAN boards with unique station numbers, and copy the work station shell to the local drive. Unfortunately, there is usually no way to alter the server configuration (i.e., to reconfigure the server by adding another hard drive or an additional server board) without repeating the entire installation process (with the exception of the hard disk prep utility). This is why it is so very important to plan ahead and try to anticipate as many needs as possible.

The Application and the Benefits

As an educational agency, and more specifically, a university library, ours is not the typical office environment. A library is interested more in information retrieval than real-time file sharing, so our LAN traffic usually stems from multi-location accesses performed one at a time. Our enjoyment of the Zenith/Novell based LAN is as follows:

1. Laser printer sharing. The LAN allows more of our faculty members to study and experiment with applications for desktop publishing, an activity which has brought some measure of recognition to our department.
2. CD-ROM searches. This provides a twofold benefit: Our library has some individual CD-ROM work stations for data bases such as ERIC, Compact Disclosure, and MedLine. The LAN made it possible to perform multiple data base searches, spooled to a single high-speed printer away from the study (quiet) areas. It also reduced the turnaround time for searches from two days to two hours.

An aside: The Zenith Data Systems computers interface quite well with a variety of CD-ROM drives. However, we have experienced some problems interfacing with Hitachi units. Specific reasons for the problem have not been determined.)

3. Training. NetWare's own built-in tutorial package, along with the portability afforded using twisted wire cabling, allows us to bring PC and network demonstrations into classrooms and conference areas. The presenter loads relevant files to the public area of the server and subsequently uses those files from the portable work station at the time of demonstration. And the ZCM 1490 flat screen monitor proves especially readable at conference tables.
4. Flexibility in future expansion. One goal was to use a CD-Net setup for departmental access of several bibliographic data bases. All of the networkable CD-ROM data base systems we have examined list Novell NetWare as their primary reference software for compatibility. And, the NetWare now shipped includes a software interface to Macintosh boxes.
5. Adaptability to currently used software programs. Novell's software supports network versions of programs we use frequently, such as Ventura Publisher, WordStar and Word Perfect. In addition, we were able to effect remote control of the ZDS computers through the Carbon Copy Plus[4] telecommunications program. I was able to use my own Z-248 to control the network over telephone lines, and could perform some maintenance work after hours from my own home study.

Advice and Suggestions

Since performing our own installation and maintenance made us de facto experts on NetWare, some thoughts for the advisement of future aspiring networkers came to mind:

1. Cable the entire system first, before proceeding with software installation. It is much easier to spot and correct hookup errors before attempting to run network programs. If problems persist once the hookups have been verified, you at least have the secure knowledge that a program dysfunction can be successfully traced to the software.
2. Use the 720K 3-1/2" disks, if possible, for installation. Simple arithmetic tells you that you'll be doing half as much disk swapping as with 5-1/4" diskettes, if you work all the way with a floppies-only installation.
3. Install a tape streamer as part of the initial hardware configuration. Hard drive backups are relatively easy using a floppy disk drive with programs such as PC-FullBak+[5], but only with drives of about twenty megabytes capacity or less. The necessary disk swapping during the copy process entails too much manual labor otherwise. Get the tape streamer and let the machine do all the backup work.
4. Buy an uninterruptable power supply (UPS). Again, this is cheap insurance. All you lose is one open slot and a few minutes' installation time.
5. If printer throughput is not time critical, use one of the available COM: ports for output. This was mentioned above, and you can save an open slot if you determine that another LPT: isn't necessary to alleviate long print queues.
6. You can save a few dollars using a monochrome monitor on the server. NetWare will allow you to set screen colors on the server's console, but unless there will be a considerable amount of network maintenance, save the color monitors for the work stations.

Alternatives and Sources

There are some inexpensive devices on the market that work well if all you want to do is share printers. These devices are sometimes called "smartboxes" and take care of switching and queuing between several sets of PCs and connected printers.

One often overlooked treasure is the ZCOM utility which ZDS has been providing with recently shipped versions of MS-DOS. It allows two computers to connect through their respective COM: ports and transfer files back and forth. One computer acts as the server and the other as a command console (you can interchange which is which), and file copying can take place at speeds of up to 57,600 baud. I have seen similar programs at computer shows selling for as much as $100. ZCOM is a bonus if you purchased MS-DOS from Zenith Data Systems.

For those persons who can't afford the price of the full-fledged NetWare, two abbreviated versions are available. ELS I NetWare 286 is a more economical alternative, and is restricted to a maximum of four connected work stations at any given time. It also supports fewer types of network cards, and offers less sophisticated printer spooling. ELS II NetWare 286 supports up to eight work stations and, if the server is set up in a non-dedicated mode, it can be run from an XT-type file server.

Novell recently announced a release of NetWare which is written for 80386-based machines[6]. The company has designated NetWare 386 version 3.0, stating that there are no current plans for a 3.0 level release of NetWare 286. At this writing, a release date and version for any subsequent release of the 80286 software have not been announced.

Novell acknowledges the use of value added packages (VAPs), and there are some that extend the usefulness of NetWare for particular applications. *Building Local Area Networks with Novell's NetWare*[7], which also offers a supplemental disk ($20) of shareware network tools, contains a chapter of annotations of third-party utilities, with addresses for further information on each. One such entry is PS-Print[8], which allows reverse spooling, or output to a local printer from another work station on the network. There are also several electronic mail (e-mail) packages listed. Public Brand Software[9] in its catalog lists an e-mail utility and a couple of support programs for networks.

Howard Sams publishers has two recent books designed for quick assimilation of the basics of networking, in general, and Novell NetWare, in particular.[10,11] Sybex has published a beginning level text on NetWare[12], and *Building Local Area Networks* (mentioned above) nicely consolidates and re-explains much of the material in the Novell User's Guide. Compared to other software products, there is not a wide variety of titles in the bookstores on NetWare. However, what I have seen thus far is authoritative and well written.

Bibliography and Software Sources

[1] Persons interested in Novell's grant support for higher education should write to: Novell Corporate Sales — Education, Education Grants, 122 East 1700 South, Provo, UT 84606.

[2] CD Net. Meridian Data, Inc., 4450 Capitola Road, Suite 101, Capitola, CA 95010. Vended as a package with hardware systems; write for prices and specifications of configurations.

[3] Stanley, Richard A. Data Communications and Networks. Benton Harbor, MI. Heath Company, 1986. Catalog #EE-8090, $99.95.

[4] Carbon Copy Plus. Microcom Software Division, 500 River Edge Drive, Norwood, MA 02062. $195.00. Note that two copies of the program are required for an initial setup, and that additional copies must be purchased for each remote station.

[5] PC-FullBak+. Westlake Data, P.O. Box 1711, Austin, TX 78767. $99.00.

[6] Derfler, Frank J., Jr., and Thompson, M. Keith. "Building Workgroup Solutions: Novell's Netware 386." PC Magazine, December 12, 1989, pp. 205-221.

[7] Corrigan, Patrick H. and Guy, Aisling. Building Local Area Networks with Novell's Netware. Redwood City, CA: M&T Books, 1989. $24.95.

[8] PS-Print. Brightwork Development, P.O. Box 8728, Red Bank, NJ 07701. No price given.

[9] LAN Support. Public Brand Software, P.O. Box 51315, Indianapolis, IN 46251. Disk #CO12.0. $5.00 plus $5.00 postage and handling.

[10] Schatt, Stan. Understanding Local Area Networks. Indianapolis, IN. Howard Sams & Company, 1987. $16.95.

[11] Hader, M. Mastering NetWare. Indianapolis, IN: Howard Sams & Company, 1989. $24.95.

[12] Woodward, Jeff. The ABC's of Novell NetWare. Berkeley, CA: Sybex, 1989. $22.95.

FREE MICROSOFT WINDOWS * DETAILS BELOW

Seagate HARD DRIVES

MODEL	CAPACITY/FORMAT/SPEED/SIZE	DRIVE ONLY	XT KIT
* ST-125	21 MEG / MFM / 40 MS / 3.5"	$229.00	$279.00
* ST-138	32 MEG / MFM / 40 MS / 3.5"	$277.00	$327.00
* ST-138-1	32 MEG / MFM / 28 MS / 3.5"	$307.00	$357.00
* ST-151	42 MEG / MFM / 24 MS / 3.5"	$353.00	$403.00
* ST-138R	32 MEG / RLL / 40 MS / 3.5"	$258.00	$313.00
* ST-157R	49 MEG / RLL / 40 MS / 3.5"	$286.00	$341.00
* ST-225	21 MEG / MFM / 65 MS / 5.25"	$199.00	$254.00
* ST-251-1	42 MEG / MFM / 28 MS / 5.25"	$312.00	$362.00
* ST-4096	80 MEG / MFM / 28 MS / 5.25" FH	$582.00	$633.00
* ST-238R	32 MEG / RLL / 65 MS / 5.25"	$218.00	$273.00
* ST-277R-1	65 MEG / RLL / 28 MS / 5.25"	$348.00	$403.00
* ST4144R	122 MEG / RLL / 28 MS / 5.25" FH	$623.00	$678.00

* IDE, SCSI, ESDI AND OTHER SEAGATE MODELS AVAILABLE. PLEASE CALL.

*** ZENITH PC COMPUTER UPGRADES ***

SmartWatch from FBE RESEARCH

⇨Installs in ROM Socket on the CPU Board in Zenith computer series Z-100/ 138/148/150/160 and most all other XT computers. This clock/calendar contains a ten year battery and keeps your computer informed of both date and time at each boot-up. Instructions and software included. $35.00

Z-150/160 MEMORY UPGRADE

⇨This kit includes a replacement memory decoder PAL chip for the standard Z-150/160 memory card (not for the Z-157/58). With this PAL and the 18 pieces of 256K RAM chips (included), you will expand the memory on the card to 640K or 704K. ZP640+/18 Kit.....$59.00. PAL chip only ZP640+....$18.00

Z-150 SERIES HARD DISK DRIVE KIT

⇨These kits include high speed Seagate drives with autopark heads. Each kit includes all cables, hardware and instructions to mount the hard drive under your two floppy drives in your Z-150 series computer.

* ST-125/Z150 Kit	21 Meg, 40 MS,	$283.00
* ST-138/Z150 Kit	32 Meg, 40 MS,	$331.00
* ST-151/Z150 Kit	42 Meg, 24 MS,	$409.00

Z-148 HARD DISK DRIVE KIT

⇨Includes the hard disk drive and a Z-148 compatable controller together with the Z-148 Expansion Card described below. All required cables, hardware and instructions are included for you to replace one floppy with a Seagate Hard Drive in your Z-148. Add only $30.00 the the following price if your would like us to include a SmartWatch.

* ST-125/Z148 Kit	21 Meg, 40 MS,	$354.00
* ST-138/Z148 Kit	32 Meg, 40 MS,	$402.00
* ST-151/Z148 Kit	42 Meg, 24 MS,	$478.00

Z-148 EXPANSION CARD

⇨Adds one full length and one half length IBM expansion slot to your Z-148 for hard drive controller, video card, modem, etc. ZEX-148......$79.00

Z-150 VIDEO ELIMINATOR

⇨For the Z-150 or Z-160 only. Not required for the Z-157/158/159 computers. A small piggyback board which replaces the scratch pad memory on your current video card. This allows the removal of the original Zenith video card and replacment with an EGA, VGA or any other 8 bit video card. Order VCE-150 .... $54.00

2400 BAUD MODEMS

⇨Fully Hayes compatible 2400/1200/300 Baud with Software. Internal $99.00 External Model $128.00 Cable for External Modem $8.50

VISA DISCOVER MasterCard

PAYLOAD

15718 SYLVAN LAKE * HOUSTON TX 77062

FLOPPY DISK DRIVES

⇨MITSUBISHI MF501	5.25" 48 TPI DS/DD 320K/360K	$ 68.00
⇨MIITSUBISHI MF504	5.25" High Density 360K/1.2 MEG	$ 81.00
⇨ MITSUBISHI M-353	3.5" in 5.25" frame 720K	$ 84.00
⇨MITSUBISHI M-355	3.5" in 5.25" frame 1.44 MEG	$ 94.00
⇨TOSHIBA ND352	3.5" with 5.25" frame 720K	$ 74.00
⇨TOSHIBA ND356	3.5" with 5.25" frame 1.44 MEG	$ 79.00

⇨M-355 and ND356 run on AT compatible or special controller only.

PAYLOAD CUSTOM ASSEMBLED COMPUTERS

⇨We assemble 8088 XT, 80286 AT, 80386SX or 80386 IBM compatible computers to your specifications. Please write or call for a work-up sheet showing items available and prices.

VIDEO MONITORS

⇨ZCM-1492	ZENITH Color Flat Screen VGA	$ 679.00
⇨MA2565	SAMSUNG Amber TTL 720x350	$ 89.00
⇨CW4656	SAMSUNG Color RGB 640x200	$ 223.00
⇨CM4592	SAMSUNG Color EGA 640x350	$ 339.00
⇨CJ4681	SAMSUNG VGA 720x400	$ 360.00
⇨CVB4581	SAMSUNG Multi-sync VGA 1024x768	$ 469.00
⇨CM1440	SEIKO VGA 1024x768 .25 dot	$ 609.00

VIDEO CARDS

⇨EGA480	PARADISE AUTOSWITCH 640x480	$ 99.00
⇨VGAPLUS	PARADISE AUTOSWITCH 800x600	$ 168.00
⇨VGAPLUS16	PARADISE AUTO 16 BIT 800x600	$ 207.00
⇨VGA 1024-256k	PARADISE AUTO 1024x768	$ 249.00
⇨VGA 1024-512k	PARADISE AUTO 1024x768	$ 309.00

MEMORY CHIPS, ETC.

⇨Memory chips are once again at reasonable prices. The market prices have been changing daily, therefore we are only able to list estimated prices. Please call for the current price before placing your order. We buy in large quantities and work on the smallest of margins in order to bring you great values.

⇨41256 256x1 80 ns.....$2.95	SIP 1Mx9 80 ns $80.00
⇨41256 256x1 100 ns $2.65	SIM 1Mx9 80 ns $79.00
⇨41256 256x1 120 ns $2.30	SIM 256x9 80 ns $24.00

Z-100 SERIES COMPUTER UPGRADES

High Density 1.2 Meg Drives

⇨External floppy drive set-up complete with drive, power supply, case and cables. Ready to connect to your 8" floppy controller. Single Drive Unit $217.00

⇨Dual Drive Unit $309.00 Bare drive and cable for internal mount $127.00

ZMF100A by FBE Research

⇨A modification package which allows 256K chips to be used on the old-style motherboard (part number85-2653-1) to reach 768K. Simple assembly with no soldering or trace cutting. Compatible with Easy PC and Gemini Emulator. Order 27 256x1 RAM chips to complete this kit. ZMF100A$60.00

UCI Memory Upgrade Pal Chip Set

⇨For the Z-100's with the newer motherboard part number 181-4918 or greater. Allows the installation of 256K RAM chips on the motherboard. With the addition of 27 256K 150 ns RAM chips (sold separately) a total memory of 768K is obtained. Chip Set....... $64.00

UCI EasyWin HARD DRIVE SYSTEM

⇨Complete Hard Disk System for mounting inside your Z-100. Includes S-100 bus board, matched XT hard disk controller, EasyWin software, manual and Misc installation hardware. Order a hard disk (ST-125 or ST-138 recommended) under the SEAGATE HARD DISK DRIVE ONLY listing to complete the kit. $288.00

Z-100 SERIES SOFTWARE

⇨PART NUMBER /DESCRIPTION	LIST PRICE	SALE PRICE
⇨CD-463-2 Condor File Manager	$175.00	$10.00
⇨MS-253-1Basic-80 (8-bit)	$175.00	$10.00

MICROSOFT WINDOWS version 1.04 for PC (not Z-100)

⇨FREE with any order from this ad over $30.00. Just ask for it and add $5.00 for
⇨shipping and handling. Includes 5 disks and 300+ page manual. Offer good until
⇨existing inventory given away (about 300 packages).

⇨Please Mail, Phone or FAX your order. All hardware carries the manufacturers warranty plus the PAYLOAD 90 day guarantee. No surcharge on credit card orders. COD shipments on request. Add $5.00 to all prepaid orders for handling and shipping in the Continental USA, we pay the balance. Actual shipping costs for foreign, overseas and net billing orders. We accept purchase orders from schools, government and approved accounts. Mail or phone your order for prompt friendly service. Texas residents please add 8.0% state sales tax.

PHONES: ** ORDERS AND INFO **(713) 486-0687** ** FAX: **(713) 486-8994** **

PAYLOAD * PAYLOAD * PAYLOAD * PAYLOAD

Reader Service #117

UNIX on an 8088

Jack W. Bazhaw
900 - 13 Street
Bellingham, WA 98225

(Well, Almost!)

In the July 1988 issue of REMark, we took a "Peek at UNIX" and how it was an operating system that could make better use of the 80386 chip than DOS could. Of course, if your budget keeps you with your H/Z-150 or other 8088 machine, there is no reason to think about UNIX™. Right? Wrong!

If you need to process large files, do programming or use UNIX elsewhere, then you may want to consider UNIX on your H/Z-150. Not real UNIX, of course, yet something so close you would have to look twice to tell it from the real thing. You might even be interested in it as a low cost way of learning about UNIX.

POLYTRON Corporation in Beaverton, Oregon makes a product called PolyShell™. PolyShell is a UNIX compatible command interpreter that runs under DOS. It is not UNIX so there is no multiuser multitasking capability, nor will it run UNIX application programs, such as Informix™. But the flavor and a lot of the power of UNIX is contained in this program. It is as much as possible functionally compatible with the Bourne Shell, standard on most UNIX systems, plus it supports several extensions from the Berkeley "C-Shell" and the newer "Korn-Shell" from Bell Labs.

For those new to UNIX there is a 40 page tutorial included in the 300 page manual. It will not make you an expert, but it is a lot clearer than most UNIX material I've seen. It does assume you are familiar with DOS.

The commands seem to support all of the normal UNIX features; some have features available only with PolyShell. Some of the nonsupported features are obvious, like the use of the "&" to have a command perform as a background task. Sorry, no multitasking supported, remember.

POLYTRON markets it as a DOS extender and command interpreter for programmers. Even if you don't program it can be very useful.

For example, a project I'm working on starts out with a large file (almost 500 KB and still growing) which I have been massaging with Microsoft Word. Recently, I decided to do the final printing using a proportional spaced typeface. Big problems. The file is mainly tabular material and the program that creates the file uses spaces to line everything up, expecting printing to be done with a monospaced typeface. When I changed to a proportional spaced typeface nothing lined up.

To print properly with a proportional typeface, it was obvious tabs were needed in place of spaces. That meant I either had to change the program that generated the file or change the file. Even if the source code for the program were available to me, it would be doubtful I could make the changes, and I could imagine the reaction of the program's author for a "one-of-a-kind" modification.

So I began to write some editing macros for Word. Now how do you write a macro to change a variable number of spaces at the start of a line into one tab? And another one to remove a variable number of spaces at the end of the line. While thinking about how to structure the macro, I felt the need for the construct that UNIX's streaming editor (sed) uses; I know how to use sed to replace spaces at the start of the line. In fact, the command is very brief. My experience with Word macros and large files led me to believe writing a solid macro would take quite a bit of time.

Having purchased PolyShell for another project, I decided to give it a try first. It seemed strange at first, sitting down at a DOS machine to run a UNIX command on it. My first crack at putting in the starting tab worked like a champ; removing the ending spaces was also straightforward. I'm not sure Word could have handled this large a file with a search and replace operation, since it is such a memory intensive function. Sed never complained or required an intermediate save. I added six more edits to the operation since I was on such a roll. Figure 1 is the final script; not elegant, but very useable.

```
sed 's/^ */             /
s/ *$//
s/:   */:                     /
s/: /:                  /
s/    */ /
s/in:                /in: /
s/Ref:                 //
/Other:/d' 28sep89.txt > 28sep89.doc
```

Figure 1

The parts that I foresaw trouble writing the macro in Word is handled by the first two lines. Everything between the two single quotes (') is a command for sed to perform. All but the last one is a substitution command. "s/^ */   /" means to substitute for zero, one or more spaces (the unseen space and the *) at the beginning of a line (indicated by the ^) the (unseen) tab between the last two slashes. "s/ *$//" means to substitute for zero, one or more spaces (again, the unseen space and *) at the end of a line (indicated by the $), which means there is nothing between the last two slashes.

Word has no ability to look for a variable number of only spaces. The white space search pattern (^w) will also find all sorts of other characters, like tabs and new lines. Trying to put a tab at the start of the line with column select produced a lot of disk activity, the message "scratch file full" but no changes to the file. In fact, it took almost as long for Word to produce no change as it took the sed script to do everything.

Sed made a nice compliment to Word by handling the brute force replacements with no hassle. Without PolyShell I would have had to bring the file to a UNIX machine for processing and then return it for final Word processing. Not near as convenient. Sed is certainly not a replacement for Word, yet for some applications it is better.

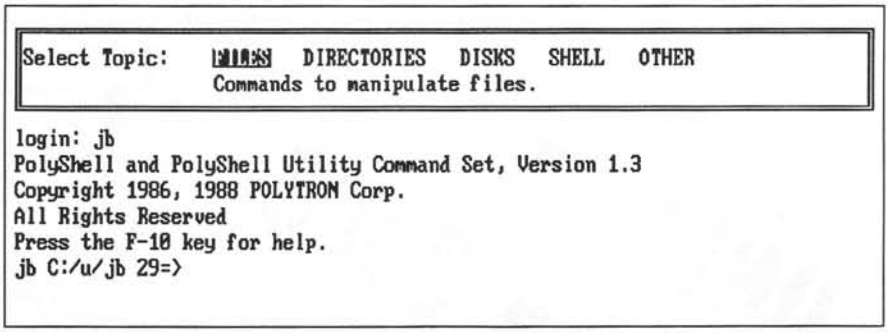
PolyShell was designed to provide programmers using DOS the power and flexibility of UNIX. It would also be a good way for someone to learn a lot about UNIX without making the larger software/hardware investment needed for "the real thing." Even straight DOS users can find applications for it. It will run on most H/Z-150 compatible machines and some generic MS-DOS machines. A hard disk would be preferred, but is not required. Everything, including the help files, will fit on a 1.2 MB floppy with 400 KB left over for your use. Only 20 to 30 KB of RAM memory is used when the shell is not actually running a command, the exact amount depending upon what aliases, history, etc. you set up.

DOS commands are run, even the internal ones like dir and type, through the use of aliases. Most of the well-used DOS commands are already set up with an alias in the alias.sys file.

UNIX has a reputation of being difficult to install. PolyShell installation takes only a few minutes. In keeping with standard practice, two main directories are used, /etc and /usr. Executable files are put in /usr/bin. It also expects to add users in /usr, as is normal practice. I opted to keep the same structure as the UNIX machine I work on and put users in /u. Part of the simplicity is due to what is missing. No uucp files, no cron running, no termcaps to fiddle with.

A feature PolyShell has I wish my UNIX machine at work had is interactive help. Just type a command, like sed, press F10 and information about command options, etc. becomes available at the top of the screen. This sure beats grabbing a manual or running a "man" command (yawn) and then having to take notes off the screen. One minor drawback to the help function is it does not restore the screen, so it's best to make sure you do not need to keep what appears at the top.

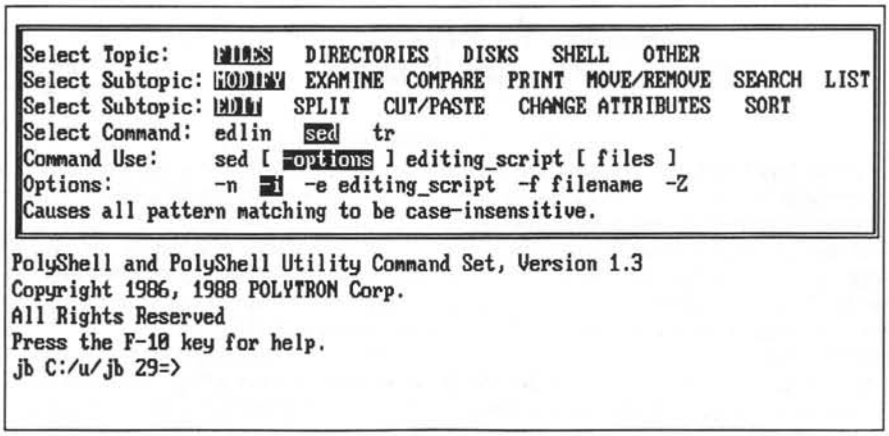
Pressing F10 with no command typed displays the general help screen show in Figure 2.



```
Select Topic:     FILES   DIRECTORIES   DISKS   SHELL   OTHER
                  Commands to manipulate files.

login: jb
PolyShell and PolyShell Utility Command Set, Version 1.3
Copyright 1986, 1988 POLYTRON Corp.
All Rights Reserved
Press the F-10 key for help.
jb C:/u/jb 29=>
```

Figure 2

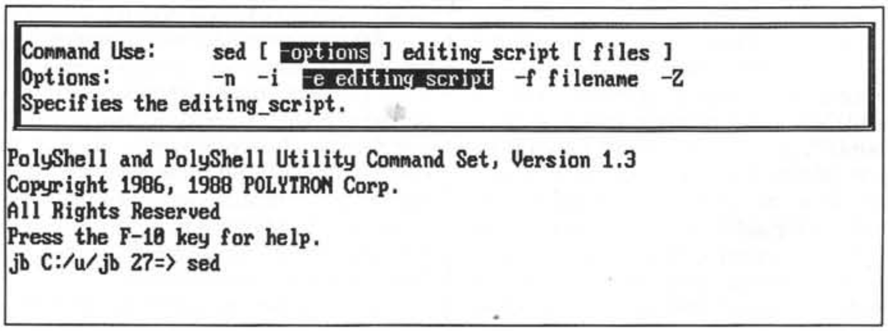
By pressing F10 again additional lines of help for the highlighted topic appear. Pressing F7 and F8 moves the highlight left and right. Figure 3 illustrates obtaining help about sed without having to know the command name.



```
Select Topic:     FILES   DIRECTORIES   DISKS   SHELL   OTHER
Select Subtopic:  MODIFY  EXAMINE  COMPARE  PRINT  MOVE/REMOVE  SEARCH  LIST
Select Subtopic:  EDIT   SPLIT   CUT/PASTE   CHANGE ATTRIBUTES   SORT
Select Command:   edlin   sed   tr
Command Use:      sed [ -options ] editing_script [ files ]
Options:          -n  -i  -e editing_script  -f filename  -Z
Causes all pattern matching to be case-insensitive.

PolyShell and PolyShell Utility Command Set, Version 1.3
Copyright 1986, 1988 POLYTRON Corp.
All Rights Reserved
Press the F-10 key for help.
jb C:/u/jb 29=>
```

Figure 3

Typing the name of a command and pressing F10 instead of Return will display help about that command as shown in Figure 4.



```
Command Use:     sed [ -options ] editing_script [ files ]
Options:         -n  -i  -e editing_script  -f filename  -Z
Specifies the editing_script.

PolyShell and PolyShell Utility Command Set, Version 1.3
Copyright 1986, 1988 POLYTRON Corp.
All Rights Reserved
Press the F-10 key for help.
jb C:/u/jb 27=> sed
```

Figure 4

The prompt you receive after login is initially a percent sign (%) and the command number. Like UNIX (and DOS), the prompt can be customized and the tutorial contains some suggestions. To make life easy, I've made my prompt mimic the one I use at work with the exception of adding the logged drive. UNIX doesn't care which drive is involved, but DOS is an old maid about such things. Figure 5 is profile.sh and shows the prompt set to display the user, drive, working directory and history number. The escape sequences are standard ansi.sys commands to set the display colors (escape is represented by "^"). The amount of history retained is set at 300.

Something like 100 commands are available. Almost all can be run either from PolyShell or directly from the DOS command line. Like DOS, there are a few "internal commands" that must be executed by Polyshell.

Here is a sample of some of the available commands:

- *basename* — strip pathname prefix and suffix
- *cut* — cut columns from a file

- *diff* — compare files, print lines that differ
- *egrep* — search files for an extended pattern
- *head* — print first lines of a file
- *tr* — translate characters
- *uniq* — print unique lines from a file
- *wc* — count words, lines, characters or pages in a file

```
echo Press the F-10 key for help.
PS1='^[1;34;40m$USER^[m ^[33;40m$dd:$pwd $his=>^[0;32;40m '
HISTSIZE=300
```

Figure 5

One of the nice features to have available is command history which lets you recall, edit and execute previous commands by name, number or scroll using arrow keys. Command history is saved to a disk file which keeps you from losing track of what you did after logging off. The "Korn-Shell" command "fc" is available to edit commands. Setting the variable FCEDIT will call your favorite editor or you can specify it on the command line; otherwise, the default edlin is called.

If you have written many batch files under DOS, you know how limiting the command structure can be. Shell scripts written for PolyShell may include the "if", "for", "while", "until" and "case" control statements.

Several example scripts are provided. Figure 6 is mkuser.sh which creates needed files and directories for new users. It is quite long and illustrates many of the features of UNIX scripts.

An alias is one or more commands which have been given a shorthand notation or macro name. Several standard aliases are included in alias.sys. Figure 7 is my alias.sys file, which contains the standard ones plus several I have added. My additions are shown with the macro name in caps, however, PolyShell is case-insensitive, like DOS. Real UNIX is case sensitive; I retained the all caps from the UNIX machine. The first word on each line in alias.sys is the alias to type. The remainder of each line is the commands that will be executed.

More than one command can be entered on a line. Just separate each command with a semicolon. This feature, combined with history substitution can let you create an alias "on-the-fly". For example, suppose you want to edit a batch file, then execute it. To check its operation, you will look at the start of another file. The first time through it fails and you realize you also need to check for the presence of another file. You might wind up doing this sequence of events:

```
%1 edlin test.bat; test; head -20 form.doc
%2 !1; ls -l /tmp/check.sum
```

Command line one calls edlin to edit the file test.bat, then executes it and then prints to screen the first 20 lines of the file form.doc. The next line will repeat the previous commands and add a long listing (dir) of the file check.sum in the directory /tmp. Note that with PolyShell running the slash (/) is useable for path names.

PolyShell supports a number of environmental variables. If you haven't done so already, you may need to take steps to increase the amount of space that DOS assigns for such variables. For DOS 3.3 this command in your config.sys file will increase the space available from the 160 bytes default value to 1600 bytes; change the 1600 to what you want up to 32768 bytes:

```
shell=c:\command.com /p /e:1600
```

One command not included with PolyShell, but available separately, is awk. The version furnished is based on the 1985 version of awk (nawk for new awk?). A slight drawback for me as I am stuck with the old version elsewhere. Awk is a programming language that easily handles data manipulation. If you program in C, for example, you may find awk very useful for a prototype. A few lines in awk will tell you if the idea works and then it can be ported to C.

Awk programs can be simple but still do a lot. For example, if you had a data file

Figure 6

```
# This shell script creates a new user home directory.
# If the username is not given as an argument, prompt for it.
# Next prompt for the home directory.
# Create the home directory, update the "passwd" file,
# and copy the default profile.sh and alias.sys files
# into the new home directory unless they were already there.
# The location of the /etc directory is taken from the SHETC
# variable, if any.
# Make a local alias so echo does not mess up backslashes.
alias echo echo -r
etc2=${SHETC:-/etc} # etc2 = $SHETC, if any, otherwise "/etc"
etc=${etc2%[\/]} # Strip trailing \ or / from etc.
if [ ! -d "$etc" -a ! -d "$etc2" ]
then
  echo mkuser: fatal error: Can not find etc directory: $etc
  exit 2
fi
# If username was not specified on command line, then prompt for it.
username=$1
if [ -z "$username" ]
then
  echo -n "'Enter user name, 8 characters maximum: "
  read username
fi
# Make sure we have a username.
if [ -z "$username" ]
then
  echo mkuser: fatal error: no user name specified.
  echo mkuser: exiting without changing anything.
  exit 2
fi
# Default directory is in /usr on current disk drive.
homedir=$dd:/usr/$username
echo -n "Shall I make the home directory \"$homedir\"? (y or n): "
read response
# If operator does not like it, let him change it.
if [ "$response" != y ]
then
  echo -n "Enter the home directory for user \"$username\": "
  read homedir
  if [ -z "$homedir" ]
  then
     echo mkuser: fatal error: no home directory specified.
     echo mkuser: exiting without changing anything.
     exit 2
  fi
fi
echo;echo "Making new user \"$username\" with home directory \"$homedir\""
# Let operator abort if he wants.
echo;echo -n 'Is this ok? (y or n): '
read response
if [ "$response" != "y" ]
then
  echo mkuser: exiting without changing anything.
  exit 1
fi
# Update the passwd file.
if [ -f $etc/passwd ]
```

```
then
  grep -v ''^${username};   $etc/passwd > $etc/passwd.tmp
  mv $etc/passwd.tmp $etc/passwd
fi
echo "${username};;;;;${homedir};sh" >> $etc/passwd
echo;echo "The $etc/passwd file has been updated."
# Special case for help files: Do not copy profile.sh or alias.sys
# files.
if [ "$username" = "help" ]; then exit; fi
# Strip off a trailing slash from the home directory specified.
# This allows the home directory to be the root directory.
roothomedir=${homedir%[\/]}
# Suppress warning message from mkdir in case directory already exists.
mkdir ${roothomedir} >& /dev/null
# Update the profile.sh file only if they do not already have one.
# Rstatus is set to "error" if cp fails.
rstatus=ok
if [ ! -f $roothomedir/profile.sh ]
then
  if cp $etc/profile.def $roothomedir/profile.sh
  then echo "A new profile.sh file has been added"
  else rstatus=error
  fi
fi
# Update the alias file only if they do not already have one,
# or if they say it is ok.
if [ ! -f $roothomedir/alias.sys ] || (
  echo -n 'Shall I update the alias.sys file? (y or n): '
  read response
  [ "$response" = "y" ] )
then
  if cp $etc/alias.def $roothomedir/alias.sys
  then echo "A new alias.sys file has been added"
  else rstatus=error
  fi
fi
if [ "$rstatus" != "error" ]
then
  echo mkuser: Successfully created new user \"$username\"
fi
```

of employee names, pay rates, and hours worked, in that order and separated by tabs, this one liner would show you who worked during the period and how much they earned.

```
awk -F'\t' '$3 > Ø { print $1, $2 * $3 }
                                      ' emp.dat
```

What does all this cost? The list price for PolyShell is $99. I purchased PolyShell and awk directly from Polytron on an offer of both for $150. Awk is normally available separately for $99.

If you are new to UNIX shell programming, I have found the book "UNIX Shell Programming" by Kochan and Wood, to be a valuable aid. One topic missing is any assistance on awk. However, Polytron furnishes the book by Aho, Kernighan and Weinberger (the authors of awk) with the program.

PolyShell
POLYTRON Corporation
1700 NW 167TH Place
Beaverton, OR 97006
(503) 645-1150

UNIX Shell Programming
Stephen G. Kochan and Patrick H. Wood
Hayden Books
ISBN 0-8104-6309-1

The AWK Programming Language
Alfred V. Aho, Brian W. Kernighan and Peter J. Weinberger
Addison-Wesley Publishing
ISBN 0-201-07981-X

```
  .awk awk -f
  a: cd a:
  b: cd b:
  c: cd c:
  d: cd d:
  e: cd e:
  f: cd f:
  g: cd g:
  L ls -aF
  LL ls -l
  LR ls -R
  H history
  TH history | tail $1
  HH history | head $1
  RH history | tail -r $1
  GH history | grep $1
  strings grep -B ....
  num cat -n
  r fc -e -
  pwd echo $dd:$pwd
  sh history -W; sh.exe
  rehash alias -R; PATH="$PATH"
  hash set -h;'alias' -t
  logout exit Ø
  cls ${SHCOMSPEC:-$COMSPEC} /c cls
  copy ${SHCOMSPEC:-$COMSPEC} /c copy
  dir ${SHCOMSPEC:-$COMSPEC} /c dir
  del ${SHCOMSPEC:-$COMSPEC} /c del
  delete ${SHCOMSPEC:-$COMSPEC} /c del
  erase ${SHCOMSPEC:-$COMSPEC} /c erase
  ren ${SHCOMSPEC:-$COMSPEC} /c ren
  rename ${SHCOMSPEC:-$COMSPEC} /c ren
  type ${SHCOMSPEC:-$COMSPEC} /c type
  ver ${SHCOMSPEC:-$COMSPEC} /c ver
  verify ${SHCOMSPEC:-$COMSPEC} /c verify
  vol ${SHCOMSPEC:-$COMSPEC} /c vol
```

Figure 7

GONE, BUT NEVER FORGOTTEN:

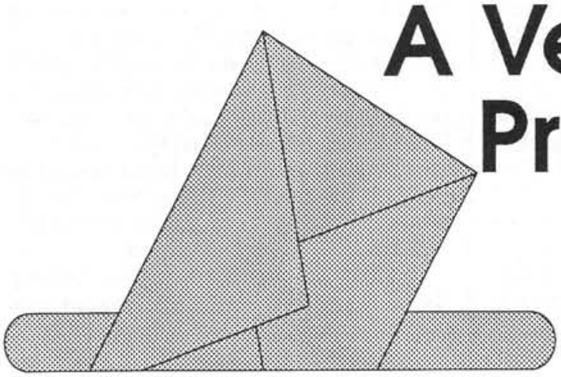
MA HUG RETIRES

As some of you have recently discovered Margaret Bacon, HUG secretary for almost the entire life of HUG, retired on July 6th. Margaret will be back home in Indiana (There's a song in that, I just know it!) by the time you read this. She plans to spend a lot of time taking care of her mother, playing with her computer, and just taking life easier.

To many of us, HUG and Margaret were interchangeable terms. She practically ran the user's group for many years, answering correspondence, taking orders, and keeping HUGgies up on the latest news and changes. If you needed to know something about products or software, current or out-of-date, she knew what you needed to know or where to find it. She was a walking information center for just about everything related to HUG, Heath Company and computers. She will, indeed, be greatly missed.

GOOD LUCK, MARGARET!!

Please keep in mind that, with Margaret gone, it will take some time for a new secretary to be hired and trained. We ask and appreciate your patience and understanding while all of this takes place.



A Versatile Mailing List Program in dBase®

Pat Swayne
Hug Software Engineer

A friend of mine who runs his own business (a cleaning service) recently decided to computerize his operation. I helped him pick out a nice 12 MHz '286 system from our HUG BBS "Bargain Centre", and told him I would help him set it up. I wasn't really sure what he wanted until he showed me a stack of 4x6 cards and said, "To start with, we want to put these in the computer." These cards represented his regular customers, which were other small businesses that had their facilities cleaned at regular intervals. On each card was the name, address, and phone number of a customer, and a description of the job (or jobs) to be done for that customer, along with the charges for each job. Neither my friend nor his wife (who would be the one to actually use the computer) are "power users", so whatever I set up for them would have to be fairly easy to use.

What my friend needed was a mailing list type of database program that could be modified to accept a job description as well as the usual information (name, address, etc.). Before I go any farther, allow me to define a couple of terms for those of you who may be unfamiliar with database programs. A database program is a kind of program that can store and organize the kind of information my friend had on his 4x6 cards. In database program terminology, each of his cards is called a "record", and each item on a card (name, address, job, etc.), is called a "field".

I already had one program that I knew would come close to meeting my friends requirement. Believe it or not, it's an old CP/M (8-bit) program called MailPro, which used to be sold in the Heathkit catalog. You can run MailPro on an MS-DOS computer using my CP/EMulator program (HUG p/n 885-6002). I actually used MailPro myself this way recently when my church needed a mailing list. However, MailPro is limited in the number of characters that each record can have, and it is also limited in the total number of records, since it does all sorting entirely in memory.

I looked at the various "shareware" mailing list programs offered for download on our BBS at the time, and found none of them to be suitable, because most of them are too rigid in the way their fields are set up. I did not find any that would let me set up a field called "Job" that would hold enough characters for a decent job description. Most mailing list programs do not even let you change the names or sizes of the fields at all.

I knew that a general database program such as dBase III Plus could easily do what my friend needed, but I also knew that teaching him to use dBase was something that I did not feel up to doing. Fortunately, dBase III Plus (and its clones such FoxBase+ and the new FoxPro) is not just a database program, but also an interpreter for a computer language which also happens to be called dBase. So the answer was to write a mailing list program in dBase that was as easy to use as any mailing list program, but that could be modified to meet special requirements such as my friend's.

The DBMAIL Program

The result is my program, DBMAIL.PRG, which is listed with this article. The program is also available for download from the HUG BBS as DBMAIL.ZIP. The secret of the program's versatility is that it uses a default database structure, mailing label format, and report format (which can be used for phone book-type lists). When you run the program, it will ask for the name of your mailing list. If you give it a new name, it copies the default structure and formats to files with your name, and then gives you the change to modify things to your liking. If you are typing in the program from this article, you will also need to create the default files.

Creating the Default Files

To create the structure file, start dBase (or FoxBase+ or FoxPro). If the dBase Assistant is on, press Esc to get out of it, and type

```
CREATE DBMAIL
```

and press Return. Then just fill in the field names, types, and widths for 10 fields using the information in this table.

Field	Field Name	Type	Width
1	FIRST_NAME	Character	20
2	LAST_NAME	Character	20
3	ADDRESS_1	Character	30
4	ADDRESS_2	Character	30
5	CITY	Character	15
6	STATE	Character	2
7	ZIP	Character	10
8	PHONE_NO	Character	14
9	ADDITIONAL	Character	30
10	COMMENTS	Memo	10

Keep in mind that you will have a chance to modify these when you use the program, so enter them just as they are here, for now. When you get to the COMMENTS field, and set the type to Memo, the width will be filled in with 10 automatically, and cannot be changed. (The actual width is not 10, because a memo field can actually hold up to 5000 characters, and it is here that my friend can put his job descriptions. He can even change the name of the field to JOB as you will see later.) When you have selected Memo as the type for the COMMENTS field, type Control-W (hold down Ctrl and type W) to cause the structure file to be written to disk. Answer N to the question "Input data records now?". Do not exit from dBase (or whatever) at this time.

To create the default label format file, type

```
CREATE LABEL DBMAIL
```

and press Return. Leave the height, width, left margin, lines between, spaces between, and number of labels across at the default setting for now. Go to the Label Contents screen (press the right arrow key with dBase, or Page Down with FoxBase+), and enter these lines

```
FIRST_NAME,LAST_NAME
```

```
ADDRESS_1
ADDRESS_2
TRIM(CITY)+", "+UPPER(STATE),ZIP
ADDITIONAL
```

Type Control-W to write the label format file to disk. To create the default report format file, enter

```
CREATE REPORT DBMAIL
```

Note: I am not familiar enough with FoxPro, having only used a demo version with limited documentation, to tell you how to create a report format. Hopefully there will be enough information in these instructions for FoxBase+ and dBase users to help you create your file. If not, download DBMAIL.ZIP, which has the default files ready made.

In the Options menu in dBase, or on the Page heading screen in FoxBase+, enter the page title or page heading as

```
Name and Address List
```

Set the left margin to 0, because the default format will be a full 80 columns wide. You can leave the other settings on this screen as is, but I would recommend setting "Page eject before printing" to No. Go to the Groups menu in dBase, or press Page Down with FoxBase+, and enter the Group on expression as

```
LEFT(LAST_NAME,1)
```

Leave the other items unchanged. Go to the Columns menu in dBase, or press Page Down again in FoxBase+, and enter three columns using this information.

first column:
Contents: `TRIM(LAST_NAME)+`
  `", "+FIRST_NAME`
Heading: `Name`
Width: `23`

second column:
Contents: `TRIM(ADDRESS_1)+`
  `IIF(LEN(TRIM(ADDRESS_2))>0,", "+`
  `TRIM(ADDRESS_2),"")+", "+`
  `TRIM(CITY)+", "+STATE+" "+ZIP`
Heading: `Address`
Width: `40`

third column:
Contents: `PHONE_NO`
Heading: `Phone no.`
Width: `14`

Enter the contents of each column as one line, even though they are shown as more than one line (due to width constraints). dBase will scroll sideways while you type a long line, and FoxBase+ will automatically wrap long lines. Type Control-W to write the report format file to disk.

If you have followed these instructions for creating the default files correctly, you will have a file called DBMAIL.DBF, which is the default structure file; a file called DBMAIL.LBL, which is the default label format file; and a file called DBMAIL.FRM, which is the default report format file. There will also be a file called DBMAIL.DBT, which is also part of the default structure. Files with a .DBT extension are created only if there is a field in a structure that is a memo type field. The default files will be in the directory that is the default while you run your database program, which is where they should be. The program itself, DBMAIL.PRG, can be in any directory, as can the data files you create with the program.

Using DBMAIL

To run DBMAIL, start your database program (dBase or whatever) and type DO DBMAIL at the command prompt. If DBMAIL.PRG is in a different drive or directory, you can include that before the file name. You can also run DBMAIL directly from the MS-DOS command prompt, if you enter

```
DBASE DBMAIL
```

and press Enter or Return. Of course, this example shows how you would do it with dBase, but you can do the same thing with FoxBase+ or FoxPro. When DBMAIL starts up, it will ask you for the name of your mailing list. If you are starting a new list, make up a name that will suit the contents of your list. For example, if you are listing family members, you can call the list FAMILY. The name should be a legal MS-DOS file name, without an extension, and it can be preceded by a drive and/or directory designation.

If you are creating a new list, DBMAIL will prompt you with

```
That file does not exist. Do you
want to create one? (Y/N)
```

If you type N at this point, DBMAIL will just exit to MS-DOS. If you type Y, it will create new .DBF, .DBT, .LBL, and .FRM files using your file name, which contain information from the default files. Then it will prompt with

```
Do you want to modify the database
structure? (Y/N)
```

If you type Y, DBMAIL will run the Modify Structure command of your database program, which will allow you to customize your mailing list. For example, you can change the name of the field COMMENTS to JOB, as I mentioned previously. You can even delete or add fields, or change the sizes of fields. You could, for example, add a field called COMPANY between the LAST_NAME and ADDRESS_1 fields. If you are making a list of Canadian addresses, you can change ZIP to POSTAL, change STATE to PROVINCE, and make the PROVINCE field larger (if a two character abbreviation is not enough). As you can see, DBMAIL is far more versatile than most mailing list programs. Just make sure that the first 9 fields are the ones that you would want to sort or search your database by. You will see why later.

After you modify your new mailing list, or type N to the modify question, DBMAIL will display its main menu, which looks like this.

```
DBMAIL Mailing List Program

Menu of functions:

1.  Add new records.
2.  Sort records.
3.  Search and Examine/Modify.
4.  Purge deleted records.
5.  Print mailing labels.
6.  Print phone book.
7.  Modify data structure
8.  Quit.

Enter your choice:
```

To select an item from the menu, just type its number. The first item you will want to run with a new list is no. 1, Add new records. This entry runs the Append command of your database program. It is basically a full screen editor for entering data into each field of each new record you add. The names of each field will be listed, with blank space in reverse video beside each name. The sizes of the blank spaces indicate how may characters can be entered into each field. When you have filled in all of the fields you want to for a particular record (you do not have to fill them all in), you can type Page Down to bring up a new empty record to fill in, or either Control-W or Control-End to save the current record and exit. If you want to exit without saving the current record you are filling in, press Escape.

After you have entered some records, the next thing you will likely want to do is sort them some how, which is the next item on the main menu. If you select item 2, DBMAIL will display

```
How do you want to sort the
records?

1.  By  FIRST_NAME
2.  By  LAST_NAME
3.  By  ADDRESS_1
4.  By  ADDRESS_2
5.  By  CITY
6.  By  STATE
7.  By  ZIP
```

```
8. By  PHONE_NO
9. By  ADDITIONAL

Enter your choice:
```

Notice that it uses the actual names of the first 9 fields in this menu, so that if you change any of those fields, the changes will show up in this menu. If you select one of these items, the database will be sorted according to what you selected. Usually, you would want to sort by last name, especially if you are going to print a phone book-type listing. If you select this menu or any other menu below the main menu by mistake, you can exit from it just by pressing Enter or Return.

You can sort by more than one item by sorting twice or more. For example, if you want your list sorted by last name, and you want the first names of persons with the same last name to be sorted, just sort by the secondary field first (the first name in this case), and then sort by the primary field. DBMAIL returns to the main menu after a sort is completed.

The third item on the main menu allows you to search for records you have entered previously, and examine or modify them. If you select this item, DBMAIL will display

```
How do you want to search the
records?

1.  By  FIRST_NAME
2.  By  LAST_NAME
3.  By  ADDRESS_1
4.  By  ADDRESS_2
5.  By  CITY
6.  By  STATE
7.  By  ZIP
8.  By  PHONE_NO
9.  By  ADDITIONAL
C.  Show current record.
F.  Go to first record.
L.  Go to last record.

Enter your choice:
```

As with the Sort menu, the first 9 fields of your list are displayed, and you can search by any of them. Additional menu items let you go to the first or last record, or start with the current record. Near the bottom of the screen, a status line shows what the current record is. If you have a list of 100 records, and the record pointer is at the 34th one (perhaps you found that one in a previous search), then 34/100 will be displayed on the status line.

If you select one of the fields to search by, DBMAIL will prompt you to enter a search string. The string can be all or part of what you are searching for. For example, if you are searching for a last name "Thompson", you can just enter "Thom" as the search string. DBMAIL

Below is a listing of the DBMAIL program. Some of the longer lines in this listing were "folded" to allow the listing to fit in this magazine. The folded parts are indented 10 spaces in from the line above, to make them stand out

```
** DBMAIL.PRG
*  A MAILING LIST PROGRAM IN dBASE LANGUAGE
*  By P. SWAYNE  26-APR-90  08-MAY-90

* Set up things

SET STATUS ON
SET TALK OFF
SET BELL OFF
SET ESCAPE OFF
SET EXACT OFF
SET COLOR TO W/B
CLEAR
? SPACE(26)+"DBMAIL Mailing List Program"
? SPACE(33)+"by P. Swayne"
? ""

* Get file to work on

ACCEPT SPACE(22)+"Enter the name of your mailing list: " to mlist
mlist=LTRIM(mlist)
mlistf=mlist+".DBF"
mlabel=mlist+".LBL"
mreport=mlist+".FRM"
IF .NOT. FILE(mlistf)
    ? ""
    WAIT SPACE(12)+"That file does not exist. Do you want to create one?
              (Y/N) " to answer
    IF answer="Y" .OR. answer="y"
        USE DBMAIL
        COPY STRUCTURE TO &mlist
        CLOSE ALL
        COPY FILE DBMAIL.LBL TO &mlabel
        COPY FILE DBMAIL.FRM TO &mreport
        ? ""
        WAIT SPACE(15)+"Do you want to modify the database structure?
                  (Y/N) " to answer
        IF answer="Y" .OR. answer="y"
            USE &mlist
            MODIFY STRUCTURE &mlist
        ENDIF
    ELSE
        QUIT
    ENDIF
ENDIF
USE &mlist

* Main menu loop starts here

DO WHILE .T.  && Loop forever
    CLEAR
    IF RECNO() > RECCOUNT()
        GO BOTTOM
    ENDIF
    ? SPACE(26)+"DBMAIL Mailing List Program"
    ? ""
    ? SPACE(26)+"Menu of functions:"
    ? ""
    ? SPACE(26)+"1.  Add new records."
    ? SPACE(26)+"2.  Sort records."
    ? SPACE(26)+"3.  Search and Examine/Modify."
    ? SPACE(26)+"4.  Purge deleted records."
    ? SPACE(26)+"5.  Print mailing labels."
    ? SPACE(26)+"6.  Print phone book."
    ? SPACE(26)+"7.  Modify data structure."
    ? SPACE(26)+"8.  Quit."
```

will find the first last name that starts with "Thom". You must, however, enter capital and lower case letters correctly.

After DBMAIL finds what you are looking for, it runs the Edit command of your database program. Edit is just like Append, except that the fields are already filled in. You can move the cursor around in the fields using the arrow keys, and type in any changes that are needed. If you type Page Down or Page Up, any changes that you have made will be recorded, and the next or previous record will be displayed. If you want to record some changes to the currently displayed record without paging to another one, type Control-End or Control-W. You can also discard the changes to the current record by typing Escape. When you type Control-End, Control-W, or Escape, DBMAIL returns to the main menu. It will also return to the main menu if you try to Page Down past the last record, or Page Up past the first one.

While you are in the Edit mode, you can mark a record for deletion by typing Control-U (Control-T with FoxPro). When you do this, the word DEL appears on the status line. The delete marker will be recorded if you page to another record, or exit with Control-End or Control-W. The record is not deleted at this point, but just marked for deletion.

The fourth item on the main menu is used to permanently remove any records that have been marked for deletion. If you select this item, DBMAIL asks "Are you sure you want to purge?", and if you type Y, it removes the files that have been marked for deletion. If no files have been marked, the list is not changed.

When you want to print mailing labels from your list, select the fifth item on the main menu. When you select this item, DBMAIL displays

```
Which records do you want to print
    to mailing labels?

1.  All records.
2.  The next n records starting
    with the current one.
3.  From the current record to
    the last record.
4.  Modify the label format.

  Enter your choice:
```

The first thing you should do if you have not printed labels before from a particular list is to select item 4 from this menu. This calls up the Modify Label command of your database program, which is just like the Create Label command you used to make the default label format file. You can now change the height, width, left margin, lines between, spaces between, and number of labels across to fit the labels you will use with this list. If you have changed the names

```
    ? ""
    WAIT SPACE(26)+"Enter your choice: " to choice
    DO CASE

* Add new records

    CASE choice="1"
        APPEND

* Sort the records

    CASE choice="2"
        CLEAR
        ? SPACE(22)+"How do you want to sort the records?"
        ? ""
        ? SPACE(26)+"1.  By ",FIELD(1)
        ? SPACE(26)+"2.  By ",FIELD(2)
        ? SPACE(26)+"3.  By ",FIELD(3)
        ? SPACE(26)+"4.  By ",FIELD(4)
        ? SPACE(26)+"5.  By ",FIELD(5)
        ? SPACE(26)+"6.  By ",FIELD(6)
        ? SPACE(26)+"7.  By ",FIELD(7)
        ? SPACE(26)+"8.  By ",FIELD(8)
        ? SPACE(26)+"9.  By ",FIELD(9)
        ? ""
        WAIT SPACE(26)+"Enter your choice: " to schoice
        IF val(schoice) > 0
            STORE FIELD(VAL(schoice)) TO stype
            ? SPACE(26)+"Sorting..."
            ? ""
            DELETE FILE STEMP.DBF
            COPY TO STEMP           && Create temporary files
            IF FILE ('STEMP.DBF')   && Make sure it's there!
                CLOSE ALL
                USE STEMP
                DELETE FILE &mlistf  && Delete user's file
                SORT TO &mlist ON &stype  && Make new one with sort
                CLOSE ALL
                USE &mlist
            ELSE
                ? ""
                WAIT "Disk error!  Cannot sort.  Press any key."
            ENDIF
        ENDIF

* Search the records

    CASE choice="3"
        CLEAR
        ? SPACE(22)+"How do you want to search the records?"
        ? ""
        ? SPACE(26)+"1.  By ",FIELD(1)
        ? SPACE(26)+"2.  By ",FIELD(2)
        ? SPACE(26)+"3.  By ",FIELD(3)
        ? SPACE(26)+"4.  By ",FIELD(4)
        ? SPACE(26)+"5.  By ",FIELD(5)
        ? SPACE(26)+"6.  By ",FIELD(6)
        ? SPACE(26)+"7.  By ",FIELD(7)
        ? SPACE(26)+"8.  By ",FIELD(8)
        ? SPACE(26)+"9.  By ",FIELD(9)
        ? SPACE(26)+"C.  Show current record."
        ? SPACE(26)+"F.  Go to first record."
        ? SPACE(26)+"L.  Go to last record."
        ? ""
        WAIT SPACE(26)+"Enter your choice: " to schoice
        schoice=UPPER(schoice)
        DO CASE
        CASE schoice="C"
            EDIT
        CASE schoice="F"
```

of any of the fields that are used on the label, you will have to change them here also. When you exit by typing Control-W (to save your changes) or Escape (to discard them), DBMAIL returns to the mailing label menu. You can now print all or part of the records in the list, as indicated by the menu. If you select item 2, you will be asked for the number of records to print.

After you select item 1, 2, or 3, DBMAIL will ask "Do you want to print samples?". If you type Y, DBMAIL will print one row of labels with all printable fields filled in with asterisks. You can adjust your printer if the labels are not printed as they should be. After DBMAIL prints the samples, it will ask if you want to print more samples. You can keep printing samples and adjusting your printer until the labels look they way they should. However, if the printing on your labels is way off, you should probably modify the label format again.

After you answer N to the question to print samples, DBMAIL will print the records you have requested onto your labels. If you have a printer problem and the system seems to lock up, wait a little while and then press Escape.

To print the information in your list in a phone book format, select item 6 from the main menu. When you do, DBMAIL displays

```
Which records do you want to
    print to the phone book?

1.  All records.
2.  The next n records starting
    with the current one.
3.  From the current record to
    the last record.
4.  Modify the phone book format.

Enter your choice:
```

This menu works just like the mailing label menu, except that item 4 runs the Modify Report command, and it does not print samples. As with the mailing labels, you may have to modify the phone book format the first time you want to print a particular list.

Item 7 from the main menu lets you change the data structure of a list after you have created it. If you created a list of names and addresses for mailing labels, and you later wanted to add a field called "COMPANY", you could use this menu selection to add it. This selection runs the Modify Structure command of your database program. Be sure to read the Modify Structure section of your database program manual before you use this feature to modify the structure of a database that already contains data.

When you want to quit using DBMAIL, select item 8 from the main menu. Do not quit by turning off your computer without using this selection, because recently added or changed data may be lost.

```
            GO TOP
            EDIT
        CASE schoice="L"
            GO BOTTOM
            EDIT
        ENDCASE
        IF val(schoice) > 0
            STORE FIELD(VAL(schoice)) TO stype
            ? ""
            ACCEPT SPACE(26)+"Enter search string: " to sname
            ? ""
            ? SPACE(26)+"Searching..."
            LOCATE FOR &stype=sname
            EDIT
        ENDIF

* Purge deleted files

    CASE choice="4"
        CLEAR
        WAIT SPACE(22)+"Are you sure you want to purge? (Y/N) "
            TO answer
        IF answer="Y" .OR. answer="y"
            ? ""
            ? SPACE(26)+"Purging deleted files..."
            ? ""
            PACK
        ENDIF

* Print mailing labels

    CASE choice="5"
        STORE "0" TO schoice  && Allow loop if modifying format
        DO WHILE schoice="0"
        CLEAR
        ? SPACE(13)+"Which records do you want to print to mailing
                  labels?"
        ? ""
        ? SPACE(13)+"1.  All records."
        ? SPACE(13)+"2.  The next n records starting with the current
                  one."
        ? SPACE(13)+"3.  From the current record to the last record."
        ? SPACE(13)+"4.  Modify the label format."
        ? ""
        WAIT SPACE(13)+"Enter your choice: " TO schoice
        DO CASE
        CASE schoice="0"
            STORE "X" TO schoice  && Don't loop on 0 input
        CASE schoice="1"
            STORE "ALL" TO stype
        CASE schoice="2"
            ? ""
            ACCEPT SPACE(13)+"Enter number of records to print: " to
                      pnumber
            STORE "NEXT"+" "+pnumber TO stype
        CASE schoice="3"
            STORE "REST" TO stype
        CASE schoice="4"
            MODIFY LABEL &mlabel
            STORE "0" TO schoice
        OTHERWISE
            LOOP
        ENDCASE
        IF VAL(schoice) > 0
            ? ""
            WAIT SPACE(13)+"Do you want to print samples? (Y/N) " to
                      answer
            ? ""
            ? SPACE(13)+"Printing..."
            SET CONSOLE OFF
            STORE RECNO() TO savrec
            IF answer="Y" .OR. answer="y"
                LABEL FORM &mlabel &stype SAMPLE TO PRINT
            ELSE
                LABEL FORM &mlabel &stype TO PRINT
            ENDIF
            GO savrec
            SET CONSOLE ON
        ENDIF
        ENDDO

* Print phone book

    CASE choice="6"
        STORE "0" TO schoice  && Allow loop if modifying format
        DO WHILE schoice="0"
        CLEAR
        ? SPACE(13)+"Which records do you want to print to the phone
                  book?"
        ? ""
        ? SPACE(13)+"1.  All records."
        ? SPACE(13)+"2.  The next n records starting with the current
                  one."
        ? SPACE(13)+"3.  From the current record to the last record."
        ? SPACE(13)+"4.  Modify the phone book format."
        ? ""
        WAIT SPACE(13)+"Enter your choice: " TO schoice
        DO CASE
        CASE schoice="0"
            STORE "X" TO schoice  && Don't loop on 0 input
        CASE schoice="1"
            STORE "ALL" TO stype
        CASE schoice="2"
            ? ""
            ACCEPT SPACE(13)+"Enter number of records to print: " to
                      pnumber
            STORE "NEXT"+" "+pnumber TO stype
        CASE schoice="3"
            STORE "REST" TO stype
        CASE schoice="4"
            MODIFY REPORT &mreport
            STORE "0" TO schoice
        OTHERWISE
            LOOP
        ENDCASE
        IF VAL(schoice) > 0
            ? ""
            ? SPACE(13)+"Printing..."
            SET CONSOLE OFF
            STORE RECNO() TO savrec
            REPORT FORM &mreport &stype TO PRINT
            GO savrec
            SET CONSOLE ON
        ENDIF
        ENDDO

* Modify structure

    CASE choice="7"
        CLEAR
        ? SPACE(13)+"Caution:  Do not change a field's name and its
                  width"
        ? SPACE(13)+"or type at the same time.  See your database
                  program"
        ? SPACE(13)+"manual for more information."
        ? ""
        WAIT SPACE(13)+"Are you sure you want to modify the structure?
                  (Y/N) " TO answer
        IF answer="Y" .OR. answer="y"
            MODIFY STRUCTURE
        ENDIF

* Exit

    CASE choice="8"
        EXIT  && Use RETURN here
                  to stay in dBase
    OTHERWISE
        LOOP
    ENDCASE
ENDDO
QUIT
```

Single-Byte Integers in BASIC

Example: Writing Character Length Files for WatchWord

Kenneth D. Granzow
1079 Haverhill Place
Colorado Springs, CO 80919

Viva BASIC!

I don't know about you, but when I need a special purpose program to use once or twice, I usually write it in BASIC. It may not give the fastest running code, but I find that I can write more correct, functioning code faster in BASIC than in any other language. And if it's only to run once or twice (or more often for small jobs), it doesn't matter how slowly the code executes (of course, execution can be speeded up drastically by compiling and linking to create an EXE file).

Single-Byte Integers are Not Supported in BASIC

Occasionally, these small jobs involve reading or creating files that contain single-byte integers. These files are created or used by software not written in BASIC. Single-byte integer files are often used for storage of various kinds of data. Even though single-byte integers are not supported in BASIC, reading or creating such files using BASIC (in a somewhat non-standard way) is very easy.

Remember WatchWord?

The application discussed in this article relates to files used by WatchWord; yes, the word processor originally written for the Z-100 (not PC) and available for PC compatibles also. I still use WatchWord on both my Z-100 and on a Z-248/12 (this article is being written using WatchWord on the Z-100).

My current need for single-byte integer files was triggered by the purchase of an HP LASERJET IIP printer and the TextEquations font cartridge. This cartridge contains the CG TIMES font that is a proportionally spaced font. For WatchWord to format this font, it must have tables containing the length of each character. WatchWord comes with files containing this information for some fonts (HP cartridges B and F), but not for the CG TIMES font. Hence, I set out to create my own character length files for use by WatchWord.

Using Single-Byte Integers in BASIC

BASIC does not have variables that it calls single-byte integers, but in effect, it has them. They are single-byte strings (or characters). It can not do integer arithmetic directly on them, but they can easily be converted to normal 2-byte integers; any arithmetic can then be performed, and then the results can be converted back to single-byte strings if needed. Not something you'd want to do in BASIC just as a matter of course, but if you want to deal with single-byte integers to interface other software, it can easily be done.

The standard unsigned value of a single byte is given by the value of bit zero plus 2 times the value of bit 1 plus 4 times the value of bit 2, etc., through bit 7. This value of a single byte is exactly the ASCII value of the character represented by the byte. Hence, if a single byte is read as a character from a file of single-byte integers, the conversion to a normal 2-byte BASIC integer is performed with the function ASC(B1$); where B1$ is the single-byte character. The value returned by ASC will be in the range 0 to 255. Conversely, given an integer, say X in the range 0 to 255, it can easily be converted to a single-byte string (a character) using the BASIC function CHR$(X).

Often single-byte integers are used to represent a range containing both positive and negative values. This is usually done using a bias. That is, the number to be represented is the ASCII value minus a known constant; if the constant is 128, the number represented is in the range -128 to +127. Furthermore, scale factors may be used so that the single byte can represent numbers in essentially any range. The accuracy of such numbers is, of course, limited by the single-byte representation.

Output of strings of such single-byte integers to disk files requires some special care. One cannot use the BASIC command:

```
PRINT #f, list;
```

or the similar

```
PRINT #f, USING string; list;
```

as one would think. The reason is that, when the list contains non-printing characters, often the special characters are "trapped" and used for control of some sort. For example, ZBASIC will not transmit character 9 to disk using the above statements. Instead, interpreting character 9 to be a horizontal tab, it transmits 8 blanks (character 32's) instead; this is disastrous in a numerical application. Using QuickBASIC, whenever a character 13 (carriage return) is encountered, QuickBASIC inserts a character 10 (line feed) following it.) Therefore, when writing files that may contain non-printing ASCII characters, a binary file must be opened to insure that the desired string of characters is written to the file. There are several ways of doing this. An example is given in Program 1; it will be discussed later.

Back to WatchWord

For WatchWord to format proportionally spaced fonts, it must have tables containing the length of each character (see WatchWord, Version 3, Manual, Chapter 22). These files contain 256 bytes that represent the lengths of ASCII characters 0 through 255 (in that order). Font CG TIMES (on the HP TextEquations cartridge) consists of four character sets: 10 point upright, 10 point italic, 10 point bold, and 8 point upright; the character lengths are different for each of these. Using Program 1, I generated four character length files, one for each of these charac-

Program 1
PRPGEN.BAS

```
10 ' PROGRAM TO GENERATE CHARACTER LENGTH FILES FOR PROPORTIONAL FONTS
20 ' FOR WATCHWORD
30 '
40 ' INPUT (INTERACTIVE): The length in (inches, 16ths) of string of 48
50 ' of each printable character.
60 ' Name of file to be generated.
70 '
80 ' OUTPUT: A 256 byte file
90 '          THE BYTES CORRESPOND TO ASCII CHARACTERS 0 TO 255
100 '         The value is the width of that character in 300ths of
110 '         an inch.
120 '
130 DIM CT%(256)  'Define 2-Byte integer array
140 CLS
150 PRINT"       GENERATION OF FONT WIDTH TABLE FOR WATCHWARD"
160 PRINT
170 INPUT "Enter name of OUTPUT file to be generated: "; NFL$
180 OPEN "R", #1, NFL$, 128 : FIELD #1, 128 AS CC$     'Open output file
190 PRINT
200 PRINT "All values are entered as integer inches and 16ths"
210 PRINT "The length of a string of 48 of each printable character."
220 PRINT "They are entered as a pair"
230 PRINT
240 INPUT "Enter value for blanks (inches, 16ths): ",NCH, SIXT
250 WDTH = 300 * (NCH + SIXT/16)/48
260 'Use width of blanks for unprintable characters
270  FOR I% = 0 TO 32
280    CT%(I%) = WDTH
290  NEXT I%
300  FOR I% = 128 TO 160
310    CT%(I%) = WDTH
320  NEXT I%
330  PRINT
340  PRINT "Printable Characters"
350  FOR I% = 33 TO 127
360    PRINT "Character no.";I%;",   ";CHR$(I%)
370    INPUT "Enter value (inches, 16ths): ", NCH, SIXT
380    WDTH = 300 * (NCH + SIXT/16)/48
390    CT%(I%) = WDTH
400  NEXT I%
410  PRINT "For the following, the character printed on the screen will"
420  PRINT "not correspond to the one printed by the printer."
430  PRINT "Use ASCII no. to correctly identify it."
440  FOR I% =161 TO 255
450    PRINT "Character no.";I%;",   ";CHR$(I%)
460    INPUT "Enter value (inches, 16ths): ", NCH, SIXT
470    WDTH = 300 * (NCH + SIXT/16)/48
480    CT%(I%) = WDTH
490  NEXT I%
500 '
510  PRINT "All data has been entered--beginning file creation."
520  A$ = ""
530  FOR I% = 0 TO 127
540   A$ = A$ + CHR$(CT%(I%))
550  NEXT I%
560  LSET CC$ = A$
570  PUT #1
580  A$ = ""
590  FOR I% = 128 TO 255
600   A$ = A$ + CHR$(CT%(I%))
610  NEXT I%
620  LSET CC$ = A$
630  PUT #1
640  CLOSE #1
650  STOP
```

ter sets.

Each byte of a WatchWord character length file contains the length of a character in units of microspaces. A microspace is one 300th of an inch for the LaserJet printers; this is the resolution of the LaserJet. Each byte is a single-byte integer giving the length in microspaces without any bias or scale factor. Since measuring the length of a single character to a 300th of an inch is something I don't know how to do, I wrote a short program (not given) to cause the LaserJet to print a string of 48 of each of the printable characters in each character set. Forty-eight of the widest characters fit easily across the width of a page, hence, a string of 48 provides a reasonable length string to measure with a ruler. For the spaces, I printed 48 of them followed by a printing character to mark the end of the 48 spaces. The input to Program 1 is the length of each of the strings of 48 characters in inches and 16ths of an inch for a character set. (The program was run four times, once for each character set.) Using the nearest 16th of an inch gives accuracy of one 32nd of an inch for 48 characters or an accuracy of about 1/1,536 inches per character. This is more than sufficient accuracy for Program 1 to determine the length of a single character to the nearest 300th of an inch.

Stepping Through Character Length File Generation

Program 1, PRPGEN.BAS, was written and run on the Z-100 using ZBASIC. There is no reason why it would not run, as is, using QuickBASIC, but I've not tried it. The program begins by dimensioning an array (CT%) of 256 (2-byte) integers. The character lengths, rounded to the nearest 300th of an inch, are loaded into this array as the physical lengths of the strings of 48 characters and are entered into the program. At the end of the program the array CT% is converted to single-byte strings and written to disk. The "lengths" of non-printing characters are filled out with the length of a space. This is what I found in the WatchWord supplied character length files.

Statements 140 and 150 clear the screen and print the program's function. In statements 170 and 180 the user enters the name of the output file and the file is opened. Note that 180 opens a random access (binary) file. This file will faithfully record any bytes contained in the string CC$ without trapping control characters. The string length 128 is the longest allowable in ZBASIC. Two PUTs are used to write 256 bytes to the file.

Statements 240 to 490 contain the data entry (NCH = number of inches, SIXT = number of 16ths of an inch). These are combined to yield WDTH; it is the unrounded length of the character in 300ths of an inch. WDTH is rounded to the nearest integer and stored in CT% (an integer array).

In statements 520 to 550, the first 128 integers in the array CT% are converted to single-byte characters and concatenated into the string A$. In statements 560 and 570, the string A$ is written to the disk file #1. Statements 580 to 630 perform the same operations for the second 128 integers completing the file creation of 256 bytes. The output file is then closed and the program ends.

The files created with PRPGEN work fine in WatchWord. And the output of the new HP LaserJet IIP is superb.

Program 2
PRPLST.BAS

```
10 ' LIST BYTES IN A CHARACTER LENGTH FILE
20 '
30 ' ENTER FILE NAME
40 CLS
50 INPUT "ENTER FILE NAME: ",NFL$
60 ON ERROR GOTO 150
70 OPEN "I", #1, NFL$
80 ON ERROR GOTO 0
90 CLOSE #1
100 OPEN "R", #1, NFL$, 1 : FIELD #1, 1 AS X$
110 LPRINT "FILE NAME IS: ";NFL$  : LPRINT
120 GOTO 200
130 '
140 'ERROR HANDLING
150 RESUME 160
160 ON ERROR GOTO 0
170 PRINT"FILE ";NFL$;" NOT FOUND, PLEASE RE-ENTER"
180 GOTO 50
190 '
200 FOR I% = 1 TO 32
210  N1% = I%       : GET #1, N1% : C1% = ASC(X$) : N1% = N1% - 1
220  N2% = I% + 32 : GET #1, N2% : C2% = ASC(X$) : N2% = N2% - 1
230  N3% = I% + 64 : GET #1, N3% : C3% = ASC(X$) : N3% = N3% - 1
240  N4% = I% + 96 : GET #1, N4% : C4% = ASC(X$) : N4% = N4% - 1
250  N5% = I% + 128: GET #1, N5% : C5% = ASC(X$) : N5% = N5% - 1
260  N6% = I% + 160: GET #1, N6% : C6% = ASC(X$) : N6% = N6% - 1
270  N7% = I% + 192: GET #1, N7% : C7% = ASC(X$) : N7% = N7% - 1
280  N8% = I% + 224: GET #1, N8% : C8% = ASC(X$) : N8% = N8% - 1
290 ' PRINT THE LINE
300 LPRINT USING "### ###  ### ###  ### ###  ### ###  ### ###  ### ###
     ### ###  ### ###"; N1%,C1%,N2%,C2%,N3%,C3%,N4%,C4%,N5%,C5%,N6%,C6%,N7%,
     C7%,N8%,C8%
310 NEXT I%
320 CLOSE #1
330 LPRINT CHR$(12);
340 STOP
```

What are the Numbers in a Character Length File?

Program 2, PRPLST.BAS, is included to print the 256 character lengths contained in a WatchWord character length file. It can be used on the files generated by PRPGEN, files distributed with WatchWord (.PRP files) or files created some other way.

The program begins by asking for the file name of the input file. Even though the file is to be opened and read as a random access file, an error trap is set and an attempt is made to open the file as an input file. If an error was made in entering the name of the file (and the file isn't found), the program jumps to statement 150 and the operator is requested to reenter the file name. **Note:** If the initial attempt to open the file was as a random access file and an erroneous (non-exitent) file name was entered, no error condition would exist; a new (empty) file would be created and the program would proceed to read garbage from the disk and print it.) Once the name entered is verified to be an existing file, the file is closed as an input file and opened as a random access file. It is then read and the contents printed.

Note that the file is opened with one-byte records. This is done so that the individual bytes can be accessed for multi-column printing. The setup for the printing is done in lines 210 to 280. At the end of each of these lines, 1 is subtracted from N1%, N2%, etc. to convert from record (or byte) number (1 to 256) to ASCII number (0 to 255) for printing. Finally, the file is closed and a formfeed is sent to the printer.

Best wishes for productive and happy byte-pushing in BASIC. ✱

Continued from Page 4

There is, we're glad to report, a happy ending for Mr. Brenner, your readers and us: We're currently working a new version of Disk Technician Advanced which will not only work on partitions greater than 32 mb, but also with ESDI and SCSI drives, and translating controllers. We will, of course, let REMark know when it becomes available.

In the meantime, thanks again to Mr. Brenner and REMark. And continued success.

Sincerely,
Kevin Meza
Prime Solutions
1940 Garnet Avenue
San Diego, CA 92109

Editor's Note: *Doesn't sound very "advanced" to me. I know I sure wouldn't want to partition my 320 mb drive into 10 pieces, just so I could use the product. Mr. Brenner's 'call' was "right on!"* ✱

Assembly Language

Part 7
The Instruction Set (Part 5)
A Sample Program
An Introduction to I/O

Pat Swayne
Hug Software Engineer

For those of you who haven't given up yet, this is another episode in my continuing series on assembly language. In this installment, I will (positively) complete my discussion of the instruction set.

Machine Control Instructions

The last group of instructions is the machine control group. This group includes a few instructions for directly manipulating flags, some other instructions that affect coprocessor operation, and some instructions that just don't fit anywhere else.

Flag Manipulation Instructions

The flag manipulation instructions can be used, as was indicated in the last installment, to set up the flags before you use a RET 2 to return from an interrupt. Here are the flag manipulation instructions.

CLC -- Clear the carry flag.

CMC -- Complement the carry flag. The flag is set if it was clear, or it is cleared if it was set prior to the instruction.

STC -- Set the carry flag.

CLD -- Clear the direction flag. This instruction is used before string manipulation instruction to determine whether the index registers are incremented or decremented. The registers are incremented if the flag is cleared.

STD -- Set the direction flag.

CLI -- Clear the interrupt enable flag. Hardware interrupts are disabled while the flag is clear.

STI -- Set the interrupt enable flag. Enables hardware interrupts. Use this instruction before you exit from an interrupt with RET 2.

Coprocessor Instructions

Here are the instructions that are used in coprocessor operations or multiprocessor operations. We will not cover coprocessors or multiprocessing in this series.

ESC -- This instruction accesses a memory operand and places it on the bus for use by a coprocessor. All 8087 math coprocessor instructions actually begin with this instruction. It usually is not expressed as a mnemonic, but is built into the 8087 instructions.

LOCK -- This instruction is a prefix (like REP or the segment prefixes). It causes the Lock pin on the processor to be brought to its active (low) state until the following instruction is completed. It is a signal to other processors on the bus that the bus is not available.

WAIT -- This instruction causes the processor to wait if the Test pin is low. A coprocessor can use therefore use the Test pin to signal when it is finished with a computation, if the processor must wait for it to finish.

Other Machine Control Instructions

Here are the other machine control instructions.

HLT -- This instruction causes the processor to wait until an interrupt occurs if interrupts are enabled, or until a non-maskable interrupt occurs (a special hardware interrupt that is not affected by the CLI or STI instructions), or until the Reset pin on the processor goes high (which causes the processor to start executing code at F000:FFF0.

NOP -- (No OPeration) This instruction is used whenever you want the instruction pointer to advance to the next address without the processor doing anything.

A Sample Program

Now that I have presented all of the processor's instructions, it is time to do something with them. I will present a small program, and explain each line of it. All this program does is print HELLO THERE on the screen. Those of you who have followed this series will recall that in the first part I said that I could write such a program using 23 bytes, including the message. Well, here it is.

```
CODE     SEGMENT
         ASSUME  CS:CODE,DS:CODE,ES:CODE,SS:CODE
         ORG     100H

START:   MOV     DX,OFFSET MSG      ;POINT TO MESSAGE
         MOV     AH,9               ;DOS CODE TO PRINT MSG
         INT     21H                ;CALL DOS ROUTINE
         INT     20H                ;EXIT TO DOS

MSG      DB      'HELLO THERE',13,10,'$'

CODE     ENDS
         END      START
```

To convert this listing into an executable program, you can enter the following commands at the MS-DOS prompt (assuming that the file containing the above listing is called HELLO.ASM).

```
MASM HELLO;
LINK HELLO;
DEL HELLO.OBJ
EXE2BIN HELLO HELLO.COM
DEL HELLO.EXE
```

The first line of our sample program begins what is called a "segment declaration". In the first line, the label CODE is defined as a segment name. In the second line, the ASSUME directive tells the assembler with which segment to associate each of the segment registers. Since the program is to be assembled into a .COM file, all of the segment registers should point to the segment containing the code at the beginning of the program. The ASSUME directive can be used anywhere in a program to specify the segment of a segment register or registers.

Following the ASSUME directive, the ORG directive is used to specify the origin of the program. Programs that are .COM files must always start at 100 hex.

The next line in the listing is the first machine instruction in the program. The label START is used as a reference point for the END directive at the end of the program, which must have as its argument the starting address of the program. The MOV instruction in this line is used to load the DX register with the starting address of the message HELLO THERE. The program uses a routine built into MS-DOS for printing the message, which requires that the address of the message be specified in the DX register. We will discuss this routine and other I/O methods more later.

The next line loads the AH register with the number 9. MS-DOS routines, such as the one the program uses to print the message, are all accessed via a single software interrupt (in the next line). The value in the AH register is used to specify which routine is to be used.

The next line calls the MS-DOS "service routine" using a software interrupt. MS-DOS contains subroutines (sometimes called "functions") for reading the keyboard, writing to the screen or a printer, reading and writing disk files, allocating memory, and other jobs. However, as we shall see later, there are other ways of doing some of these things besides using the MS-DOS routines. It is the use of these other methods that makes programs dependent on a particular type of computer. For example, it is the reason why programs made for a PC-compatible computer often do not run on a Z-100 series computer.

The next line of the program is the last of the machine instructions. It is a special software interrupt that is used to cause the MS-DOS command interpreter to regain control of the computer. This interrupt is normally used only in .COM files. There is also a function that can be called via INT 21H that can be used to exit from a program, which can be used in both .COM and .EXE files, but it uses more bytes.

At the end of the program, the ENDS directive is used to signal the end of the CODE segment, and the END directive (with its argument) marks the end of the program.

An Introduction to I/O

Before you can start writing assembly language programs, you need to know at least a few ways in which computer software performs I/O (Input/Output). In an MS-DOS computer, there are different ways to get characters on the screen, read the keyboard, send data to a printer, and read or write disk files. The different ways of doing these things each have their advantages. For example, the sample program in this article uses a way to print characters on the screen that is very easy to code, and is compact. However, if you wanted to make the program print characters in a different color from the default screen colors, you would have to use a different method to put the characters on the screen. The MS-DOS routine used in the program knows nothing about screen colors.

The routines in a computer that handle Input/Output are arranged in layers. At the bottom of layer is the hardware itself. Some hardware systems are easier to work with than others. For example, it would be difficult to read a disk file by working directly with the disk controller hardware. On the other hand, you can put characters on the screen in a PC-compatible computer just by placing the ASCII value of the characters in the video controller's memory space. (Actually, for each character there must also be an "attribute byte", which determines the character's color, whether it is blinking or not, etc., and the controller must be in a "text mode", not a graphic mode.) Because it is so easy to put characters on the screen by working directly with the hardware on a PC-compatible computer, many program authors choose to do it that way, and as a result the programs will not run on any computer that is not 100 per cent compatible, including the Z-100 series. The screen on a Z-100 is always in a graphic mode, and in order to put characters on the screen by writing directly to the video controller's memory, you would have to "build" each character pixel by pixel.

The direct hardware method of doing I/O is usually the fastest method. For example, the Fastback™ program works with disk controller directly, which enables it to back up a hard disk to floppy disks must faster than the BACKUP and RESTORE programs that come with MS-DOS can. A program that writes directly to the video hardware can fill the screen with text so fast that it appears to be instantaneous.

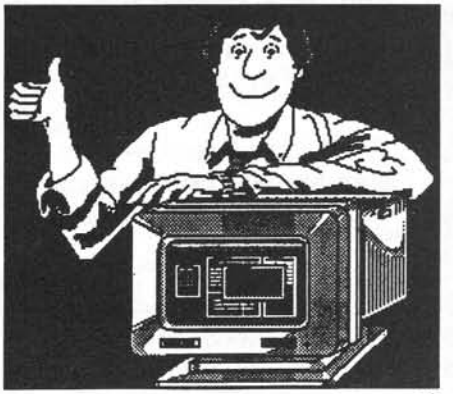
The first layer of Input/Output routines above the hardware in a computer is the BIOS (Basic Input/Output System). The BIOS contains routines for performing all of the normal I/O functions. The BIOS routines in a PC-compatible computer are different from those in a Z-100, so PC program that use BIOS routines, as well as those that work directly with the hardware, are incompatible with a Z-100, and vice versa.

The BIOS routines offer varying degrees of sophistication, depending on what I/O device you are working with. For example, the video routines in the BIOS offer the most sophistication of any layer. You can change video modes, move the cursor, change the shape of the cursor, and control the screen colors by using BIOS routines. The BIOS disk routines, on the other hand, can do little more than read and write absolute sectors on a disk. It would be quite a complicated task to read a disk file into memory using BIOS routines. You would have to determine which sectors on the disk the file occupied before you could begin reading it.

The top layer of I/O routines is the DOS (Disk Operating System), which in our case is MS-DOS. Although MS-DOS contains routines for handling all standard I/O devices, its routines for handling disk I/O are the most sophisticated, which is no doubt why it is called a Disk Operating System. Besides routines for reading and writing files, there are routines for creating and deleting files, finding files, creating and deleting directories, etc. In addition, MS-DOS can treat the other I/O devices as disk files. For example, you can copy data from a disk file to a printer using the same MS-DOS routine that you would use for copying data from one disk file to another.

It is not uncommon for a program to use all three types of I/O. For example, a program may put characters on the screen by writing directly to the video hardware, read the keyboard using a BIOS routine, and read and write disk files using MS-DOS routines. As an assembly language programmer, you are free to use whatever method is best for each situation.

In the next installment of this series, I will show you some other ways to put HELLO THERE on the screen, and how to read the keyboard.



Continued from Page 6

and then to 32 as the need presents itself. I'm told that their board will become the central hub for all Zenith Data Systems support, tech support, communications, sales, HUG support, file uploading and downloading, and more. It will be **the** place to be, and it is being done with Zenith Data Systems' blessings and cooperation. CIS, keep in mind, is not ZDS, but is totally independent.

HUG is constantly working to keep their board up to date and adding new features — something TBBS is good at. That is the place to be — and it's free! I think the least you can all do is if you are not a member of HUG (requirement to get on their board), to join up real soon now if for no other reason, to offer them support and to get on their board! HUG is definitely a nice board with no access fees. All they ask is to be a member of HUG for most things, although anybody can log in for seeing part of the board.

Although I can't go into it all now, some exciting changes will be taking place at what is presently known as Heath Users' Group. I was also told that all this ZDS liquidation crap is being stopped. All merchandise they want to liquidate will be done through their board, with special consideration being given to dealers who were in the past left in the dark. So keep an eye on the HUG bulletin board. If you don't have the number, it's (616) 982-3956.

Continued from Page 8

{ENTER} commands that come after "=A:" can be created in three different ways:

<CTRL-V><ENTER> produces {ENTER}
<CTRL-PgUp><CTRL-PgUp><ENTER>
  produces {RETURN}
<CTRL-PgUp> moves cursor to produce
  {RETURN}
{RETURN} and press <ENTER>

These all produce the same result when the macro executes. Pick one of them and do it twice. Finally, press F7 until the main editing screen is displayed. The macro is done.

This macro could be incorporated into my keyboard redefinition file, but I prefer to leave it as a separate macro. WordPerfect allows the automatic execution of one macro as a part of the loading sequence, and this is my automatic macro. This feature will not work with contents of a keyboard redefinition file.

Temporary Macros

While writing this article, I needed to type <ENTER> many times. I wanted a macro to expedite the process, but I didn't want the macro to use up precious disk space. Once the article was done, I had no further need of the macro. A temporary macro is ideal for this purpose.

The first time it was needed, I actually typed <ENTER>. Next I blocked those seven characters. With the block on I pressed <CTRL-PgUp>. The prompt at the bottom left said "Variable:", and I pressed 1. Blocked text was assigned as the value of the variable named {VAR 1}. When I next needed <ENTER>, I simply pressed <ALT-1> and the desired characters appeared in my document.

Conclusion

WordPerfect macros can save lots of time and keystrokes. The macro feature is so powerful that you can do almost anything you can imagine. Keystroke macros and temporary macros are very easy to create. Command language macros are more complicated, but the power they unleash is impressive.

If you're a serious WordPerfect user, it's worth the time required to learn the macro programming language. Watch for another article dealing with the finer points of this language.

Continued from Page 15

paper.

For example, suppose you used your application program to create a one-inch box. You could do this easily in WordPerfect through the Alt-F9 graphic function, or using line draw in Microsoft Word. When you print the file to a Postscript device, the driver will transmit ASCII text commands similar to those illustrated previously. The only drawback is that most applications cannot take advantage of all of Postscript's power.

Because Postscript programs are text files, you could also write your own program using a word processor or text editor. You save the program as an ASCII file, then download it through the printer's port. The Postscript interpreter will treat the program just as if it was created by an application's driver. While you'll have to know Postscript programming, you'll be able to use all of the printer's capabilities.

The last way to create a program also involves writing it yourself, but in an interactive mode. Some Postscript printers, such as the Apple Laserwriter, can be placed in the interactive mode directly connected to your keyboard. This is normally done using a telecommunications program through the printer's serial port instead of a modem. Every line you type will be transmitted directly to the printer and interpreted immediately.

Entering commands one at a time is helpful when you're first learning Postscript. You can type small programs without having to switch in and out of a text editor.

Postscript is a full programming language designed to harness the power of laser printers. If you're happy with how your application programs utilize that power, then you may never have to write your own Postscript programs. But if you want to create special effects, utilizing every feature that Postscript has to offer, then try your hand at programming.

WordPerfect Executive

Edwin G. Wiggins
13 Clare Drive
E. Northport, NY 11731

An Integrated Package for Laptops

Introduction

Laptop computing has become a way of life for busy people on the move. Some people opt for a full-featured laptop, including a good sized hard disk. If this is your strategy, you can run all the same software that you run on your desktop computer. Other people choose a modestly equipped laptop to save cost, weight or both. When I recently joined the ranks of the laptop brigade, I followed the latter path.

My laptop computer has two 3.5 inch, 760 kilobyte floppy disk drives, but no hard disk. I was pleasantly surprised at how many program files I could pack onto a 760k floppy, but still there are limits. WordPerfect is my word processor of choice, and I could not seem to get enough of version 5 onto one of these disks to satisfy me. Then I discovered WordPerfect Executive.

WordPerfect Executive Overview

Although I bought the program primarily for word processing, WordPerfect Executive is a modest, but capable, integrated package. In addition to a word processing program based on version 4.2 of its big brother, Executive includes a spreadsheet based on PlanPerfect, an appointment scheduler, a telephone and address list manager, a note card manager and a pop-up calculator. All of this fits on one 3.5 inch disk.

These program modules are all integrated into a DOS shell. The integration is quite nice. You can move directly from one application to another without unloading the first one and retreating to the shell. Later you can move back to the first application, and you'll find your document or spreadsheet waiting just as you left it. Data is easily moved from one application to another. For instance, you can move all or part of a spreadsheet directly into a word processing document.

There's even a macro feature. All macros are global — they will try to execute in all applications. This is fine if it's appropriate, but sometimes when you're in the wrong application, the results are strange or worse. I wish macros could be limited to a particular application. There's no macro editor. If you want to change a macro, you have to redo the whole thing. Switching from one module to another is accomplished by "invisible" macros. To move from the word processor to the spreadsheet, you press <ALT-S>. If you create your own macro named <ALT-S>, it doesn't destroy the built-in macro, but your macro supersedes the built-in one.

Clearly some compromises must be made to get everything to fit on a single disk. Although the phone list manager and the note card manager are data base managers of a sort, there is no full featured data base manager. The word processor lacks super- and subscript capability, a feature that I care about. There is a spelling checker but no thesaurus. The spreadsheet can not freeze row and column headings so they do not scroll off the screen. Perhaps most surprising of all, there is no built-in communication module. You can install the communication program of your choice and run it from the shell menu.

The lack of a communication module doesn't bother me. I seem to be one of the few laptop users who doesn't have the urge to communicate on the road. I just want to write and compute and then transfer the files to my desktop when I get home. Transfer is accomplished with the Laplink program.

Despite the compromises, the various program modules are quite competent. Each one is described in some detail below.

The Word Processor

Since the word processor is based on WordPerfect version 4.2, the commands differ a bit from what I'm used to in version 5, but it's still quite similar. That similarity is one of the reasons I selected this program.

This module includes most of the standard word processing features: text enhancements like bold and underline, search and replace, ASCII file conversion, mail merge and spell checking. It lacks sub- and superscripting, a thesaurus and double wide print. It's really a nice program, but version 5 of WordPerfect spoils you. After you've driven a Porche, a VW never quite satisfies. WordPerfect 5.0 reads WordPerfect Executive word processing files directly. Documents created with WordPerfect 5.0 must be exported in 4.2 format in order to be readable to Executive.

The default left margin is permanently set at 10. You can change the margin anytime you want for the document in memory, but you can't change the default. It so happens I don't like such a wide left margin, so I have to change it at the beginning of every document. It's no big deal, but it's frustrating.

The spell checker failed to recognize the words "WordPerfect" and "laptop." Still the spell check feature is very nice. It allows you to select an alternate spelling from its list by pressing one key, and it also permits you to directly edit the word you typed. Of course you can add new words to the approved list, so omissions are easy to remedy.

The Spreadsheet

PlanPerfect is the basis for the spreadsheet module. Since I'm not a PlanPerfect user, this was not a plus. The module reads and writes Lotus 1-2-3 files, but it does so through a translation program. I can load my 1-2-3 files, but I have trouble running them. I've been a Lotus, and more recently a Quattro, in Lotus mode, user for so long that my finger automatically reaches for the slash key when I want to execute a command. In Executive, the slash key gives you — a slash! What sort of foolishness is that?

Most of the commands I want are available, but they are invoked by pressing function keys with or without the SHIFT, CTRL or ALT key. That's a perfectly sensible way for a word processor to work (since I use WordPerfect 5.0), but spreadsheets are "supposed" to use slash commands. I wish this spreadsheet had a Lotus emulation mode.

The spreadsheet handles ranges, but you have to type the beginning and ending cell addresses from the keyboard. You can't highlight ranges as you do in Lotus. Even though typing ranges is faster, highlighting is more natural, and therefore, less error prone. I really do miss that feature.

Also troubling is the inability to freeze columns and rows so they don't scroll off the screen. If I could change one thing about Executive, this would be it. You get your columns labeled in row 1 and your rows labeled in column A. Then as you move around, these labels scroll off the screen! The only solution seems to be to repeat the labels every eight columns and every 21 rows. That's a nuisance.

On the plus side, copying spreadsheet data into the word processor could hardly be easier. Simply block (highlight) the data to be copied and copy it onto the built-in clipboard. Shift directly to the word processor and copy the clipboard contents into your document.

Many functions are supported in the spreadsheet. These include the usual trigonometric and financial functions, but the statistical functions are incomplete. There's no variance or standard deviation function. Building up the formula for standard deviation is a mess.

Executive's spreadsheet includes a graph feature. Only bar graphs can be created, but within this limitation all the usual features are available. Axis scaling can be automatic or manual; automatic is the default. Labels can be entered for x and y axes, and the graph can have a title and a subtitle. Graphs can be sized to occupy the full screen or half of it. The clipboard can be used to copy graphs into the word processor. They are converted to text in the process, so they can be edited after arrival. Since they are not graphics, printing of graphs is fast. The disadvantage is that your printer may or may not support all of the characters needed to print them.

The Phone Directory

This is really a flat file manager, but the fields have already been defined. They include first name, last name, title, work and home phone numbers, work address and a large field for notes. Data from these records can be transferred easily into the word processor via the clipboard, or they can be transformed automatically into a secondary merge file.

Although there are no specific fields for home address, this information can be typed in the notes field. This is, at worst, a minor inconvenience. The ability to generate a secondary merge file is, on the other hand, a major convenience.

Phone Directory "cards" are automatically sorted by last name as you create them. I can't imagine wanting to sort them by any other field, but you can do so if you wish to.

When you first enter the Phone Directory module, you are normally in the list display. This shows one row for each Phone Directory "card." The default fields displayed here are last name, first name, company, work phone, and home phone. These defaults can be changed to suit your preference. You can display a particular record in full by moving the bar cursor to the list entry for that record and pressing <ENTER>.

I've written a macro that automatically transfers a name and address from a selected phone list "card" to the word processor. I use this to create the inside address when I'm writing letters.

The Note Cards

Like the Phone Directory, Note Cards is really a flat file data base manager. Again, the fields are preset. In this application the fields are: subject, description, date and notes. The notes field can be quite large.

When you first enter the Note Cards module you are normally in a list display similar to Phone Directory. Subject, description and date fields are displayed, one row per record (card). The full text of one card can be displayed by moving the bar cursor to that row in the list display and pressing <ENTER>.

As you create new Note Cards, they are automatically inserted into the list in alphabetical order by subject. Note Card contents can be easily moved into the word processor.

The Calculator

Here again my personal preference gets in the way. I've been hooked on Hewlett-Packard calculators and Reverse Polish Notation for a long time. The calculator in Executive isn't RPN, so I don't like it. It's perfectly good — for other people.

The calculator is a basic one: four arithmetic functions, square root, percent, and one memory register. It has a paper tape emulation, and the tape can be easily copied to the clipboard and from there to the word processor.

The Appointment Calendar

Next to the word processor, the appointment calendar is my favorite feature. I've been using the appointment calendar in PC Tools Deluxe for several months, and I like it a lot. The calendar in Executive is equally nice. If only I could transfer data between the two.

The initial screen in the appointment calendar shows a calendar page for the current month on the left side of the screen with the current date highlighted. It's easy to jump to any past or future date. A list of appointments for the current day appears on the right side of the screen. If no appointments have been entered, this window is blank. New appointments can be added, and old ones can be deleted or edited.

Any appointment can have an alarm associated with it. Default settings control how many minutes in advance of the appointment the alarm will sound and how long the alarm will sound. I wish I could make the alarms a bit louder, but alarm volume is not adjustable. When the alarm sounds, the appointment text appears in a small pop-up screen window.

Pressing SWITCH, <SHIFT-F3>, replaces the appointment list with the current day's TO DO LIST. Items can be added to this list, and old items can be deleted, edited or marked complete. Uncompleted items can be automatically carried forward or deleted as you wish. Completed items can be deleted or archived. There's also a handy cut-and-paste feature just for TO DO LIST items.

Daily schedules can be transferred to the word processor by means of the clipboard. Executive comes with a macro that will retrieve schedules a week at a time.

Predefined Macros

In addition to the "invisible" macros used to switch from module to module, Executive comes with nine predefined

Continued on Page 48

Using the Apple LaserWriter® II With a Heath/Zenith System

Pat Swayne
Hug Software Engineer

If you have both Heath/Zenith (or other PC-compatible) and Apple computer equipment in your work area as we do here where we produce *REMark* and *Heath Electronics* magazines, and your Apple equipment includes a LaserWriter® II printer, then you could be taking advantage of that printer with both types of systems. Now, if you go down to your Apple dealer and tell him you want to use your LaserWriter with a PC-compatible system, he might try to sell you a "LocalTalk" card, or an Apple-compatible network system, such as Tops. But you don't need any of those things. Your LaserWriter has an RS-232 serial port on it, and your PCs can "talk" to it through that port. The latest versions of many PC-type word processing programs can work with a PostScript laser printer such as the LaserWriter II, as can many other programs. Using a simple BASIC program that I will present later in this article, you can even print plain ASCII text files on your LaserWriter.

Making the Connection

To connect your Heath/Zenith PC to your LaserWriter printer, you will need to construct a special cable. The LaserWriter requires a "null modem" cable between it and the PC computer, but the gender of its serial cable is wrong for a standard null modem cable, such as the Heath part no. 134-1474 cable.

If your computer has a 25-pin serial port connector, make up a cable as shown below.

1 ----------------------------------	1
2 ----------------------------------	3
3 ----------------------------------	2
5 ----------------------------------	20
7 ----------------------------------	7
25-pin female conn. Computer end	25-pin male conn. Printer end

If your computer has a 9-pin serial port connector, make up a cable like this.

(connector shell) ---------------	1
2 ----------------------------------	2
3 ----------------------------------	3
4 ----------------------------------	20
5 ----------------------------------	7
9-pin female conn. Computer end	25-pin male conn. Printer end

With the power off on your LaserWriter II, set the dip switches on the back so that switch 2 is down and switch 1 (on model NT) or all other switches (on model NTX) are up. Normally, all switches are up for operation with an Apple computer, so the only switch you change is switch 2.

The LaserWriter operates at 9600 baud with XON/XOFF (DC1/DC3) handshaking, so you will have to configure your system appropriately. To do this, run CONFIGUR and select "Configure an LPT device". From the next menu, select "Map parallel output to serial output". Select the port to be mapped (usually it should be LPT1, even if you have a parallel printer connected to it). Then select "Map to COM1" or "Map to COM2" depending on which serial port you are using. Now, configure your serial port by selecting "Configure a COM device" from the CONFIGUR main menu. Select which port you are configuring, and then select "User Defined" from the next menu. Answer N to the questions about stripping parity and mapping lower case. Set the baud rate to 9600, and set 1 stop bit, no parity, 8 bit words (even though the LaserWriter manual says 7 bit words), and DC1/DC3 handshaking. Hit Return in response to the "pad characters" prompt. Then have CONFIGUR update the changes to memory, or to memory and disk. I'm going to tell you how to make the LaserWriter run faster than 9600 baud in the next paragraphs, and that is why you may only want to make your configuration changes to memory.

After you have prepared your cable, LaserWriter, and computer, connect the cable to the computer and LaserWriter, and then turn the LaserWriter on. After about half a minute, the LaserWriter will print a "test page" (unless it has been previously configured not to print one). The test page should indicate that your printer is set up for RS-232 operation at 9600 baud. To ensure that it is working, use a text editor or the non-document mode of your word processor to create a file containing these lines.

```
/Helvetica findfont 14 scalefont setfont
30 500 moveto
(The options number for the 25-pin port is )show
statusdict begin 25 sccbatch 10 string cvs show
pop showpage
```

Name the file containing these lines PSCHECK. This file is actually a small program written in the PostScript® language. The LaserWriter printer contains a PostScript interpreter, and programs that can "print" to it actually send it PostScript programs. If you can have your postscript program print to a disk file instead of directly to the printer, you can examine the PostScript programs it produces. This program, which is from the LaserWriter II manual, should cause the message "The options number for the 25-pin port is 0" to be printed. To "run" the test program, enter this line at the DOS prompt.

```
COPY PSCHECK PRN
```

If you get that message, you know that your system is working correctly, and you can begin using your programs that support PostScript. If you run into trouble, study the section on using the 25-pin port in your LaserWriter II manual.

Going Faster

The LaserWriter II is capable of running at faster baud rates than 9600, which will cause it to work faster. Unfortunately, MS-DOS does not support faster baud rates, and the LaserWriter II must be programmed to work at a faster rate (there are no switches for the faster rates). To overcome the limitation of MS-DOS, I have written a program called FASTPRN which is available for download from the HUG Bulletin Board System. If you do not have a modem, send me a disk in a self-addressed stamped disk mailer, and I will send you a copy of FASTPRN. FASTPRN supports serial printers at 19200 baud and 38400 buad.

If your printer is model NTX, you can make it operate at 19200 baud with this PostScript program.

```
serverdict begin 0 exitserver
statusdict begin 25 19200 0 setsccbatch
```

If you want to try 38400 baud, just insert that value in the program. If your printer is model NT, use this program to set a new baud rate.

```
statusdict begin 25 sccbatch exch pop
0 eq {stop} if
serverdict begin 0 exitserver
statusdict begin 25 19200 0 setsccbatch
```

The extra lines are required in the NT version because the erasable ROM in the NT that stores options can only be reprogrammed a limited number of times. This program protects against reprogramming the ROM when it is not necessary.

Use your editor or word processor (in the non-document mode) to create one of these programs, and name it SETBAUD. To set the new baud rate, just COPY the program to PRN as you did with the PSCHECK program. Then run FASTPRN by entering this line at the DOS prompt.

```
FASTPRN LPT1=COM1,19200,XON
```

Be sure to use your COM port and your baud rate in the above command line. After you run FASTPRN, turn the printer off, wait a few seconds, turn it back on, and wait for the test page, which should show the new baud rate. Copy PSCHECK to the printer again to see if it is working at the new rate. The printer will retain the new baud rate setting even while it is turned off. The setting will be erased if switch 2 is changed and then the printer is turned on again (as it would be if you wanted to use the printer with your Apple computer again).

You should only try 38400 baud if your computer is an 80286 or 80386 model, and if you are using a built-in serial port. The serial ports in older model computers and on add-in cards may not be able to work at higher than 19200 baud.

Printing Text Files

While the LaserWriter II is operating in its normal PostScript mode, it can only accept PostScript programs. Therefore, if you try to send it an ordinary text file for printing, it will not know what to do with the text, and will probably "lock up". To make it possible for you to print text files, I have written a small BASIC program that incorporates the text into PostScript programs, and then sends them to the printer. When you run the program, it will ask you for the name of a file to print, and after you enter that, it will ask for the number of lines per page. If the file is not paginated, enter the number of lines that you want to be on each page (usually 55 to 60). IF the file is paginated but does not use form feeds, enter 66 as the number of lines per page. If the file is paginated and there are form feeds at the end of each page, you will have to examine the file with a file viewer or text editor to determine exactly how many lines are on each page, and enter that number.

Below is the BASIC program for printing text files.

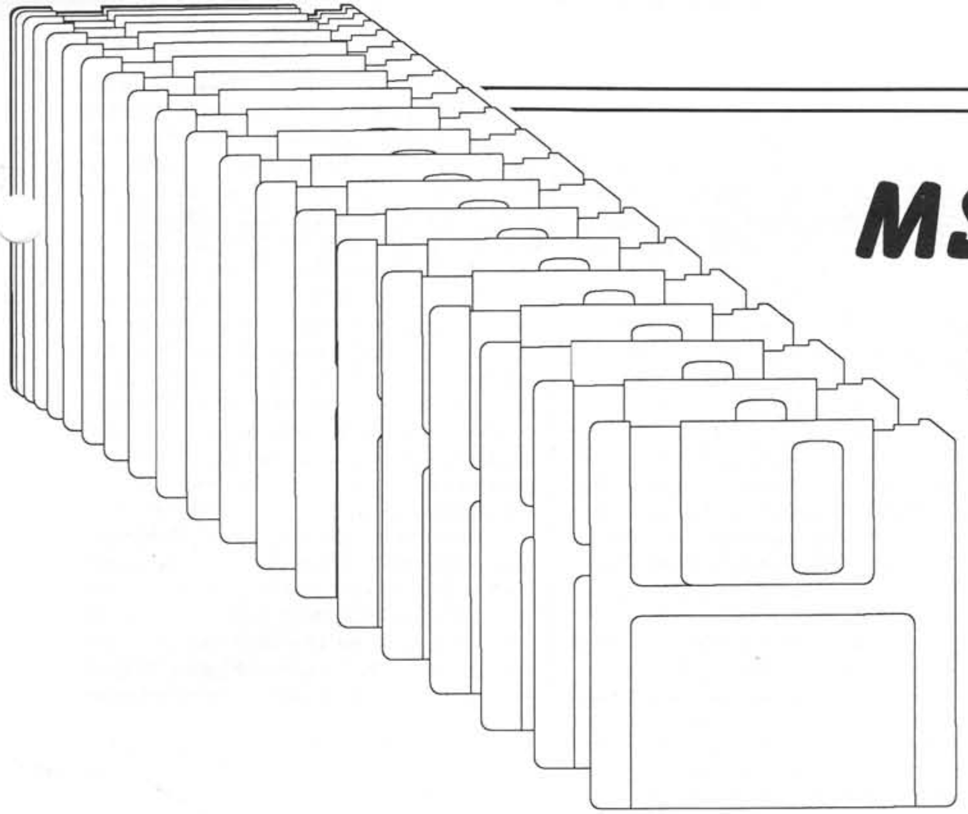
```
10 REM PROGRAM TO PRINT FILES TO POSTSCRIPT LASER
20 WIDTH LPRINT 255:LN=0:LS$="12":ON ERROR GOTO 30
30 LINE INPUT "ENTER FILE TO PRINT: ";F$
40 LINE INPUT "ENTER LINES PER PAGE: ";LP$
50 LP=VAL(LP$):IF LP>66 OR LP<1 THEN 40
60 IF LP>64 THEN LS$="11"
70 OPEN "I",1,F$
80 ON ERROR GOTO 250
90 LPRINT "/@{x y moveto show/y y ";LS$;" sub def}
   def/x 18 def/y 756 def x y moveto"
100 LPRINT "/Courier findfont 12 scalefont setfont"
110 IF EOF(1) THEN 220
120 LINE INPUT #1,L$:L=LEN(L$)
130 I=1
140 C$=MID$(L$,I,1)
150 IF C$="(" OR C$=")" OR C$="@" OR C$="\" THEN
    L$=LEFT$(L$,I-1)+"\"+C$+RIGHT$(L$,L-I):L=L+1:I=I+1
160 I=I+1:IF I <= L THEN 140
170 L$="("+L$+")@"
180 LPRINT L$
190 LN=LN+1
200 IF LN=LP THEN LPRINT "showpage":
    LPRINT "/y 756 def x y moveto":LN=0
210 GOTO 110
220 IF LN <> 0 THEN LPRINT "showpage"
230 LPRINT CHR$(4);CHR$(12)
240 STOP
250 PRINT "PRINTER ERROR!":STOP
```

In case you did not notice, this article was printed on an Apple LaserWriter II. I used WordStar version 6 (believe it or not) to prepare the article. The current versions of most of the Big Name word processing programs have at least some desktop publishing capability built in.

In Case of Trouble

When the LaserWriter II is connected to an Apple computer system, it works interactively with it, so that if anything goes wrong, you will get some kind of message on the screen. When it is connected via the serial port, however, it cannot "talk back" to the computer. Therefore, if a problem occurs, it will probably just "lock up". This can happen if the printer receives an impropper PostScript command, an unfinished PostScript statement, or if it just runs out of paper. MS-DOS or the FASTPRN program may time out in this case and give you some kind of message. The solution is to turn the printer off, fix the problem (if it is not bad PostScript information), and turn it on again. You probably would not have lock ups if you were using a LocalTalk card or an Apple-compatible network, but then think of the money you have saved yourself or your company by using this method.

Some applications, especially those that run under Windows, will not work with FASTPRN. Usually, these applications can be made to print to a file, which you can copy to PRN after you exit from the program (or from Windows).



MS-DOS FILE RECOVERY

Gale R. Horst
Zenith Data Systems

Increase Your Confidence in File Recovery By Understanding the MS-DOS File System

I sometimes reflect on my first major disk disaster when the MS-DOS "DIR" command displayed all sorts of indecipherable garbage all over the screen. I had little understanding of what could cause MS-DOS to do this. The problem appeared to be severe so I reformatted the drive and rebuilt it from my backups. Much later I realized that equipped with an understanding of the MS-DOS file system I could have recovered most of my data using a "disk doctor" utility such as *General Disk Utilities* (which is provided along with Zenith Data Systems' MS-DOS 3.30+).

Successful file recovery is not the only benefit of understanding the internal structure of the MS-DOS file system. It will also enable you to discern a swift, effective, and confident course of action when MS-DOS reports an error.

Before we begin our discussion of MS-DOS' method of managing disk space, let's look at what is necessary for any file system to organize the storage space on a disk:

1. First we need an index. Obviously, we must be able to see what files are on a disk.
2. Considering the large number of files that will be contained on the disk we should have a way to divide the files into related groups of files.
3. We need to know where the data is located (physically) for each file on the disk.
4. We must be able to easily delete, add, or change the size of files at any time with minimal effort.
5. The "holes" left in the disk when files are deleted must be used effectively when new files are added.
6. We also need a method to keep track of available (unused) disk space for allocation to new files.
7. We need a way to mark areas on the disk that are unusable due to defects.

Now let's see how MS-DOS meets these disk organization requirements.

Disk Organization

Directories

MS-DOS meets the first two requirements by allowing many directories on a disk. Each MS-DOS directory is divided into 32-byte pieces. Each 32-byte piece can represent one file. The file information in each 32-byte directory entry is defined as follows:

Bytes	**Usage**
1-8	File name (as shown by the "DIR" command)
9-11	File name extension (as shown by "DIR")
12	File attribute byte
13-22	Unused (zeros)
23-24	Time the file was last written
25-26	Date the file was last written
27-28	First cluster used by the file
29-32	File size (in bytes)

Let's use a file called "MYFILE.TST" as an example. From the MS-DOS "DIR" command, the following information may be reported:

```
MYFILE   TST      9129    5-22-89    3:15p
```

If we look at the "raw data" in a single directory entry it will look like this (displayed in hexadecimal representation):

```
4D 59 46 49 4C 45 20 20 54 53 54 20 00
            00 00 00    MYFILE TST ....
00 00 00 00 00 00 FB 79 B6 12 00 00 A9
            23 00 00    ......###########
```

You do not need to be familiar with hexadecimal (base 16) numbering to understand this article. However, the "raw" directory data displayed above will give you an idea of what to expect when you have to look for directory data to recover information from a crashed disk.

A special area on the disk is reserved for the root directory. Since the number of sectors reserved for the root directory is predefined, the maximum number of files in the root directory is limited. (On most hard disks the limit is 512.) However, subdirectories are of variable size and have no limit on the number of files they may contain. That is because a subdirectory is a special type of file. The file attribute byte (12th byte in the directory entry) has a bit set (bit 10h) indicating that the file is a subdirectory. Also note that the file size for a subdirectory is always 0. The file name, time, date, and first cluster look just like any other file.

The data in subdirectory clusters looks just like the data in the root directory with the addition of two special files at the beginning of each directory. These first two special files are subdirectory entries (file attribute byte = 10H) with file names "." and "..". The first one is a self pointer and the second is a directory entry that tells MS-DOS where to find the "parent" of this subdirectory. This is why when you enter the "DIR" command for a subdirectory the first two files shown always look something like:

```
.              <DIR>          5-17-89   2:20p
..             <DIR>          5-17-89   2:20p
```

MS-DOS defines the smallest amount of storage space that may be assigned to a file as a *cluster*. The size of a cluster varies depending on the size of the disk and the version of MS-DOS. However, at this point of our discussion, the size of a cluster is not important.

When you create a subdirectory, a single cluster is allocated to the subdirectory file. Adding more files to the subdirectory will eventually fill the cluster with directory data. At that point, MS-DOS will allocate another cluster to the subdirectory file. That allows subdirectories to grow as large as necessary.

Files

We need several more lists to implement a file system. We need a variable-length list of clusters for each file so we can find each file's data. We must have a list of available ("empty") clusters. And finally, we need a list of clusters that are unusable due to defects in the disk itself.

Although that sounds like a lot of lists, MS-DOS manages with just one list called the *FAT* (File Allocation Table). The FAT is a single list containing one entry for each cluster on the disk. The contents of each entry in the FAT informs us about the cluster it represents:

> If the value of the FAT entry is zero, the cluster associated with this entry is not allocated to any file. MS-DOS uses these zero entries to keep track of unused space on the disk.
>
> If the value is between 2 and x+1 (x being the total number of clusters on the disk), then that cluster belongs to a file. This value is also a pointer to the next cluster in the file. The next cluster can be any other cluster on the disk.
>
> The value FFFFH, marks the last cluster of a file. Note that FFFFH is the hexadecimal notation for the largest number that can be contained in this storage location. If this number is displayed in decimal, it will appear as either a -1 or 65,535. Also note that some MS-DOS documentation manuals state that any number greater than or equal to FFF8H (65,528) marks the last cluster of a file. In reality, I've never seen anything used other than FFFFH.
>
> If the value is FFF7H (65,527 in decimal), the cluster associated with this FAT entry resides on a damaged and unusable area of the disk. MS-DOS will not use this cluster.

A closer look at the FAT structure will show us that all of the necessary lists we mentioned earlier (except for the file index) are handled by the FAT structure. Keep in mind that each entry in the FAT represents one cluster and it is a list pointer telling us where the next cluster in the file is located (assuming the FAT value is not 0, FFF7H, or FFFFH).

Example

Let's use our example file again. "MYFILE.TST" has a directory entry that contains the first cluster number allocated to the file. When we want to access MYFILE.TST, MS-DOS finds it in the directory and determines that the first cluster used by our file is cluster number 3. Now by looking at the FAT we can determine what clusters belong to MYFILE.TST:

```
              /-------------------------------------------    \
FAT data -->  |  0 |  4 |  5 |  8 |  0 |  0 |  9 |FFFF|  0 |... |
              \-------------------------------------------    /
FAT entry #      2    3    4    5    6    7    8    9    A   ...
                      ^
Directory Entry -----^
```

We see that FAT entry #3 contains the value 4. The 4 means that cluster 3 is used by MYFILE.TST and the next cluster of the file is cluster 4. Similarly, we can determine that the third cluster of MYFILE. TST is cluster 5. So far, the clusters used by MYFILE.TST are grouped right next to each other. However, now we see that the next cluster after cluster 5 is cluster 8 since the value in the FAT representing cluster 5 is 8. Now we skip to cluster 8 and see that cluster 9 comes next. The FAT entry representing Cluster 9 contains "FFFF". The entry FFFF indicates that cluster 9 is the last cluster of the file.

You may be wondering why the FAT has zeros in entries 6 and 7. This type of configuration implies that there was another file using clusters 6 and 7 at the time MYFILE.TST was created or enlarged. This forced MS-DOS to find other clusters to use.

We have determined that MYFILE. TST is using clusters 3, 4, 5, 8, and 9. The first time many people attempt to understand the FAT they mistakenly believe that the file is using clusters 4, 5, 8, and 9 because these are the numbers that appear in the FAT entries. Remember that the number in the FAT represents the next cluster number in the file. Don't forget to include the current cluster whose FAT entry points to the next one.

Now your understanding of the MS-DOS file system should begin to take shape. Starting with the next "0" entry found in the FAT, MS-DOS allocates the clusters for a file in sequential order. If there happens to be enough "0" entries in the FAT lined up right next to each other to hold the file, then these are the ones that are used. Otherwise, MS-DOS will scan the FAT until another "0" entry is found. The clusters corresponding to these FAT entries are written with the file's data and the FAT is made into a "linked list" as we described above.

To "delete" a file, all MS-DOS must do is remove the reference to the file in the directory and change all the file's FAT entries back to zeros. The data in the clusters does not have to be read to "delete" a file nor does it have to change. Just marking the clusters in our list as unused effectively deletes the file even though the data is still intact.

One small, but important, reserved area of the disk remains to be mentioned. Each logical drive has its own boot sector that tells the operating system pertinent information about the disk format. Boot sector data includes (among other things):

Size of the FAT
Size of the root directory
How many sectors the disk contains
The cluster size (number of sectors per cluster)

Now that we know about the MS-DOS file structures, we can draw a picture of the disk space. The disk space is arranged as follows:

```
/-----------------------------------------\
| Boot Sector (logical sector #Ø)         |
|-----------------------------------------|
| File Allocation Tables, two copies beginning|
| with logical sector #1.  The number of  |
| sectors needed for the FAT depends on the |
| size of the disk.                       |
|-----------------------------------------|
| Root Directory, what sectors are used by |
| the root directory depends upon the size of |
| the FAT and the format of the disk.     |
|                                         |
|-----------------------------------------|
| Cluster #2  (Cluster numbers start with 2) |
|-----------------------------------------|
| Cluster #3                              |
|-----------------------------------------|
| Cluster #4                              |
|-----------------------------------------|
        .
        .
        .
|-----------------------------------------|
| Cluster #x+1  (x = # of clusters)       |
\-----------------------------------------/
```

File Recovery

As was mentioned earlier, when MS-DOS "deletes" a file, it removes it from the directory. In reality, the directory entry is NOT removed. MS-DOS replaces the first character of the file name with a special character (E5H) indicating that this directory entry is available.

File recovery utilities can ask the user to enter the first character of the file name. The first cluster number of the file and the file size (unless the file was a subdirectory), is available in the "erased" directory entry. We know that after a file is "deleted" all of the information that was in the file is still on the disk somewhere. Here's the catch. After the first cluster (obtained from the "erased" directory entry), it is impossible to ascertain what other clusters belonged to the file since our file's FAT entries have been replaced by zeros. We know that when MS-DOS created the file it used clusters that were all right next to each other if they were available at the time. However, there is no guarantee whether that was the case. Most file recovery utilities (with a few exceptions) must do some guess work. Let's go back to our example of MYFILE.TST which uses clusters 3, 4, 5, 8, and 9. If we delete our file, the FAT will look like:

```
             /-----------------------------------------\
FAT -------> | Ø | Ø | Ø | Ø | Ø | Ø | Ø | Ø |... |
             \-----------------------------------------/
FAT Entry #    2   3   4   5   6   7   8   A   ...
```

We know that the directory entry tells the operating system that the first cluster was number 3. By dividing the file size (obtained from the "erased" directory entry) by the number of bytes in each cluster we can determine the number of clusters needed (five clusters in our example).

Almost all the file recovery utilities I have tested will assume that the next five erased clusters contain the file's data. In our example, the file called MYFILE.TST would be incorrectly recovered using clusters 3, 4, 5, 6, and 7. If we loaded this file back into our word processor, the first part of the file would be fine since clusters 3, 4, and 5 were part of our original file, but clusters 6 and 7 do not belong to us. The last part of our file (clusters 8 and 9) will be missing. These incorrect clusters (6 and 7) could contain any type of data. If the data is binary data, it could cause your word processor to behave strangely and/or display all sorts of strange characters on your screen. On the other hand, these incorrect clusters could contain text from another text file that was deleted earlier. In either case, something went wrong even though the file recovery utility doesn't admit it.

As long as there were enough erased clusters on the disk to create a file of the same size as the erased file, most of the popular "disk doctor" utilities will report something like 'file was successfully recovered'. As we have demonstrated, **don't believe it until you have examined the file.**

Note that *General Disk Utilities®* (GDU.EXE on the Zenith Data Systems MS-DOS 3.30+ diskette) uses some additional logic to locate the correct clusters when reconstructing a file. GDU analyzes the data in each erased cluster to determine if an erased cluster is likely part of the erased file. Since we can be certain of the first cluster, the first cluster is analyzed. Then, each sequentially erased cluster encountered in the FAT is also analyzed to see if the data corresponds. The advanced Zenith Data Systems algorithm used by GDU will correctly recover files that cannot be recovered successfully by other utilities. However, even this more advanced recovery algorithm can still fail depending on the data that just happens to be in the other erased clusters near the same location on the disk.

In addition, GDU provides a TSR (Terminate-and-Stay-Resident) utility that will capture a file or information about a file just before it is deleted. The saved information may be used if it is necessary to restore the deleted file. In the case of an "oops", when you realize your mistake immediately after you deleted an important file, installation of this TSR will ensure 100% correct file recovery regardless of how fragmented the file was.

So what do we do now if our file was recovered incorrectly? If you erase the recovered file again, you will be back to where you started. If you are recovering a text file and the recovery attempt found some of the correct clusters, you may want to make a note of which clusters were correct. GDU will let you examine a file one cluster at a time. Get a pencil and write down the cluster numbers that are correct as you examine each cluster. Later, you can use the *Rebuild File* feature of GDU to manually piece your file together. (We will discuss manual file recovery later in this article.)

Let's examine some other cases. If a disk has been written to after a file was deleted, the chances of successful file recovery are greatly reduced. When another file is written to the disk, one or both of the following could happen.

1. If the file is written to the same directory, MS-DOS may re-use the directory entry that was once occupied by the deleted file. Recovery utilities no longer have any knowledge of the deleted file since it has disappeared from the directory. Recovery of the file may still be possible by searching the erased clusters of the disk for data known to be a part of the deleted file.
2. A file written to the disk may now be using all or some of the clusters that were previously part of the deleted file. In this case, a partial recovery may be possible if at least some of the original file's clusters remain unused.

Manual File Recovery

GDU includes several disk search options to search for data that was part of your deleted file. If the file was from your word processor, you may want to search for words or phrases that were a part of your document. If you are working on a large hard disk, the searches may take several minutes, but if your file was an important one, it's worth the wait. When the word or phrase is found, you are allowed to examine the contents of the cluster to decide if it is part of your file. Keep in mind that the word or phrase could possibly span clusters. The first part of the phrase could be at the end of one cluster and the last part at the beginning of another. To help you with this problem, the data searches in GDU will stop if the first part of your search phrase is found at the end of a cluster. Try searching for several phrases before giving up. In most cases, you will want to start by searching for data that was in the first part of your file. If the search fails, then look for other data that was contained later in the file. Once you have found one cluster, the rest of the clusters are usually nearby unless your disk was severely fragmented.

Here you can see the advantage of periodically running a disk compaction program that will make sure that all your files are using sequential clusters. Compaction not only reduces file access time, but also makes file recovery much easier. A disk compaction program (COMPACT. EXE) is also provided with ZDS' MS-DOS 3.30+.

As you locate the correct data clusters you can add them to your file. It is very likely that the file's clusters should be put together in ascending order by cluster number (in the same order as you located them). If the order is not ascending, you will have been able to make that determination as you located and examined the clusters. In this case, you must write down the cluster numbers as you find and examine them. After you have found all of your data, then you can use GDU's *Rebuild File* feature to build your file using the clusters you have found.

Example

Assume I have a document in a text file on a floppy disk that allocates one sector per cluster and the file occupies 20 clusters. I delete my file and then write a small file that needs only one cluster. Let's also assume that the first cluster of my old file was reallocated to the new file. Now, I realize I didn't want to delete the file so I attempted an undeletion by typing GDU <filename>. GDU reports "file not found". This means that either the name of the deleted file is not in the directory or the first cluster, which was used by the deleted file, has been reallocated to another file (such as in this example). Note that in this case, if you select *Display/Edit Directory* from the main menu of GDU, your file may appear as an erased file. (GDU indicates erased files by displaying a question mark in place of the first character of the file name.) Even though your erased file name appears here, it may not appear in the list of "recoverable" files if you select *Undelete File* from the main menu. This indicates that the first cluster the file once used has been reallocated to another file.

In our example, only a small amount of data has been lost since 19 of the file's 20 clusters are intact. Using a GDU's *Rebuild File* feature, you will be able to search the erased areas of the disk and find 95% of the file's data (19 out of 20 clusters).

I must mention another peculiarity of the MS-DOS file system that relates to file recovery. Let's assume you are working on a hard disk that allocates four sectors per cluster. Each cluster will contain 2,048 bytes of data. Now let's assume you write a file of 8,193 bytes. Dividing the file size by the number of bytes in each cluster shows us that your file will occupy five clusters. (8,193/4 = 4 with a remainder of 1.) Four clusters will be completely full, but the last cluster will contain only one byte of data. When MS-DOS writes this file, the last cluster will have only the first byte changed. Whatever data was in that last cluster will remain except for the first byte. The next time MS-DOS reads the file, it will not use the leftover data since MS-DOS can determine how much data is valid by dividing the file size by the number of bytes per cluster.

The extra data beyond your file's data may be text or it may be unrecognizable (part of an executable file or binary coded data). Don't let the extra data alarm you. If it's a word processing file you may have to chop off the last part of the file when you edit it.

Recovering A Subdirectory

In a severe disk disaster it is possible that your directory could become corrupted. A more likely scenario is that you accidentally deleted all the files in a subdirectory and removed the subdirectory before realizing that you still need the files. In both cases, you must recover a subdirectory.

Most disk utilities will let you "undelete" a subdirectory. However, recall that, unlike normal files, the file size for a subdirectory is always listed in the parent directory as 0. This means that the recovery utility cannot accurately determining how many clusters the subdirectory required. If your subdirectory was larger than one cluster, you may have to locate some of the clusters manually.

Unlike normal files, the clusters used by a subdirectory will probably NOT be adjacent to each other since you must add files to your disk to necessitate a larger subdirectory. Adding more files will likely use up any unused clusters that may have been near the first subdirectory cluster. However, some disk compaction programs, such as Zenith Data Systems' COMPACT.EXE, will move data around on a disk to move subdirectory clusters together. This makes recovery much easier.

GDU provides the ability to locate lost subdirectory clusters and rebuild a subdirectory. This means that even if you have deleted all the files in a subdirectory and removed the directory (MS-DOS RMDIR command), you can still recover the directory and then undelete files in that directory. An erased directory still appears in the parent directory as an erased file (with the attribute indicating that it was a directory). If the directory name no longer appears in the parent directory, you may locate a lost subdirectory by searching erased clusters for the name of a file that you know was in the subdirectory. File names always use 11 character positions in the directory: an eight-character file name plus a three-character extension. The "." between the file name and extension is not recorded in the directory. Therefore, to search for a lost directory containing the file "MYFILE.TST" you would search for "MYFILE TST". Recall that if the file was deleted from the directory, the first character of the file name will be missing. Therefore, you should search for the ten characters "YFILE TST".

As I mentioned before, the first two files in a subdirectory are special entries with the file names "." and "..". If the cluster you found does not have these entries at the beginning, then you have probably found one of the subsequent clusters of your directory. Write down this cluster number and search for some other files in your directory. When you have found all the pieces of the directory, then you can rebuild the directory in the same way as you rebuild a file. However, this file will have to be flagged as a subdirectory in its parent directory. If you selected an erased directory entry in GDU's *Rebuild File* feature, then GDU will already know that you are rebuilding a directory. Otherwise, GDU will prompt you to select either a file or a directory before you get started looking for the directory clusters. To familiarize yourself with what a directory cluster looks like, Select *List/Edit in Hex and ASCII* from the main menu of GDU. Then select a subdirectory to be displayed. You will see subdirectory data as MS-DOS sees it, which is the way you will have to identify it during recovery.

After you recover a subdirectory, all the deleted files that were in the subdirectory will appear again. Now you can recover these deleted files as we discussed earlier.

If the problem you are correcting is that the root directory was corrupted with garbage data (and the FAT is still OK), MS-DOS may not have any knowledge of your subdirectories since they appear as files in their parent directory. All directories that branch off of a lost subdirectory will also be unreachable. Fortunately, the reverse is also true. If the only thing that was corrupted was a directory that also contained subdirectories, then recov-

Continued on Page 48

Robert Maskasky
3220 Carlton Court
Green Cove Springs, FL 32043

Using an NEC Multisync Monitor on a VGA Equipped Z-150

In 1986, the computer monitor market was turned upside down with the introduction of the NEC JC-1401P3A monitor, better known as the "Multisync" monitor. This was the first monitor capable of switching between the IBM CGA standard, EGA standard and a little used IBM PGA (Professional Graphics Adapter) video card. The Multisync monitor was also compatible with several other video standards used with non-IBM computer systems.

IBM designed the EGA video card to operate at 21.5 kHz horizontal frequency. Since IBM also wanted compatibility with the older CGA video, IBM also designed the EGA to operate at 15.735 kHz horizontal frequency. Until that time most computer monitors were designed to operate at a single horizontal frequency. The EGA monitor now had to sense which mode the EGA video card was operating in and automatically switch from CGA (15.735 kHz) to EGA (21.5 kHz). IBM designed a complicated system in which the polarity of the horizontal sync signal was changed from 0 volts with positive 5 volt pulses in CGA mode to 5 volts with 0 volt pulses in EGA mode. IBM designed the EGA monitor with a detector to sense if the average level of the horizontal sync was 4.8 volts (indicating EGA mode) or 0.2 volts (indicating CGA mode). IBM then used automatic switches to change the internal circuitry of the monitor as necessary.

The multisync monitor did not use the complicated sync polarity switching system designed by IBM. Instead, NEC used a bank of timers to sense the horizontal frequency and automatically adjust the horizontal circuitry as necessary. This enabled the NEC multisync to automatically adjust to any horizontal frequency between 15.5 kHz and 35 kHz. The Multisync was also able to operate with either polarity sync pulse (0 volts with 5 volt pulses or 5 volts with 0 volt pulses). Thus, the Multisync was compatible with the CGA and the EGA video. Since the NEC Multisync was capable of operating at horizontal frequencies of up to 35 kHz, this gave birth to a whole new generation of video cards known as the enhanced EGA cards. IBM never manufactured an enhanced EGA video card, but almost every other EGA vendor did. These enhanced EGA cards used horizontal frequencies of between 21.5 kHz and 35 kHz to produce even finer detail graphics images.

The IBM Professional Graphics Adapter used a horizontal frequency of 31 kHz which was within the range of the Multisync. However, the PGA used analog video.

TTL Video Versus Analog Video

Computer monitors typically used TTL logic levels (0 volts and 5 volts) to control the video signal and change the screen from dark to bright. Some computers used a video modulator to change this TTL signal into a signal which was compatible with standard television RF signals. The color of the screen was related to the type of phosphor used in the Cathode Ray Tube in the monitor. Color monitors basically consist of three monitors in one. Each of these three had a different color (Red, Green, and Blue). Various combinations of these three colors produced a total of eight colors. The CGA video standard used these eight colors and also had an additional intensity video signal which allowed the display to be either bright or dim. Thus, the CGA video card had a total of 16 colors (eight bright and eight dim colors). The EGA video replaced the single intensity signal with three video intensity signals which were used to control the brightness of each of the three primary colors individually. The combination of these colors gave the EGA the capability of producing 64 colors. In order to add more color capability, the PGA video card used three analog video lines. With analog video the brightness of the color is related to the voltage level of the signal (between 0 volts and 1 volt). The intensity or brightness signals used with the CGA and the EGA video are not necessary with analog video. Using analog video, the PGA video card was capable of almost an infinite variety of colors.

NEC anticipated that PGA would eventually replace the EGA video cards, so NEC designed the Multisync to be compatible with the PGA video standard, as well as the EGA and CGA. A switch on the back of the monitor changed the monitor from TTL input to analog input. This was the first monitor to be compatible with all three IBM video standards, as well as numerous other standards used by other computer manufacturers. NEC also designed the Multisync monitor to be compatible with composite sync (both vertical sync and horizontal sync on the same wire) which was used on the PGA video card or separate horizontal and vertical sync signals (used in the CGA and VGA standards). Until this time, almost all the computer monitors used 60 Hz for the vertical frequency.

Shortly after the release of the NEC Multisync, IBM announced the Video Graphics Array (VGA) video card. The VGA video card used analog video similar to the PGA video card and used three horizontal frequencies (15.735 kHz, 21.5 kHz and 31.5 kHz). This was well within the capability of the Multisync. However, the VGA video card hit NEC in its one weakness, the vertical frequency. The NEC Multisync was designed to operate at a 60 Hz vertical frequency only. The VGA card switched from 60 Hz to 70 Hz vertical frequencies depending on the mode the card was operating in. IBM used separate vertical and horizontal sync signals. IBM used the polarity of the vertical sync signal to switch the monitor circuits from 60 Hz to 70 Hz and the polarity of the horizontal sync signal to switch the monitor's horizontal frequency. The NEC Multisync was capable of operating at 70 Hz vertical rate with adjustments to the vertical hold and vertical size adjustments. A compromise adjustment can be found for the vertical hold control which allows the monitor to operate at both 60 Hz and 70 Hz. The vertical size is reduced by about 20% when the monitor operates in the 60 Hz mode. Thus, the vertical size requires adjustment when changing modes. NEC quickly modified the Multisync monitor to make it compatible with the VGA card. The new Multisync monitor was called the Multisync II. The Multisync II was later replaced by the Multisync 2A which eliminated the TTL compatibility to reduce the cost.

When the original Multisync monitor was released, I was so impressed with the clarity of the picture, I sold my Zenith Data Systems color monitor and purchased an NEC Multisync to use on my CGA equipped Z-151. I anticipated switching to EGA in the future and the NEC seemed flexible enough to be compatible with any future video standard. Eventually, I upgraded my Z-151 to EGA and was perfectly happy with my Multisync. Recently, I upgraded to a VGA video card and I did not want to purchase a new monitor.

First, I had to locate a cable. The VGA used a DB-15HD connector and the DB-09 connector was used on the CGA and EGA video cards. My local computer store was of no help. I contacted NEC and they gave me a part number for an adapter. The adapter costs $69.00. I don't know what this adapter consists of, but $69.00 seems high considering I just needed a cable. I decided I had to make my own cable. I searched every computer store in town for the DB-15HD connector. I finally found the connector in the Radio Shack catalog. The salesman at Radio Shack denied the connector existed until I showed him the catalog, he had the connector in stock for less than $5.00.

Making a VGA Cable

Having the connector, I had to determine the wiring. I disassembled the NEC Multisync EGA/CGA video cable and replaced one of the DB-09 connectors with the DB-15HD connector. Since the DB-15HD is the same physical size as the DB-09, I used the NEC supplied backshell. The NEC EGA/CGA video cable consists of two DB-09 connectors and an RF choke. Since the wiring is pin for pin from one DB-09 connector to the other, it doesn't matter which one you remove. The RF choke is a ferrite ring which is potted in a rubber case around the cable. If you desire, you can cut the rubber away from the ferrite ring and the ring will slide anywhere along the cable you desire.

The DB-15HD connector is wired as follows:

Db-09 Pin	DB-15HD Pin	Signal
1	1	Red Video
2	2	Green Video
3	3	Blue Video
–	4	Not Connected (Monitor ID #2)
9	5,10	Ground (pins 5 and 10 connected together
6	6	Red Ground
7	7	Green Ground
8	8	Blue Ground
–	9	Do Not Install Pin (used as a key)
9	5,10	Sync Ground
9	11	Monitor ID #0
–	12	Not Connected (Monitor ID #1)
4	13	Horizontal Sync
5	14	Vertical Sync
–	15	Not Used

Some VGA cards use the monitor ID pins to determine the type of monitor connected. Depending on the make of your VGA card, the monitor ID pins may have to be connected to ground in various combinations. The EGA monitors I tested had monitor ID pin #0 connected to ground. Most of the non-IBM VGA cards do not use the monitor ID pins. If you decide to manufacture your own cable, the Red, Green and Blue video lines must be shielded cable with the shield tied to ground.

Selection of VGA Cards

Almost any VGA card will work in the Z-151 computer. I tested the Paradise VGA + 16 and the Video-7 VEGA VGA-16. Both cards worked well. Even though the Z-151 uses an 8-bit buss, either card could be used in the 8-bit buss system. Both cards have a special mode the IBM VGA card does not have. By setting a jumper (or switch) on the card, the card will scan all modes at 60 Hz vertical frequency. This is called a Multi-frequency monitor mode or Multi-synchronous monitor mode. This mode was obviously intended to simplify the problems of vertical hold associated with changing frequencies and works great with the NEC Multisync. The monitor still changes vertical height in some graphics modes (which I'll address later).

The Video-7 VEGA VGA card had a BIOS conflict with a Sysgen floppy disk controller card I was using. This was easily solved by changing the BIOS address jumpers on the Sysgen card. Both cards had a conflict with the extra memory I had installed in the computer. I was using one of the memory address PAL chips allowing me to use 256 kB chips on the memory board to obtain 704 kB of main memory. This was a common modification for the Z-151 and was available from several aftermarket manufacturers. These PALs were available in several versions allowing 640 kB, 704 kB, 640 kB plus a 512 kB RAM Disk or 704 kB plus a 512 kB RAM Disk. Some of the VGA graphics modes expect VGA memory to be in the locations occupied by my main memory between 640 kB and 704 kB. This problem was solved by setting the CPU switches to 640 kB as per Table 2.4 of the Z-100 PC Operations Manual (Z-151 Owners Manual) and replacing U455 with a PAL which allowed only 640 kB of main memory. Since I did not wish to purchase another PAL chip, I found that by lifting pin 4 of U455 (allowing pin 4 to sit outside the socket, not connected), the PAL would not address the memory above 640 kB and would work fine with the VGA card. If your computer does not have a replacement PAL, or the PAL allows only 640 kB of main memory, you should not have this problem.

NEC Multisync Switch Settings

On the rear of the NEC Multisync monitor there are several switches. For EGA or CGA operation the TTL/Analog switch MUST be set to TTL. For VGA or PGA operation this switch MUST be set to Analog.

CAUTION: SETTING THIS SWITCH TO THE WRONG MODE WILL CAUSE DAMAGE TO THE MONITOR.

The Manual ON/OFF switch should be set to OFF. On the top of the monitor, the TEXT switch should be set to OFF (toward the rear of the monitor) and the H Width switch should be set to ON (toward the front of the monitor).

Vertical Height Modification

Since the NEC Multisync was manufactured to operate at 60 Hz vertical frequency only, operating the monitor at other vertical frequencies requires re-adjustment of the vertical size. Some of the VGA cards can be set to scan all video modes at a 60 Hz vertical frequency. I recommend using this mode with a Multisync monitor. These VGA cards do not compensate for the vertical height difference, thus the Multisync will require a vertical height re-adjustment when changing from text to some graphics modes and vice versa.

In the IBM PGA card, there were several different modes which required changing vertical height. The NEC Multisync monitor was designed to compensate automatically for this change in vertical height. The monitor used a "MODE" control signal from the PGA card. On the Multisync monitor, the MODE control line is used for vertical sync in a separate sync system. The Multisync monitor automatically senses if a vertical sync signal is applied to the Mode control/V Sync line and automatically switches the function of the line as required. In this VGA application, we use this line as Vertical Sync. The Multisync detects the V Sync and disconnects the Mode control hardware. It is possible to modify the Multisync monitor to automatically compensate for the vertical height using much of the same circuitry designed into the monitor for the PGA Mode Control. For those people who built their own H-151s from Heathkits, or for those people who can handle a soldering iron with some precision, this modification should be fairly simple and requires less than an hour.

First you'll need a padded work surface to prevent scratching the monitor, some standard tools and a small soldering iron. Parts required is a 2N2222 NPN Transistor, a 1000 ohm 1/8-watt carbon resistor, a 1 microfarad 16-volt nonpolarized ceramic capacitor and a short piece of 22 ga single-strand wire (wire wrapping wire). Perform the modification as follows:

1. Remove the cables from the Multisync monitor.
2. Remove the swivel base from the monitor. To do this, turn the monitor upside down. Move the swivel base all the way back (toward the rear of the monitor) and rotate the base 180 degrees (so that the words "NEC Multisync" are in the rear. Now the swivel should lift off the base of the monitor. Do not force it or remove the phillips screw on the bottom of the swivel. Once the swivel has been removed, remove the two screws that hold the monitor base to the monitor and remove the monitor base.
3. Remove the four phillips screws that hold the back on the monitor (one screw in each corner) and remove the back from the monitor.
4. Now you will see the monitor is constructed from a steel cage. Remove the foil cardboard cover from the rear of the monitor with four phillips screws.
5. Remove the top cover from the monitor. This cover is attached by four screws (two into each side panel).
6. Remove the Interface circuit board from the monitor. The Interface circuit board is the circuit board which contains the video connector and the TTL/Analog switch. The Interface circuit board is held by two phillips screws attaching it to the back panel and one phillips screw which is accessible through the top of the monitor. Once the Interface circuit board is loose, using a pair of (very long) long nose pliers remove each of four connectors attaching the circuit board. Some of these connectors are difficult to remove, none of them are retained by clips and thus can be removed by gentle prying. Be careful not to slip with the pliers and damage anything else inside the monitor. After all four connectors are removed, the circuit board and board carrier can be removed from the rear of the monitor.
7. Remove five screws attaching the Interface circuit board to the circuit board carrier and remove the Interface circuit board.
8. Disable the Automatic mode sense override circuitry by removing resistor R-880 (5.6k ohms).
9. Redesign the Automatic mode sense circuitry as follows:
   a. Locate resistor R-881; desolder and remove the end of resistor R-881 which is farthest away from transistor TR-829.
   b. Solder a short piece of 22 ga single-strand wire from the open end of resistor R-881 to +5 volts. +5 volts is available from either side of R-886 (1 ohm) located near the corner of the circuit board.
   c. Attach a 2N2222 transistor across TR-829 as follows: The collector of the 2N2222 is connected to the base of TR-829. The emitter of the 2N2222 is connected to the emitter of TR-829. The base of the 2N2222 is bent 180 degrees and rises above the top of both the 2N2222 and TR-829. **Note:** TR-829 is an E-C-B pattern. Most 2N2222 transistors are E-B-C pattern, this means the outside legs of the 2N2222 connect to the outside legs of TR-829. The center leg of the 2N2222 is the base which at present is sticking up in the air.
   d. Connect one end of a 1000 ohm, 1/8-watt resistor to the open pad where resistor R-881 was connected. Connect the other end of the 1000 ohm resistor to the base of the 2N2222.
   e. Connect a 1 microfarad capacitor between the emitter and collector of the 2N2222 (also across the emitter and base of TR-829).

For those who are technically inclined, we have just installed a 2N2222 as an inverter in series with the base of TR-829. This is connected to the vertical sync line. The 1 microfarad capacitor is a filter to detect the average level of the vertical sync signal (inverted). The output of TR-829 (Collector) feeds to a solid state switch which connects potentiometer R-451 across the vertical size potentiometer when the average level of the vertical sync signal is high.

10. Re-install the Interface circuit board in the reverse order of steps 7, 6, 5, 4 and 3 above.

Test the monitor as follows: **Note:** Ensure the monitor TTL/Analog switch is in Analog mode.

1. Connect the monitor to the computer with a VGA video card. Turn on the computer and boot the computer to the DOS prompt. Adjust the controls on the top of the monitor as desired. Fill the screen with text (ex: execute a "DIR"). The screen text should look normal.
2. On the back of the monitor change the Manual switch to ON. There should be no change in the vertical height. The screen text should look normal.
3. Call up a program which uses 400 line resolution (generally VGA graphics programs or a VGA test program or 132 column mode in certain word processors or Lotus 1-2-3). The vertical height should be compressed by about 20%. On the back of the monitor, change the Manual switch to OFF. The vertical height should expand to normal height. If adjustment is desired, use the "V. Size" control on the top of the monitor to control the size of the normal text mode. When in high resolution mode, adjust potentiometer R-451 for the desired vertical height. R-451 is on the bottom circuit board (Deflection Circuit Board). R-451 can be accessed through a slot in the bottom of the right side cover plate. R-451 should be either the first or the second potentiometer back from the front of the monitor (use a flashlight and read the printing on the circuit board). When adjusting R-451, use an insulated screwdriver so as not to short R-451 to the metal monitor case (ground). With the Manual switch ON, R-451 is disconnected. With the Manual switch OFF, R-451 is controlled by the switching transistor TR-829, which is controlled by the polarity of the vertical sync signal.

Re-assemble the monitor and be proud of your NEC Multisync VGA compatible monitor.

Installing the Video Board

The following was originally printed in REMark Magazine, Volume 8, Issue 5, May 1987, "Installing an EGA Video Board" by R. Maskasky. I have had numerous requests for the information contained in that article. The pertinent information from the original article is reprinted here:

The Z-151 computer utilizes a peculiar hardware configuration on the video board. The MFM-150 ROM requires 16 kilobytes of RAM in order to operate. Unfortunately, this RAM was located on the Video board, making removal of the video board impossible. An EGA/VGA board will not operate properly with the Zenith Data Systems Video board installed. The simplest solution to this problem is to purchase a special address decoder ROM from your Heath/Zenith dealer. This ROM disables the Video portion of the Video board and still allows the system RAM to operate on the Heath/Zenith video board. This solution uses an extra expansion slot in the computer, the Heath/Zenith video board is still required to provide 16 K of RAM. ZDS designed the CPU board to hold the extra RAM required, but this requires some reorganization of the chips on the CPU board.

The following modification requires some special equipment, such as a ROM burner capable of burning a 27256 EPROM and a 74S287 ROM.

First, we must free a 128 K memory socket. This can be accomplished by copying the two 128 K system ROMs, U207 and U208, into a single 27256 EPROM (this EPROM must be 150 nanoseconds or better if the machine is to be used at 8 MHz). This 27256 chip will eventually replace U207. **Note:** The program from U208 should be placed in the lower half of the 27256 and the program from U207 should be placed in the upper half of the 27256 chip.

The address decoder chip U236 must be replaced with another address decoder to allow access to the full 256 K of ROM as U207. The address decoder chip required is a Harris 7611 or a 74S287. This chip is a high-speed ROM and requires the following pattern be burned into it using a Bipolar ROM Burner.

Address	Data
0000	0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
0010	0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
0020	0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
0030	0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
0040	0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
0050	0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
0060	0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
0070	0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
0080	0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
0090	0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
00A0	0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
00B0	0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
00C0	0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
00D0	0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
00E0	0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F 0F
00F0	09 09 09 09 0F 0F 0F 0F 0A 0A 0A 0A 0A 0A 0A 0A

Next, we will install a 16 kilobyte RAM in socket U208 on the CPU board. The NEC 43256 Static RAM is recommended. This chip is actually 32 kilobytes, but is pin compatible, and can be used as a 16 kilobyte RAM. This chip sells for about $15.00 (mail-order) for a 120 nanosecond part.

Now we are only but a few jumpers away from eliminating the Video Board. Remove all jumpers from P203 and install the following jumpers:

P203 Pins 1 to 6 (ROM 2 = RAM)
P203 Pins 5 to 10 (ROM 1 = 256K)

I have a limited number of 74S287 ROMs pre-programmed at a price of $25.00. The BIOS ROM is copyrighted by Zenith Data Systems, and therefore, cannot be legally copied for commercial purposes. ROM Burners are fairly common and transferring the contents of the ROM from the two 27128 ROMS to a single 27256 EPROM is left up to the reader. The ROM contents is considered software and can be copied by individuals for personal use, provided those individuals own a legal copy of the software (ROM).

Continued from Page 44

ery of the lost directory will also recover its subdirectory tree and all associated files.

By now the MS-DOS file system should make sense to you. I suggest that you experiment with deleting and recovering files on a floppy disk until you become confident in file recovery. The next time you have a problem with your disk, or accidentally delete some files, you will be back on your feet again with little effort.

About the Author

Gale Horst is a Systems Software Engineer at Zenith Data Systems. He is the author of *General Disk Utilities* and is also involved in the development of ZDS' MS-DOS.

Continued from Page 38

macros. These macros and their functions are as follows:

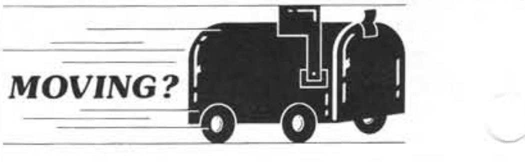
Macro Name	Function
<ALT-B>	Retrieves appointments into the word processor. Also retrieves phone directory or note card data into the word processor in secondary merge file format. What it does depends on where the cursor is when the macro is executed.
<ALT-D>	Retrieves the calculator result into the word processor.
<ALT-E>	Retrieves blocked spreadsheet cells into the word processor. Intended for use with the travel report macro.
<ALT-I>	Creates a travel itinerary form in the word processor.
<ALT-L>	Creates a business letter form in the word processor.
<ALT-M>	Creates a memorandum form in the word processor.
<ALT-O>	Retrieves note card or phone directory data into the word processor in standard text format.
<ALT-Q>	Retrieves the name, address and phone number of the current phone directory card into the word processor. Intended for use with the travel itinerary macro.
<ALT-T>	Creates a travel report form in the word processor.

If you don't care for the format of the forms created by these macros, you can define new macros with these names. The predefined macros will be over-written if you do so.

Conclusion

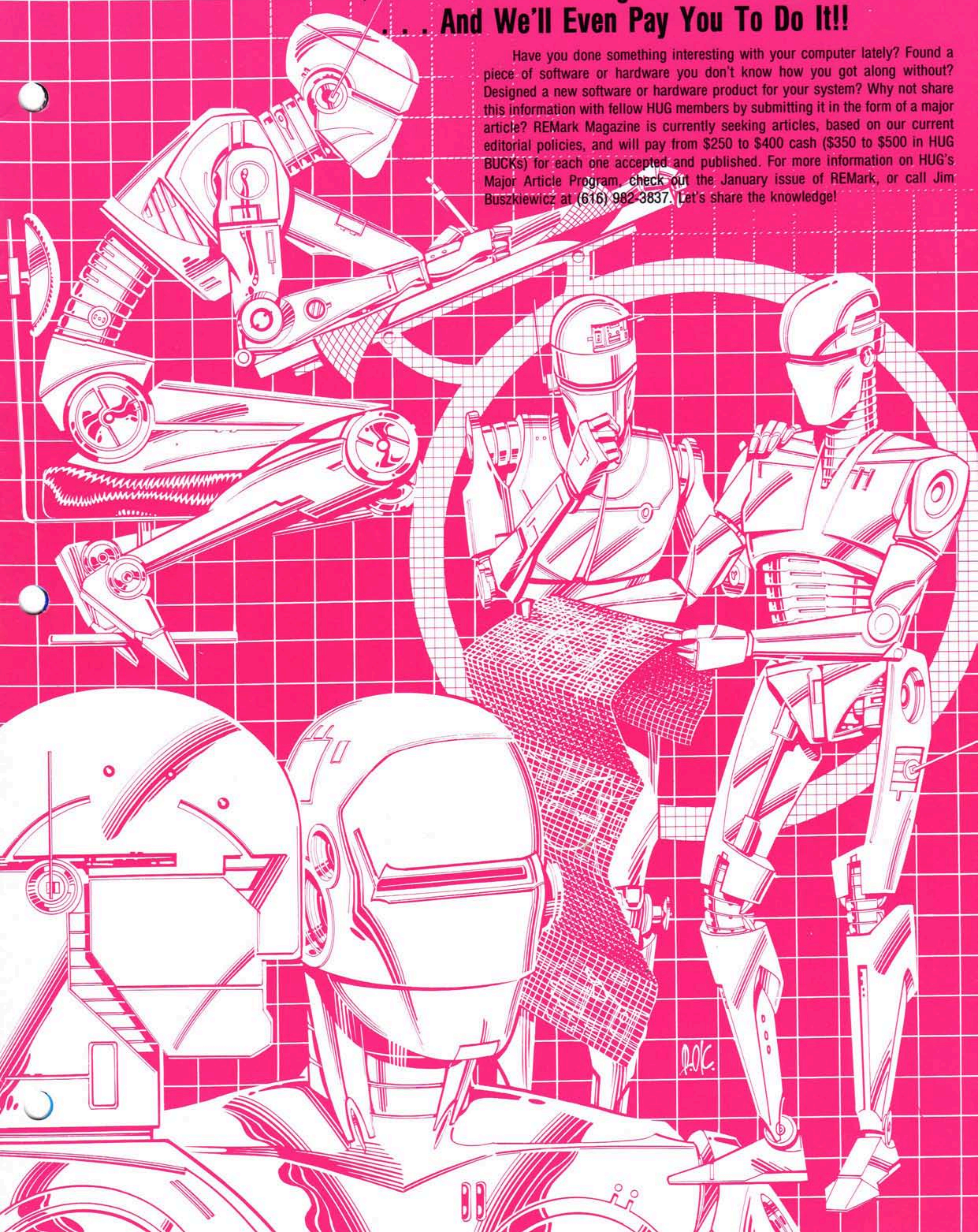
WordPerfect Executive contains an impressive array of features in a program that fits on a single 3.5 inch disk. Although the word processing and spreadsheet modules are not as powerful as their stand alone counterparts, the modules are more than adequate for most purposes. Data base management capability is limited, but useful. The appointment scheduler is very nice.

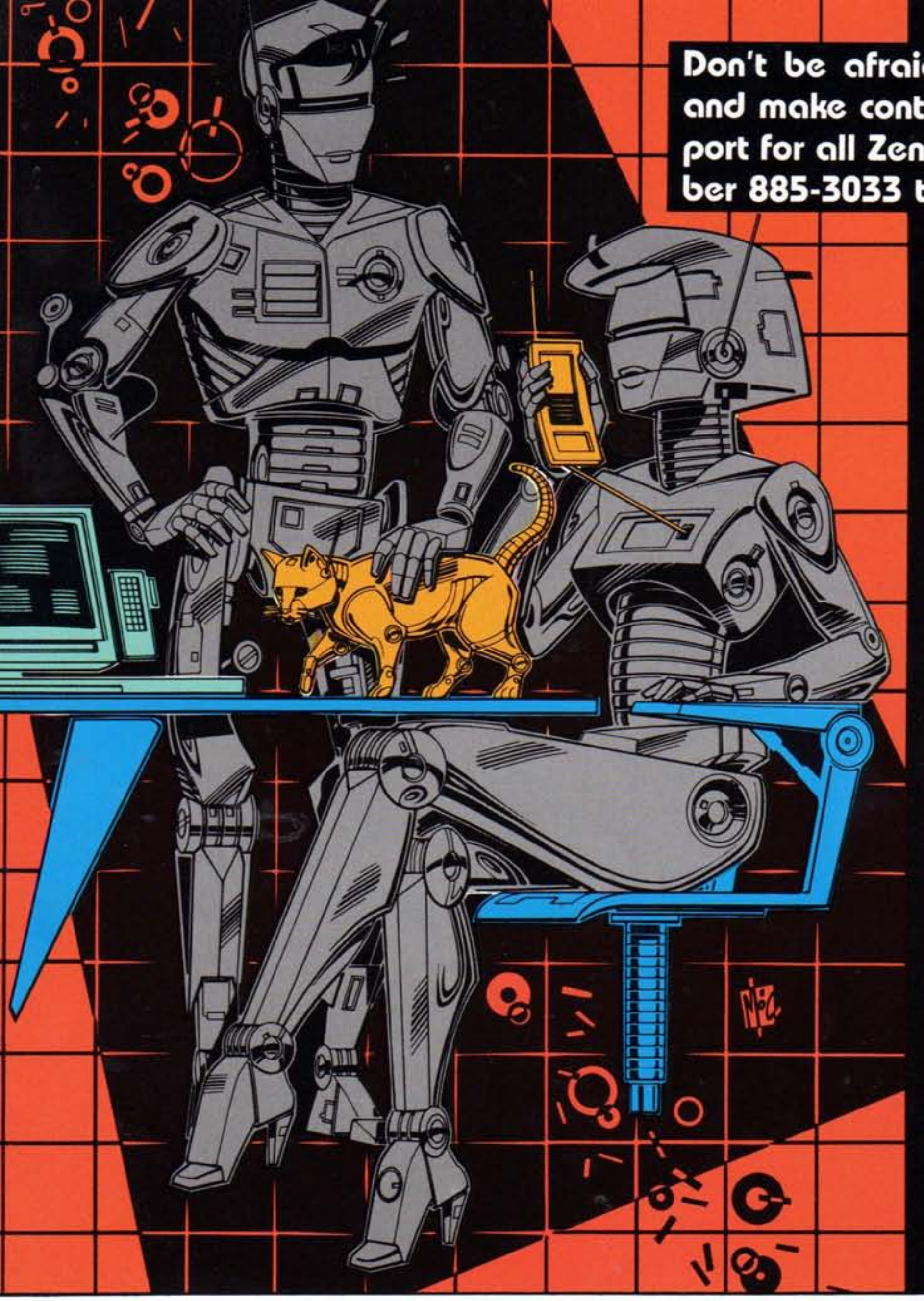
This package would not suit me as my primary word processor, spreadsheet or data base manager, but that's not what I bought it for. It suits me fine for use in my laptop computer while I'm on the road.



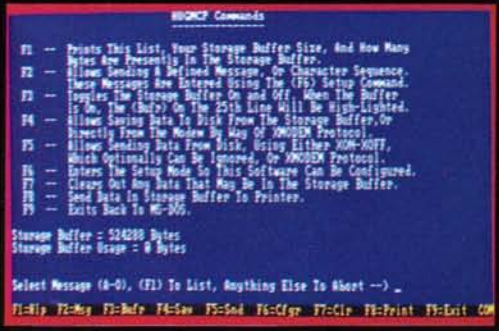
Authors, Share Your Knowledge With Other HUG Members, . . . And We'll Even Pay You To Do It!!

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 issue of REMark, or call Jim Buszkiewicz at (616) 982-3837. Let's share the knowledge!

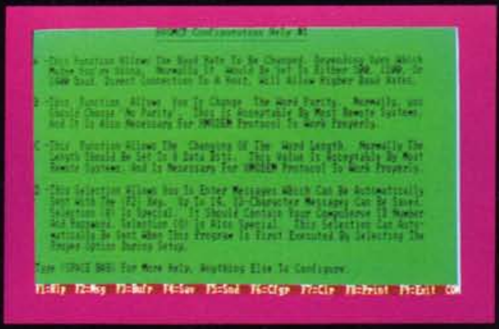




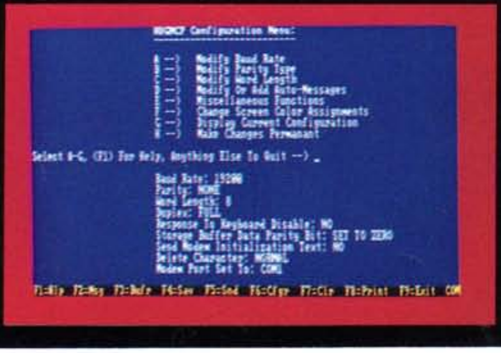
Don't be afraid to communicate! Get HUGMCP and make contact the easy way. Now with support for all Zenith Laptops, order HUG Part number 885-3033 today.



```
                         HUGMCP Commands
                         ---------------

F1 -- Prints This List, Your Storage Buffer Size, And How Many
      Bytes Are Presently In The Storage Buffer.
F2 -- Allows Sending A Defined Message, Or Character Sequence.
      These Messages Are Entered Using The (F6) Setup Command.
F3 -- Toggles The Storage Buffer On and Off. When The Buffer
      Is On, The (Bufr) On The 25th Line Will Be High-Lighted.
F4 -- Allows Saving Data To Disk From The Storage Buffer,Or
      Directly From The Modem By Way Of XMODEM Protocol.
F5 -- Allows Sending Data From Disk, Using Either XON-XOFF,
      Which Optionally Can Be Ignored, Or XMODEM Protocol.
F6 -- Enters The Setup Mode So This Software Can Be Configured.
F7 -- Clears Out Any Data That May Be In The Storage Buffer.
F8 -- Send Data In Storage Buffer To Printer.
F9 -- Exits Back To MS-DOS.

Storage Buffer = 524288 Bytes
Storage Buffer Usage = 0 Bytes

Select Message (A-O), (F1) To List, Anything Else To Abort --> _
F1=Hlp  F2=Msg  F3=Bufr  F4=Sav  F5=Snd  F6=Cfgr  F7=Clr  F8=Print  F9=Exit  COM
```



```
                    HUGMCP Configuration Help #1

A -This Function Allows The Baud Rate To Be Changed. Depending Upon Which
   Modem You're Using, Normally It Would Be Set To Either 300, 1200, Or
   2400 Baud. Direct Connection To A Host, Will Allow Higher Baud Rates.

B -This Function Allows You To Change The Word Parity. Normally, you
   Should Choose 'No Parity'. This Is Acceptable By Most Remote Systems,
   And It Is Also Necessary For XMODEM Protocol To Work Properly.

C -This Function Allows The Changing Of The Word Length. Normally The
   Length Should Be Set To 8 Data Bits. This Value Is Acceptable By Most
   Remote Systems, And Is Necessary For XMODEM Protocol To Work Properly.

D -This Selection Allows You To Enter Messages Which Can Be Automatically
   Sent With The (F2) Key. Up To 16, 32-Character Messages Can Be Saved.
   Selection (A) Is Special. It Should Contain Your CompuServe ID Number
   And Password. Selection (O) Is Also Special. This Selection Can Auto-
   matically Be Sent When This Program Is First Executed By Selecting The
   Proper Option During Setup.

Type (SPACE BAR) For More Help, Anything Else To Configure.
F1=Hlp  F2=Msg  F3=Bufr  F4=Sav  F5=Snd  F6=Cfgr  F7=Clr  F8=Print  F9=Exit  COM
```



```
                  HUGMCP Configuration Menu:
                  --------------------------

                  A --)   Modify Baud Rate
                  B --)   Modify Parity Type
                  C --)   Modify Word Length
                  D --)   Modify Or Add Auto-Messages
                  E --)   Miscellaneous Functions
                  F --)   Change Screen Color Assignments
                  G --)   Display Current Configuration
                  H --)   Make Changes Permanant

Select A-G, (F1) For Help, Anything Else To Quit --> _

                  Baud Rate: 19200
                  Parity: NONE
                  Word Length: 8
                  Duplex: FULL
                  Response To Keyboard Disable: NO
                  Storage Buffer Data Parity Bit: SET TO ZERO
                  Send Modem Initialization Text: NO
                  Delete Character: NORMAL
                  Modem Port Set To: COM1
F1=Hlp  F2=Msg  F3=Bufr  F4=Sav  F5=Snd  F6=Cfgr  F7=Clr  F8=Print  F9=Exit  COM
```

Heath Users' Group

P.O. Box 217
Benton Harbor, MI 49022-0217

$2.50

P/N 885-2127